Virtual Tape Control System

Managing VTCS

MVS Software

Version 6.2

docs.sun update only

June 2010, Revision E

Submit comments about this document by clicking the Feedback [+] link at: http://docs.sun.com

Preface

Oracle’s StorageTek Virtual Tape Control System 6.2.0 (VTCS 6.2.0, hereafter referred to as “VTCS”) is MVS host software, which together with the portions of NCS 6.2.0 that support VTCS and the Virtual Tape Storage Subsystem (VTSS), comprises Virtual Storage Manager (VSM).

Audience

This guide is for StorageTek or customer personnel who are responsible for managing VTCS.

Prerequisites

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

- MVS or OS/390 operating system
- JES2 or JES3
- System Management Facility (SMF)
- System Modification Program Extended (SMP/E)
- Nearline Control Solution (NCS)

About This Book

This book is new for 6.2, and is designed for anybody whose job is to keep VTCS up and running. VTCS/NCS is basically a server, and like all servers, if you install and configure it correctly, construct intelligent policies, and provide a reasonable amount of conscientious management, VTCS pretty much takes care of itself.

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

- "The Toolkit" on page 1 describes, in table format, the VTCS tools you use to do the management tasks in this book. "The Toolkit" describes the requirements for and value add for these tools. For the details, you'll want to have within easy reach either *VTCS Command and Utility Reference* or *VTCS Quick Reference*.
- "Using the VTCS Dashboard" on page 3 talks about the stuff you need to do on an ongoing basis. What happens if you don't change the oil and filter in your car every 3,000 miles? Answer: The same thing that happens if you don't regularly consult your VTCS Wellness Meters (Display command and MVC and VTV reports). In addition to monitoring the health of your VTCS system, there are several key items you'll want to put in place, such as a Demand MVC Space Reclamation batch job, and it's all in "Using the VTCS Dashboard".
- "Working the Must Do (Sometime) Chores List" on page 19 consists of items you do...well, as they need to be done. Think of what you do when you notice that you're down to the wear bars on your tires...you buy new tires, and quickly. For most drivers, this might be as much as 30,000 miles plus. I drive at high speeds on interstates most of the time, so I don't get much more than 15,000 miles, even on the best tires.

  There's obviously some overlap/grey area between periodic and as-needed management, just as there's some overlap between these two categories and resolving problems. Most likely, if you're scrupulous about your management tasks, you won't *have* to resolve many problems, and vice versa. "Periodic" and "As-Needed" are two general boxes, and you have to cut to fit according to the needs of your shop.
- "Finding and Fixing VTCS Problems" on page 47 is what to do when things go off kilter. "Finding and Fixing VTCS Problems" is here to help you get back on track ASAP, and has some ideas for staying out of trouble in the first place.

What's New in this Book?

Revision E

Revision E of this guide contains technical updates and corrections.

Revision D

Revision D of this guide contains technical updates and corrections, including information about the enhancements described in TABLE P-1.

TABLE P-1 Updates to Managing VTCS 6.2, Revision D

This Enhancement...	...is described in...	...and requires the following...
Using manual drives as RTDs	"Defining Manual RTDs" on page 25	PTF L1H156E

The VTCS 6.2, Revision C of this guide contains information about the VTCS 6.2 enhancements described in TABLE P-2.

TABLE P-2 VTCS 6.2 Updates to Managing VTCS, Revision C

This Enhancement...	...is described in...	...and requires the following PTFs...
INVENTRY utility	"Checking Nearline Tape Status (Daily)" on page 8	■ L1H14OC for SWS6200 ■ L1H14OA for SOS6200 ■ L1A00PL for SMC6200 ■ L1H14OB for SMS6200
MVCATTR Control Statement	"Using MVCATTR" on page 53	■ SOS6200 - L1H150A ■ SWS6200 - L1H1509 ■ SMS6200 - L1H1508
MVCMAINT enhancements	"Replacing MVC Media" on page 32	L1H14JK (SWS6200)

Contents

CHAPTER 1

The Toolkit

TABLE 1-1 describes the tools you need to do the tasks in the rest of this book. Remember, this is the Executive Summary for these tools: "This is a picture of a hammer, which you use to pound nails." For more information about the hammer, see either *VTCS Command and Utility Reference* or *VTCS Quick Reference*.

TABLE 1-1 The "Managing VTCS" Toolkit

This tool...	...does the following...	...and is used to do these tasks
AUDIT	Updates the MVC and VTV information in the HSC CDS.	■ "Changing RTD Device Types" on page 23 ■ "Changing MVC Attributes with MVCMAINT" on page 33 ■ "Logically Dismounting VTVs in an Offline VTSS" on page 42
DELETSCR	Deletes scratch VTVs from VTSSs and unlinks any migrated VTVs from MVCs.	"Deleting Scratch VTVs" on page 38
Display	Displays the status of All Things VTCS and, as such is one of your Primary Wellness Meters.	■ "Checking Virtual Tape Status (Daily)" on page 4 ■ "Checking Nearline Tape Status (Daily)" on page 8 ■ "Doing Demand MVC Space Reclamations" on page 20 ■ "Draining MVCs" on page 31 ■ "Poor VTV Mount Performance" on page 48 ■ "Poor Migration Performance" on page 49 ■ "RTD/MVC Failures" on page 53 ■ "What the RTV Utility Can Recover" on page 62
INVENTRY utility	Lists all VTVs on specified MVCs.	"Checking Nearline Tape Status (Daily)" on page 8
MIGRATE	Demand migrates VTVs to MVCs.	"Doing Demand VTV Migrations" on page 21
MVCATTR	Lets you specify the "swap to" RTD device type for each MVC media name.	"Using MVCATTR" on page 53

TABLE 1-1 The “Managing VTCS” Toolkit

This tool...	...does the following...	...and is used to do these tasks
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).	“Draining MVCs” on page 31
MVCMAINT	Sets MVC attributes.	■ “Changing MVC Attributes with MVCMAINT” on page 33 ■ “Replacing MVC Media” on page 32
MVCRPT	Reports the status of your VSM system’s MVCs and is Primary Wellness Meter #2.	■ “Looking at the Big Picture (Weekly)” on page 12 ■ “Draining MVCs” on page 31 ■ “Recovering the CDS” on page 58 ■ “Using the RTV Utility” on page 62
RECLAIM	Does demand MVC space reclamation.	“Doing Demand MVC Space Reclamations” on page 20
RECONcil	Reconciles VTV media and location (moves VTVs from one Storage Class to another).	“Changing VTV Storage Class with RECONcil” on page 43
The RTV utility	Converts VTVs contained on MVCs to data sets on Nearline volumes (real tape volumes).	“Using the RTV Utility” on page 62
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)” on page 4 ■ “Doing Demand VTV Migrations” on page 21
Vary RTD	Changes RTD state	■ “Checking Nearline Tape Status (Daily)” on page 8 ■ “Working with RTDs” on page 22
Vary VTSS	Changes VTSS states on all hosts.	“Working with VTSSs” on page 35

CHAPTER 2

Using the VTCS Dashboard

“Using the VTCS Dashboard” is basically about watching your Primary Wellness Meters to make sure everything is on track and doing what’s necessary if it isn’t. The main pieces and parts of your virtual tape empire are VTSSs, VTDs, VTVs, RTDs, and MVCs, and so it’s not surprising that a lot of your daily and weekly routine is making sure they’re all present and accounted for.

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's easy to see why it's important to make sure all your VTSSs, VTDs, and VTVs are doing their things.

▼ To check Virtual Tape status:

1. **Enter a Display VTSS command.**

   You should see something like FIGURE 2-1.

```
VTSSNAME   CAPACITY(  DBU     HI      LOW     VTV       MX    MN    DEF   AUTOMIG  STATE
           MB)

                              AMT     AMT     COUNT     MT    MT    ACS

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
```

FIGURE 2-1 Example output from Display VTSS - Everything in the Green

FIGURE 2-1 shows the status of four fairly healthy 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 slightly the LAMT, which means there's still space to grow on the VTSS before auto-migration kicks in. You might even consider routing more work to these VTSSs to optimize your investment in virtual tape.
- Let's assume you have 8 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 kick off, there are enough tasks to handle the load.

What if Display VTSS doesn't look so good? Maybe something like FIGURE 2-2...

VTSSNAME	CAPACITY(MB)	DBU	HI	LOW	VTV	MX	MN	DEF	AUTOMIG	STATE
			AMT	AMT	COUNT	MT	MT	ACS		
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

FIGURE 2-2 Example Output from Display VTSS - A VTSS in Trouble

FIGURE 2-2 shows the status of four VTSSs that have some major operational issues:

- At least they're 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 in the Red Zone. Anything in the 90 plus range means the VTSSs are struggling to auto-migrate VTVs, which isn't surprising because...
- ...again, let's assume you have 8 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 didn't like what you saw in Step 1, tune up your operating parameters.**

First, get some more migration tasks going:

```
.vt 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 will get you out of the danger zone, and the next time around, migration will kick in sooner and bring the buffer to a lower DBU.

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

FIGURE 2-3 shows an example of Display VTD output.

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

FIGURE 2-3 Example output from Display VTD - Smooth Sailing

Once again, we're in good shape in FIGURE 2-3 with some VTDs gainfully employed and others ready to join in.

What if all VTDs showed VTVs mounted? Not so good, because then you risk job allocation failures if no drives are available. If it's because of something like the *imbroglio* that occurred in Step 2, then just accept it and vow that, in the future, you'll arrange workloads to let your VTDs better handle the influx of data. If this is a long term problem, however, you may need a more power...more VTSSs, or upgrades to VTSSs with more capacity and more VTDs.

3. Now it's time to see if you have enough scratch VTVs on hand...

...by entering a Display SCRATCH command, which produces the output shown in FIGURE 2-4.

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

FIGURE 2-4 Example output from Display SCRATCH

In FIGURE 2-4, what you're seeing are VTV scratch counts for HSC subpools. If you don't use HSC subpools for VTVs, you'll 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.

Let's say that FIGURE 2-4 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 aren't enough scratch volumes available.

Actually, you don’t do the scratching, it’s your TMS that’s supposed to do the work, and the initial configuration should have defined your VTV volsers to your TMS. If that didn’t happen, go back and do it. More likely, you may have added VTV ranges via 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 via your TMS is just part of the puzzle. 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 Big Move (data gone, never to return...), we’re going to make a judgement call and say it’s an “as-needed” task, so if you go this route, turn to “Deleting Scratch VTVs” on page 38.

- Use CONFIG 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’s not just CONFIG...you also have to do the appropriate MVCPOOL statement(s), 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’re 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)

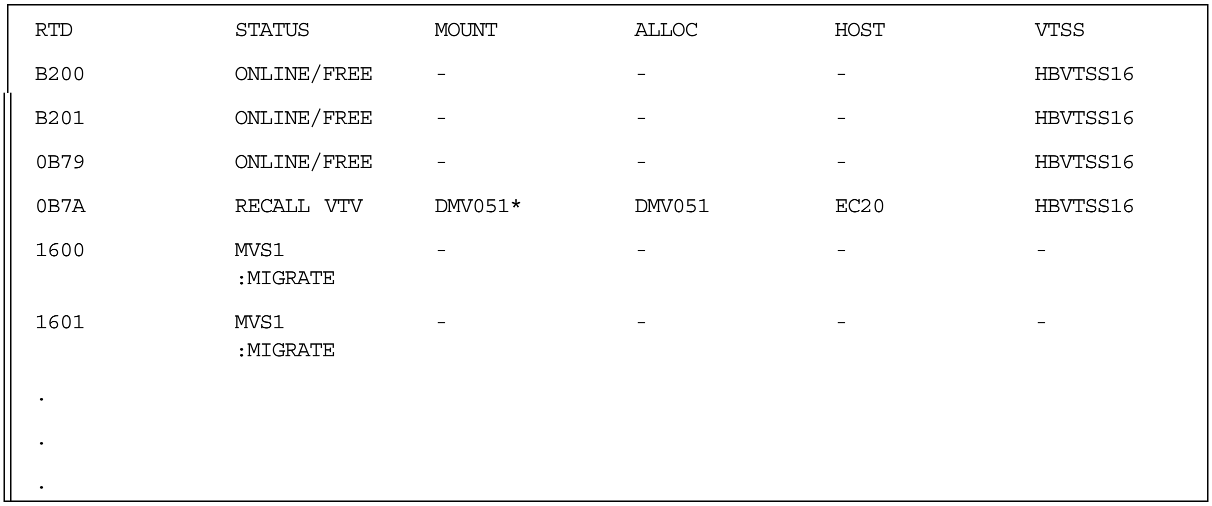
In “Checking Virtual Tape Status (Daily)” on page 4, we talked about how important it is to ensure that your system’s VTSS, VTDs, and VTVs are all present and accounted for.

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

▼ To check Nearline Tape status:

1. **Enter a Display RTD.**

   A good picture might be something like FIGURE 2-5.



```
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
.
.
.
```

FIGURE 2-5 Example output from a VT Display RTD command - all is well

In FIGURE 2-5 on page 8, things are moving along smartly because there’s a good balance of RTDs migrating, recalling, and available for new work...which is **not** true in FIGURE 2-6.

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

FIGURE 2-6 Example output from a VT Display RTD command - RTDs in Distress

If FIGURE 2-6 is a result of the kind of emergency measures employed in Step 2, then there’s 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...by all means, vary them offline to MVS and run, do not walk, to Vary RTD to make them available to VTCS.

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

FIGURE 2-7 shows an example of Display MVCPool output with no MVC pool name specified, so you see information for all your system's MVCs.

```
             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 2-7 Example output from Display MVCPool (no pool name specified)

FIGURE 2-7 shows that your MVC collection is Looking Good. There appear to be plenty of Free MVCs (100% usable space, no migrated VTVs) and plenty of GBs of free space across multiple ACSs and MVC media types. Numbers for MVCs eligible for reclaim are pretty low, meaning that automatic space reclamation probably won't get in the way of migrate/recall activity.

Used MVCs looks okay 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 2-8?

```
            MVCPOOL  INFORMATION

ACS         MEDIA       FREE-MVCS               RECLAIM-MVCS            USED-MVCS

                        VOLS        GB          VOLS        GB          VOLS        GB

00          ECART       310         248         4           1.2         300         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       115         115         55          529.5       365         1160


NON-LIB     STK2P       22          1100        0           0           12          1565

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

FIGURE 2-8 Example output from Display MVCPool - Trouble in MVC Land

As you can see, things have gotten *considerably grimmer* in ACS 01. You're not in big trouble yet, but you're close. What to do? Well, consider the following, probably in this order:

- Do demand reclamation to free space; for more information, see "Doing Demand MVC Space Reclamations" on page 20.
- Add MVCs as described in "Adding MVCs to the MVC Pool" on page 26.
- 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.

3. **Consider running the INVENTRY utility.**

   The INVENTRY utility lists all VTVs on specified MVCs. Additionally, you can use INVENTRY to:

   - Request that the contents of the specified MVCs are cross-checked with the information recorded in the CDS.
   - Terminate the processing based on the end-of-tape position recorded in the CDS for the MVC, rather than at the MVC's physical end-of-tape.
   - Have the processing terminated for an MVC the first time a VTV is discovered that is incompatible with the information in the CDS.

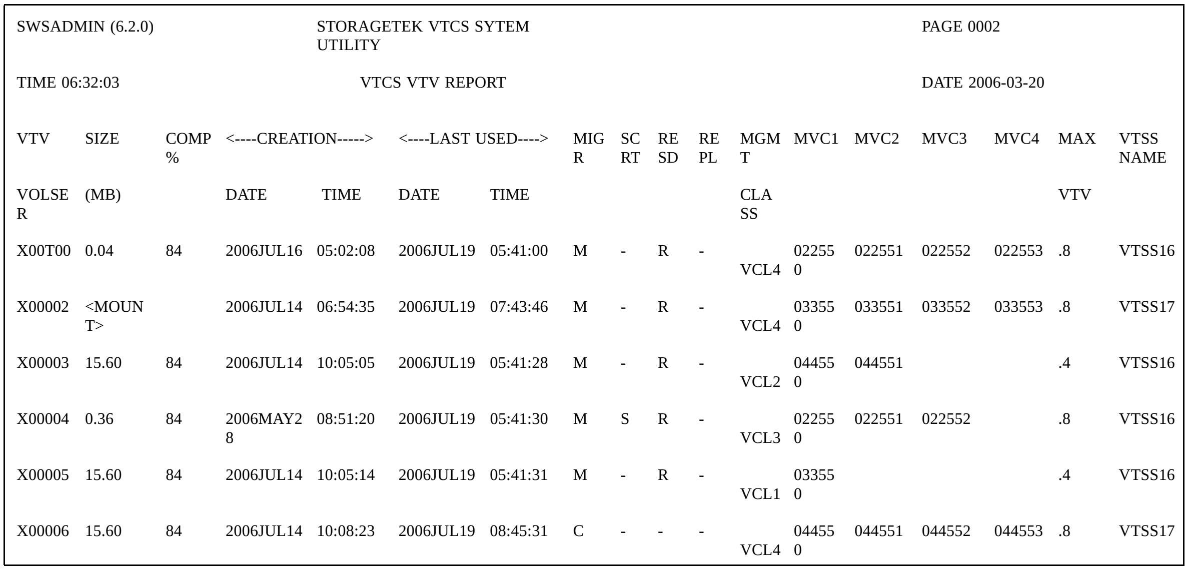
   For more information about `INVENTRY`, see *VTCS Command and Utility Reference*.

Looking at the Big Picture (Weekly)

This one isn't too complicated, and basically consists of running your other two Primary Wellness Meters (MVC Report and VTV Report) on a weekly basis.

Using the VTV Report

Let's start with a VTV Report, which looks like FIGURE 2-9.



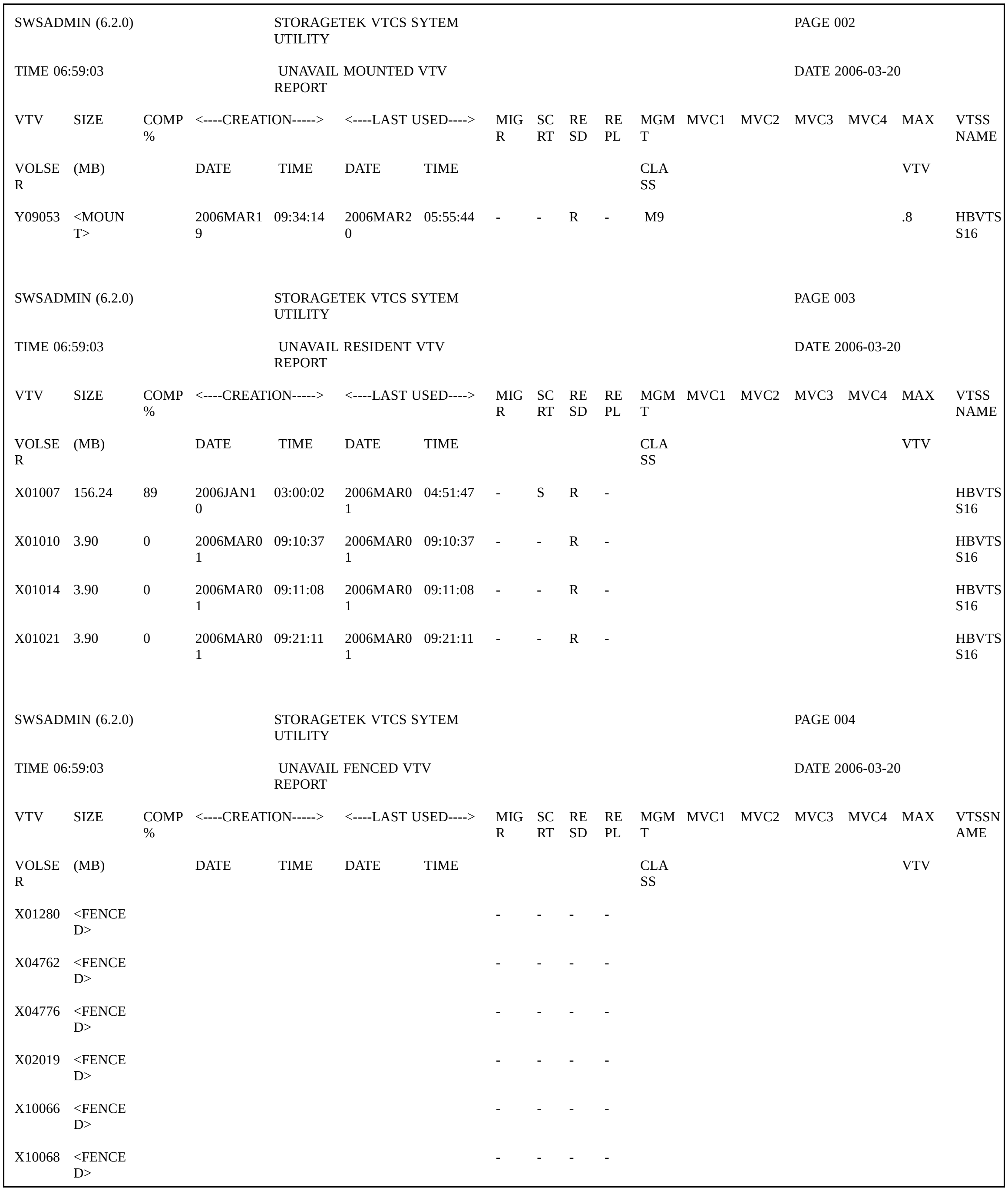
```
SWSADMIN (6.2.0)                 STORAGETEK VTCS SYTEM                                              PAGE 0002
                                 UTILITY

TIME 06:32:03                         VTCS VTV REPORT                                               DATE 2006-03-20

VTV     SIZE     COMP  <----CREATION----->      <----LAST USED---->   MIG  SC   RE   RE   MGM   MVC1   MVC2    MVC3    MVC4    MAX   VTSS
                 %                                                     R    RT   SD   PL   T                                          NAME

VOLSE   (MB)           DATE        TIME         DATE        TIME                           CLA                                  VTV
R                                                                                          SS

X00T00  0.04     84    2006JUL16   05:02:08     2006JUL19   05:41:00   M    -    R    -          02255  022551  022552  022553  .8    VTSS16
                                                                                           VCL4  0

X00002  <MOUN          2006JUL14   06:54:35     2006JUL19   07:43:46   M    -    R    -          03355  033551  033552  033553  .8    VTSS17
        T>                                                                                 VCL4  0

X00003  15.60    84    2006JUL14   10:05:05     2006JUL19   05:41:28   M    -    R    -          04455  044551                  .4    VTSS16
                                                                                           VCL2  0

X00004  0.36     84    2006MAY2    08:51:20     2006JUL19   05:41:30   M    S    R    -          02255  022551  022552          .8    VTSS16
                       8                                                                   VCL3  0

X00005  15.60    84    2006JUL14   10:05:14     2006JUL19   05:41:31   M    -    R    -          03355                          .4    VTSS16
                                                                                           VCL1  0

X00006  15.60    84    2006JUL14   10:08:23     2006JUL19   08:45:31   C    -    -    -          04455  044551  044552  044553  .8    VTSS17
                                                                                           VCL4  0
```

FIGURE 2-9 Example output from VTVRPT

At first glance, the VTV Report looks pretty 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 shop? To start with, you can run the VTVRPT utility against a list of volsers, a volser range, or even an individual volser. So if you have one or more VTVs in mind that you want to put under the magnifying glass, go right ahead.

Second, the VTVRPT utility also provides the OPTION(UNAVAIL) parameter, which spits out a report on unavailable VTVs that looks like FIGURE 2-10.



```
SWSADMIN (6.2.0)                    STORAGETEK VTCS SYTEM                                                 PAGE 002
                                    UTILITY

TIME 06:59:03                        UNAVAIL MOUNTED VTV                                                  DATE 2006-03-20
                                    REPORT

VTV      SIZE     COMP  <----CREATION----->   <----LAST USED---->  MIG  SC  RE  RE  MGM  MVC1  MVC2  MVC3  MVC4  MAX  VTSS
                  %                                                R    RT  SD  PL  T                                 NAME

VOLSE    (MB)           DATE        TIME      DATE        TIME                      CLA                          VTV
R                                                                                   SS

Y09053   <MOUN          2006MAR1    09:34:14  2006MAR2    05:55:44  -    -   R   -    M9                         .8   HBVTS
         T>             9                     0                                                                       S16


SWSADMIN (6.2.0)                    STORAGETEK VTCS SYTEM                                                 PAGE 003
                                    UTILITY

TIME 06:59:03                        UNAVAIL RESIDENT VTV                                                 DATE 2006-03-20
                                    REPORT

VTV      SIZE     COMP  <----CREATION----->   <----LAST USED---->  MIG  SC  RE  RE  MGM  MVC1  MVC2  MVC3  MVC4  MAX  VTSS
                  %                                                R    RT  SD  PL  T                                 NAME

VOLSE    (MB)           DATE        TIME      DATE        TIME                      CLA                          VTV
R                                                                                   SS

X01007   156.24   89    2006JAN1    03:00:02  2006MAR0    04:51:47  -    S   R   -                                    HBVTS
                        0                     1                                                                       S16

X01010   3.90     0     2006MAR0    09:10:37  2006MAR0    09:10:37  -    -   R   -                                    HBVTS
                        1                     1                                                                       S16

X01014   3.90     0     2006MAR0    09:11:08  2006MAR0    09:11:08  -    -   R   -                                    HBVTS
                        1                     1                                                                       S16

X01021   3.90     0     2006MAR0    09:21:11  2006MAR0    09:21:11  -    -   R   -                                    HBVTS
                        1                     1                                                                       S16


SWSADMIN (6.2.0)                    STORAGETEK VTCS SYTEM                                                 PAGE 004
                                    UTILITY

TIME 06:59:03                        UNAVAIL FENCED VTV                                                   DATE 2006-03-20
                                    REPORT

VTV      SIZE     COMP  <----CREATION----->   <----LAST USED---->  MIG  SC  RE  RE  MGM  MVC1  MVC2  MVC3  MVC4  MAX  VTSSN
                  %                                                R    RT  SD  PL  T                                 AME

VOLSE    (MB)           DATE        TIME      DATE        TIME                      CLA                          VTV
R                                                                                   SS

X01280   <FENCE                                                     -    -   -   -
         D>

X04762   <FENCE                                                     -    -   -   -
         D>

X04776   <FENCE                                                     -    -   -   -
         D>

X02019   <FENCE                                                     -    -   -   -
         D>

X10066   <FENCE                                                     -    -   -   -
         D>

X10068   <FENCE                                                     -    -   -   -
         D>
```

FIGURE 2-10 Example output from VTVRPT (UNAVAIL option)

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

Let's also talk about the flexibility you get with XML output from the VTVRPT utility. For VTCS/NCS 6.2, you can generate, for selected reports and utilities, output in either Structured XML or Comma Separated Variables (CSV) XML.

What's 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.

So for 6.2, you now have two ways you can effectively customize a stock VTV Report for your shop's needs. For more information on this topic, see *ELS UUI/XML Guide*.

Finally, you should be aware that ExLM 6.2 provides enhanced management and reporting capabilities that correspond to NCS/VTCS functions, including the VTCS MVC and VTV Reports. For more information, see TABLE A-1.

Using the MVC Report

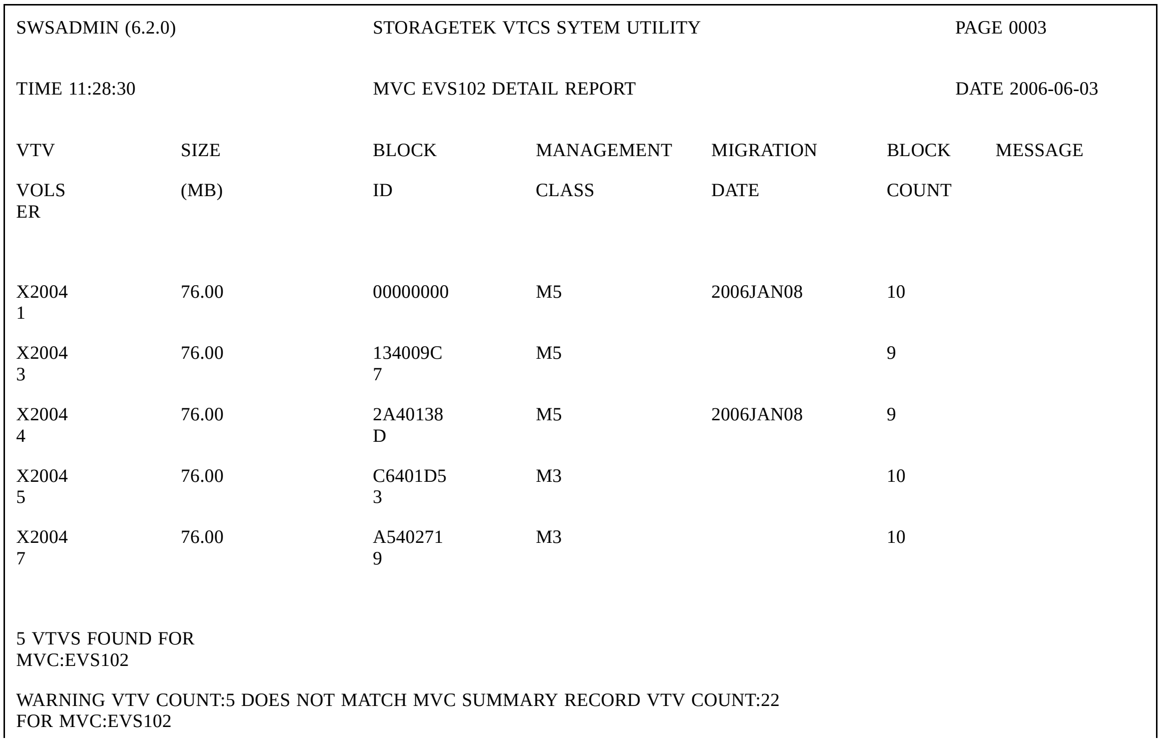
Finally, let's look at an MVC Summary Report, which looks like FIGURE 2-11.

SWSADMIN (6.2.0)					STORAGETEK VTCS SYTEM UTILITY						PAGE 0002	
TIME 09:26:54					VTCS MVC SUMMARY REPORT						DATE 2006-04-13	
MVC	NUMBER	%USED	%AVAIL	%FRAG	MEDIA	TIMES	STATUS	<-----LAST MOUNTED---->			ACS	OWNER/
VOLSER	OF VTVS				SIZE (MB)	MOUNTED	I B L D R U T M	DATE	TIME	VTSS	ID	CONSOLIDATE TIME
EVS99	200	10.80	84.57	4.63	2000	310	I - - - - U - M	2006MAR15	03:20:23	VTSS8	00	S1
EVS100	0	0.00	100.00	0.00	UNKNOWN	206	- - L - - U - -	2006MAR10	05:24:04	VTSS8	--	
EVS101	1009	99.00	0.00	1.00	400	306	I - - - - U - -	2006MAR15	03:20:23	VTSS8	00	S1
EVS102	5	8.25	91.75	0.00	400	6	I - - - - U - -	2006MAR15	04:23:04	VTSS8	00	S3
EVS103	EXPVTV	0.12	99.88	0.00	400	194	I - - - - J - -	2006MAR15	03:20:28	VTSS10	00	VTSS10
EVS104	0	0.00	100.00	0.00	400	5	I - - - R C - -	2006MAR18	03:49:14	VTSS8	00	2006APR12 03:49:14
EVS104	200	10.80	84.57	4.63	102040	254	I - - - R U T -	2006MAR18	04110:09	VTSS8	00	
EVS105	300	15.80	54.57	4.63	102040	154	I - - - R U W -	2006MAR18	04110:09	VTSS8	00	
EVS106	0	0.00	100.00	0.00	400	202	I - - - - C - -	2006MAR18	03:49:20	VTSS8	00	
EVS107	0	0.00	100.00	0.00	400	171	I - - - R E - -	2006MAR18	04:13:00	VTSS8	00	
		137	Initialized MVCs processed									
		8	Non-Initialized MVCs processed									

FIGURE 2-11 Example MVC summary report

The MVC Summary Report is a lot like the stock VTV Report...nice, if you know what you're looking for, Too Much Information otherwise.

What's perhaps more useful in a Big Picture sense are the additional fields you get from an MVC Detailed Report, as shown in FIGURE 2-12.



```
SWSADMIN (6.2.0)          STORAGETEK VTCS SYTEM UTILITY                            PAGE 0003

TIME 11:28:30             MVC EVS102 DETAIL REPORT                                 DATE 2006-06-03

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


X2004         76.00         00000000     M5            2006JAN08     10
1
X2004         76.00         134009C      M5                          9
3             7
X2004         76.00         2A40138      M5            2006JAN08     9
4             D
X2004         76.00         C6401D5      M3                          10
5             3
X2004         76.00         A540271      M3                          10
7             9


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

FIGURE 2-12 Example MVC detailed report (additional fields)

Here's some information on a micro level re 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 6.2 MVC Report:

- Generate output in either Structured XML or Comma Separated Variables (CSV) XML as described in *ELS UUI/XML Guide*.
- Use the corresponding ExLM reports, as described in TABLE A-1.

And Don’t Forget to Back Up Your CDS...

Before we exit the Periodic Tasks List, let’s talk about one Big Ticket item, which is to ensure that the CDS is backed up at least daily, and more often if possible (VTV and MVC changes typically occur throughout the day and cause CDS changes). For more information, see *HSC System Programmer’s Guide*.

Caution – A VTCS audit is **not** a substitute for frequent backup of the CDS. A VTCS audit will regain access to your VSM resources, but in general it will not completely reconstruct the VSM records in a CDS. For example, if you delete all copies of the CDS, an audit will **not** recreate any Management Class definitions. Therefore, we strongly recommend that you do **not** delete all copies of the CDS, then expect to run a VTCS audit and regain a working system *in toto*!

What happens if you do lose the primary copy of the CDS? Or worse yet, all copies? Answer: Run, do not walk, to “Recovering the CDS” on page 58.

So, In Closing...

...that’s about all there is to using the VTCS Dashboard. Sure, there’s a bunch of other stuff 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’s information for the “As Needed” part of the show.

The main thing you should take away from this chapter is do your dailies per “Checking Virtual Tape Status (Daily)” on page 4 and “Checking Nearline Tape Status (Daily)” on page 8 and your weeklies per “Looking at the Big Picture (Weekly)” on page 12, and you’ll go a long way toward keeping your VTCS system running happily...

CHAPTER 3

Working the Must Do (Sometime) Chores List

Welcome to the “Must Do (Sometimes) Chores List,” a. k. a., the “As-Needed Tasks List.” Think of your “Maintaining My Country Estate” List, which prominently features “Clean Out the Garage...Again.” What happens 5 seconds after you actually *do* clean out the Garage? Answer: “Clean Out the Garage...Again” comes off the list and you celebrate by watching reruns of “The Price is Right.” What happens the week after you clean out the garage? You guessed it...”Clean Out the Garage...Again” makes a sudden reappearance on the “Maintaining My Country Estate” list.

The same thing is probably true of the “Must Do (Sometime) Chores List” for VTCS. For example, let’s say that this is the week you decide to run DELETSCR to zero out a gaggle 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’s going to be before you have to run the same operation? Especially if you don’t change your delete on scratch policies? Answer: It might be a day, a month, or a year, but you’ll get to do it again.

No big deal, however. We’re here to help (where have you heard that before?) with helpful procedures to pare down the Must Do (Sometime) List, and, as you’ve already seen from reading “Using the VTCS Dashboard” on page 3, if you keep a close eye on your Wellness Meters, you may not even *need* a list, because they (your Wellness Meters) will tell you when it’s 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 we’ll include them here because (a) they’re proactive in nature (i. e., don’t let the garage get dirty and you’ll never have to clean it), which makes them doubly valuable as Best Practices “As Needed” chores, and (b) they’re operating techniques you can use, back out, and reintroduce as they benefit (or not) your shop at any point in time. Having said that, let’s lead off with three of our favorites in this category as described in “Doing Demand Space Reclamations, Migrations, and Recalls” on page 20.

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 will soon become obvious.

Doing Demand MVC Space Reclamations

As you already know, VSM automatically reclaims space only one MVC at a time 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've 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 handy companion (*ELS Command, Control Statement, and Utility Reference*) and you'll see some useful knobs and dials you can use to optimize demand reclamation and run it most efficiently, to whit:

- You can use **one only** of the MVCPOOL, STORCLAS, ACSid, or MVC parameters to filter the list of MVCs to process. Your Wellness Meters, as described in "Using the VTCS Dashboard" on page 3, help you narrow the list of likely suspects to an MVC Pool, Storage Class, specific ACS, or range or list of MVCs. Feed this list into RECLaim, and you're 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 sensor for detecting if no demand reclaims have happened within an interval you specify. 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 we already noted, VTCS/NCS is basically a server, which means that most of the time, it pretty much takes care of business. For example, VSM automatically manages VTSS space and migrates VTVs to ensure a balance of optimum data availability, resource use, and data protection.

That’s great for a steady-state environment, but what if you find out your VSM system is about to experience a Major League infusion of Mass Quantities of Critical 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 bells and whistles:

- You can migrate VTVs by volser (repeats allowed), Management Class, or...and this is really cool...the data set name associated with the VTV. There’s 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’s 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’s a nice Bigger Hammer 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 kicks in 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 “Mass Quantities of Data About to Enter VTSS Land” situation? For example, you’re doing end-of-year processing, and *know* there’s a bunch of jobs that will want to read data from VTVs that are on tape only. No problem, RECALL to the rescue.

As you can guess, RECALL provides you with all the fine-tuning 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...there is some fine print 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...
- ...and there is a NOWAIT option to speed things up.

Working with RTDs

There's a lot of RTD management that we'll confine to the "Finding and Fixing VTCS Problems" part of the show, because they're almost exclusively error recovery scenarios. Working with RTDs...I suppose the short form of the 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...and we already talked about the operating parameters you can use to tweak this mix in "Checking Nearline Tape Status (Daily)" on page 8.

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

The big as-needed tasks you're likely to have handed to you are related, and the first two use Vary RTD:

- "Changing RTD Device Types" on page 23, which is basically how to do a technology upgrade of some or all of your system's RTDs.
- There is a special use of varying RTDs offline/online, and we describe that in "Sharing Transports Between VSM and MVS" on page 24.
- "Defining Manual RTDs" on page 25, which you may need to do in a DR situation.

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 to the MVC Pool" on page 26.

Sharing Transports Between VSM and MVS

VSM **does not** support dynamic sharing of transports between VSM and MVS. That is, a transport cannot simultaneously be online to both MVS and to VSM as an RTD. But what you *can* do is have an RTD that is online to VTCS, vary it offline to VSM, then MVS vary it online to MVS to use as a pure Nearline drive. Or vice versa. There is, of course, some fine print, see below.

Caution – Note the following:

- You **must have physical connectivity** from VSM and MVS to support sharing transports between the two. For example, you can use an ESCON director to allow both MVS and VSM to physically connect to the same RTD.
- **Do not**, therefore, use MIM, JES3, or tape autoswitch or similar facilities to manage RTDs, otherwise an RTD can be brought online to both MVS and VSM, which can cause data loss.

  If the RTD is defined to MVS, however, you can use Vary RTD to vary the RTD offline to VSM, then use the MVS VARY command to vary the transport online to MVS. Note, however, that varying RTDs offline to VSM can impact migration, recall, and reclaim processing, especially during peak VSM workloads. Similarly, if a transport is defined to MVS and also defined to VSM as an RTD, you can vary it offline to MVS, then online to VSM. Each VTSS must have a minimum of two library-attached transports as RTDs for each media type used for MVCs.

Tip – You may want to define additional RTDs as backups (up to the limit of 8 RTDs per VTSS) and vary them offline to VSM until needed. If an online RTD malfunctions, you can vary it offline and vary a backup RTD online until the malfunctioning RTD is repaired or replaced. You can also vary backup RTDs online for peak load VTV migration or recall processing.

When a transport is online to VSM as an RTD, the logrec type OBR and MDR records for the RTD always show a channel path ID of 00, which has no meaning to the host because the transport is not online to MVS.

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.

Working with MVCs

As you already know, it's somewhat difficult to limit a discussion to any one of your virtual entities. MVCs contain VTVs, so it's hard to talk about either in a vacuum, because you inevitably end up talking about the other...and if you're discussing VTVs, you're also talking about VTSSs...and VTDs. You get the picture.

Having said that, the following sections are some bare bones procedures for doing fairly typical "as-needed" tasks with MVCs that cut across a number of reasons *why* you might run a particular procedure. For example, you might add MVCs because you're running out of space per the scenario described previously...or because you're being proactive and *don't* want to get into trouble.

▼ Adding MVCs to the MVC Pool

We're going to give you the long version. You may be able to skip some steps. For example, if you already have MVC volsers allocated via VOLATTR, continue with the next step. We're including all the requirements for adding MVCs to make sure nothing gets lost in the shuffle.

Note – If you must rerun CONFIG to define new MVC ranges, you must restart VTCS/HSC after you rerun CONFIG.

To add MVCs to the MVC pool:

1. **Create VOLATTR statements to define the MVCs to HSC.**

   The fine print here is as follows:

   - MVCs require VOLATTR statements to ensure that VTCS selects the correct RTD device type for each MVC. Therefore, select volumes for MVCs that are compatible with your system's RTD transport types.
   - For mixed–media VSM systems, select volumes that include **at least** one media type compatible with each of your system's RTD transport types.

     Note that VSM selects media for migration processing and reclaim processing according to the media types of volumes in your system's MVC pool.
   - If you define **new** Nearline volumes as MVCs, you must create MVS volsers for these volumes and initialize STANDARD, ECART, and ZCART volumes as 36-track format standard label volumes.
   - **If possible**, create a new and separate volser range for MVCs. Ensure that if you define new volumes, you do not overlap existing TMS ranges.

   Let's do a simple example, and assume that all your Nearline drives (RTDs and otherwise) are T9840Cs. The VOLLATR statement for 300 new MVCs looks like the following:

   ```
   VOLATTR SERIAL (NEW000-NEW2999) MEDIAMEDIA(STK1R) RECTECH(STK1RC)
   ```

2. **Second, use VTCS CONFIG to define all MVCs available to VTCS.**

   What does “available to VTCS” mean? In Step 1, you defined 300 new T9840C volumes to HSC. Let’s say that you have some T9840Cs that are RTDs and some that are purely Nearline drives, and you want 200 of the new volumes as MVCs with the other 100 as Nearline volumes only. You do it via the following CONFIG statement:

   ```
   CONFIG MVCVOL LOW=V1000 HIGH=V1199
   ```

   Now you have 200 new MVCs that are available to VTCS but **not** (directly) to Nearline and 100 new real tape volumes available to Nearline but not to VTCS.

   Just so we don’t forget, here’s the fine print with CONFIG:

   - If the CDS does not contain sufficient space to run VTCS CONFIG, you will also have to run HSC RECONFIG. For more information about sizing the CDS for VSM, see *Installing ELS*.
   - You can only add new MVC ranges. A range can consist of a single volume. You cannot delete or modify existing ranges.
   - A VSM audit of all MVCs will audit all MVCs defined with CONFIG including those that are *not* specified in the MVCPool statements.

3. **Next, add MVCPool statements for the new MVCs (or update your existing MVCPool statements).**

   Yes, I know...in Step 2, you made the new volumes available to VTCS, so what’s this noise with the MVCPool statement? Answer: MVCPool statements specify the pool of MVCs available for migration and consolidation requests.

   What we’re getting at here is that you might define more volumes than you immediately need with CONFIG, then expand your MVC Pool with new or updated MVCPool statements as needed. One reason for doing business this way is that adding volumes via CONFIG requires that you restart HSC/VTCS, while changing/creating MVCPool statements does not.

   In this case, we’ll say that we want everything we defined via CONFIG, so our example MVCPool statement is as follows:

   ```
   MVCPOOL VOLSER(NEW000-NEW2999)
   ```

   The above is a plain vanilla MVCPool statement, where you have no Named MVC pools, the MVC media is all the same, and you’re willing to take the global defaults for automatic reclamation.

   And now, the fine print on MVCPool statements:

   - StorageTek recommends that you use identical MVCPool statements on all hosts. A host can automigrate any VTV on any VTSS to which the host is connected, including VTVs created by another host. If your VSM configuration consists of hosts cross-connected to multiple VTSSs, therefore, separate MVC pools do **not** guarantee that a host automigrates only VTVs it creates to only its MVC pool. To most effectively segregate VTVs on groups of MVCs, use Storage and Management Classes.
   - Ensure that your MVC pool consists of volumes that physically reside in ACS that contains your system’s RTDs. If not, enter or move the volumes so that they **are** accessible.

Caution – In a VSM configuration with multiple hosts that share the same HSC CDS, StorageTek strongly recommends that you do *not* use HSC/VM to enter MVCs into an ACS, otherwise these MVCs will be eligible for selection as scratch volumes by any host in the configuration with HSC installed.

4. **Run the VT MVCDEF command on all hosts to activate the updated/new MVCPool statements.**

   For example:

   ```
   .VT MVCDEF DSN(VSM.PARMS)
   ```

5. **Finally, protect MVCs and Nearline volumes that are *not* MVCs from accidental overwrites as follows:**

   - As described above, if possible, create a new and separate volser range for MVCs to prevent HSC from writing to MVCs and to prevent VSM from writing to conventional Nearline volumes.
   - VTCS, not MVS, controls access to MVCs. The tape management system does not control VSM access to an MVC volume and does not record its usage. Access to the MVCs via an RTD bypasses the MVS intercepts put in place by the tape management system so that it does *not* record within its database any access to the MVCs by VSM and does *not* automatically provide protection against inadvertent overwrites of non-expired data on MVCs.

     Therefore, if you choose to define MVCs to the tape management system, StorageTek **strongly recommends** that you define them as non-scratch, non-expiring volumes.
   - Use your security system to restrict access to MVCs as described in *Installing ELS*.
   - HSC automatically marks newly entered MVC volumes as non-scratch. If you define existing Nearline volumes as MVCs, ensure that these volumes do not contain data you need, then run the HSC UNSCratch Utility to unscratch them. For more information, see *HSC System Programmer's Guide for MVS*.

And that's it! It's hard to believe you have to jump through all these hoops to get some new MVCs up and running, and maybe you don't. It depends on how much prework you did. If the requisite VOLATTR and CONFIG definitions are already in place, you can basically start at Step 3. The "Fine Print" is included to ensure that you don't get to the finish line, have stuff go to parts unknown, and not know why...

Removing MVCs from the Pool

Removing MVCs from the pool, what fun...and why might you be doing this? Lots of reasons, where a typical scenario might be that you're 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 to the MVC Pool" on page 26 and then remove the old media as described in "Permanently Removing MVCs" on page 29".

Note that there are occasions where you might want to temporarily remove MVCs from the pool. For example, you've got some bad media or suspected bad media. You want to remove the offenders and put in replacements, basically under the same volsers as described in "Temporarily Removing MVCs" on page 30

▼ Permanently Removing MVCs

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

1. **Delete the MVCPool statements for the MVCs you want to remove...**

   ...or update the volser ranges on existing MVCPool statements to exclude the MVCs you want to remove.

2. **Run the VT MVCDEF command on all hosts to activate the MVCPool statement changes.**

3. **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:

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

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

   For more information, see *HSC/MVS Operator's Guid*e.

5. **Remove the Nearline definitions, security restrictions, and tape management system restrictions you defined for the MVC.**

6. **If you want to reuse the tape volser for Nearline (non-VTCS) usage, do one of the following:**

   a. 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.

   b. Reenter the cartridge into the ACS.

   OR

   a. Create a new set of CDS data sets.

   b. Modify VTCS CONFIG to remove the definition of the volser as an MVC.

   c. Use HSC CDSMERGE to copy the original CDS to the new CDS.

   d. Reenter the volser into the library.

▼ 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:

   ```
   .VT 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 a 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:

   .VT 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.

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 will 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. For more information on Named MVC Pools.
- STORCLAS to drain the MVCs in a Storage Class. 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.

Replacing MVC Media

When a MVC has incurred a data check or has come to the end of its useful life, typically you physically replace the cartridge. However, you may be using information stored in the MVC record (for example, number of times mounted) to track usage for operations such as media scrubbing or to plan for media replacement. Therefore, when you replace an MVC, you may want to reset the fields in the MVC record in the CDS. You can now reset these fields via the `MVCMAINT REPLACED` parameter.

To replace MVC media:

1. **Remove all the VTVs from the MVCs.**

   Typically, you use `MVCDRAIN EJECT` to ensure that the MVC does not get selected for migration. If the MVC had a datacheck, then do an MVC audit followed by an MVCDRAIN. For example:

   ```
   .VT MVCDRAIN MVCID(V900-V999) EJECT NOWAIT
   ```

2. **Eject the MVCs from the library.**

   For example:

   ```
   EJECT VOL(V900-V999) SEQ(YES)
   ```

3. **Enter the new MVCs into the library.**

   For example:

   ```
   ENTER 01
   ```

4. **Run MVCMAINT REPLACED.**

   For example:

   ```
   MVCMAINT MVC(V900-V999)REPLACED(MEDIA)
   ```

5. **Run MVCDRAIN (without EJECT) to put the new MVCs into service.**

   For example:

   ```
   .VT MVCDRAIN MVCID(V900-V999) NOWAIT
   ```

Changing MVC Attributes with MVCMAINT

MVCMAINT is like your favorite adjustable wrench. You know, the one you use to put together a barbecue grill...or take the grill apart when the burner units rust out and you have to replace them. You can even use an 8-inch adjustable wrench to hammer a nail, in a pinch...and *we* won't tell anybody that's what you used on the new bird feeder...

So MVCMAINT is a similarly handy tool in VSM Land, and we'll describe all the marvels it can do (and why) by looking at its parameters:

- First, MVC volser (range, list, individual volser) or MANIFEST are your two MVC selection criteria. MVC volser makes sense, but why manifest? Well, 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're moving MVCs from one system to another. When you import the MVCs into the new system, it'd probably be a good idea if they started their new lives in read-only mode, so they don't get stepped on until you can straighten out the working arrangements.
- READONLY (ON or OFF). See the previous bullet. Also, remember the discussion about adding MVCs to the pool? If it's up to me, I'll 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 Kids on the Block, MVCMAINT READONLY(OFF) is the tool.
- LOST (ON or OFF). How does an MVC get lost? For example, can MVCs get lost on the way to work? 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 depreferences 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.

  Okay, what if you know the MVC isn't *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's temporarily in manual mode? You might want to (temporarily) depreference 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’s probably because you think the media is bad, and so you depreference it by setting on the “logical eject” status. You then really eject the MVC, run some tests, find out it’s 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, to whit:
  - WARRANTY. VTCS also detects media warranty expiration and sets the WARRANTY status to `ON`. Alternatively, you can use SMF, LOGREG data, or your Wellness Meters 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 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, LOGREG data, or your Wellness Meters 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.

Working with VTSSs

The main thing you can do, as needed, with VTSSs is use the VTCS Vary VTSS command/utility to vary a VTSS online, offline, or to quiesced state. It probably goes without saying, but we'll say it anyway: Always know what you're doing, and why, when you vary a VTSS offline or to quiesced state. The why is probably that the VTSS needs maintenance or you're going to remove it from the configuration, which we'll talk about in "Finding and Fixing VTCS Problems" on page 47.

First, however, we'll provide you with a handy chart that shows 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 3-1 VTSS States

If you specify the following Vary VTSS parameter...	The VTSS first goes to state...	And then goes to 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 configurations, the Clinks to the VTSS should be varied offline to stop replication processing.

▼ Removing a VTSS

This could happen, right? 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. The “how to add a VTSS” is covered in *Installing ELS*, so we’ll confine this section to what you do to take a VTSS out of the mix.

▼ To remove a VTSS

1. **Before you remove the VTSS, do the following:**
   - You do not need to empty a VTSS prior to 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.**

FIGURE 3-1 shows example 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.

```
//UPDATECFG  EXEC PGM=SWSADMIN,PARM='MIXED'

//STEPLIB    DD DSN=hlq.SLSLINK,DISP=SHR

//SLSCNTL    DD DSN=FEDB.VSMLMULT.DBASEPRM,DISP=SHR

//SLSCNTL2   DD DSN=FEDB.VSMLMULT.DBASESEC,DISP=SHR

//SLSSTBY    DD DSN=FEDB.VSMLMULT.DBASETBY,DISP=SHR

//SLSPRINT   DD   SYSOUT=*

//SLSIN      DD   *

 CONFIG

 GLOBAL      MAXVTV=32000MVCFREE=40

 RECLAIM     THRESHLD=70MAXMVC=40  START=35

 VTSS        NAME=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

 VTSS        NAME=VTSS2
```

FIGURE 3-1 CONFIG example: updating configuration to deny host access to a physically removed VTSS

Working with VTVs

There's a million things you *can* do with VTVs, but here we'll stick to a short list of the most likely tasks you might have to do on an as-needed basis: deleting scratch VTVs and changing VTV attributes.

Deleting Scratch VTVs

There are two ways you can delete scratch VTVs:

- Via policy, by specifying DELSCR(YES) on a VTV's Management Class and using HSC or ExLM scratch synchronization to do the actual scratch.
- *Ad hoc*, 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.

We're going to leave the policy discussion of deleting scratch VTVs to *Installing ELS* and stick to the *ad hoc* version. After all, *ad hoc* is basically "as needed" in Latin, and that's the theme of this book.

Before we tell you how to do it, let's dispense with the obligatory warning, which is:

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

Okay, so deleting VTVs is **not** something you do on a Tuesday afternoon because you've run out of stuff to do. If you have to manually delete scratch VTVs, it's because...you guessed it...you're in trouble to the tune of the scenario on page 9.

To prevent inadvertent VTV deletion via operator command, DELETSCR is a SWSADMIN utility only, and has the following knobs and switches:

- You can specify VTVs by volser (individual volser, list, or range), Management Class, or HSC Scratch Pool. Using your Wellness Meters, you should already have a good idea of the best way to identify the culprits and apply the corresponding DELETSCR option. You can only specify one option (VTVid, MGMTclas, or SCRpool) and if you don't specify anything, DELETSCR deletes **all** eligible VTVs...which 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're running DELETSCR proactively during a non-peak period, you might not care about MAXVTV. If you're 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 would have been deleted...at the point at which you ran DELETSCR (that is, the report is just a snapshot). A handy "try before you buy" option...

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

DELTSCR JCL Example

FIGURE 3-2 shows example 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=SWSADMIN,PARM='MIXED'

//STEPLIB     DD DSN=hlq.SLSLINK,DISP=SHR

//SLSPRINT    DD SYSOUT=*

//SLSIN       DD *

 DELETSCR MGMTCLAS(MC1) NOTREF(60) MAXVTV(800) DET
```

FIGURE 3-2 DELETSCR utility example

Changing VTV Attributes with VTVMAINT

VTVAINT is like your favorite adjustable wrench...whoops, can’t go there, already used up *that* routine. Let’s just note that VTVMAINT is another handy tool, this time for VTV maintenance, which does the following:

- Selects VTVs by volser (range, list, individual volser...your choice).
- “Unlinks VTVs from MVCs”...which I would want to do, because? Answer: It’s likely you’re 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”.
- Change 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 Bigger Hammer (I promised we weren’t going to talk about adjustable wrenches...) 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 a tricky one, and is best explained in “Logically Dismounting VTVs in an Offline VTSS” on page 45.

Note – And don’t 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 Storage Class:

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" on page 43 for procedures for moving VTVs to MVCs *ad hoc*.

▼ 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 will 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*.

Don't 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 will also dismount 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, StorageTek strongly recommends that you do not recall these non-current MVC copies!

Tip – When the offline VTSS is ready to be brought back online, StorageTek **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.

Changing VTV Storage Class with RECONcil

As described in "Changing VTV Management Class and Unlinking VTVs from MVCs" on page 41, 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: RECONcil to the rescue.

Before we fire off our first RECONcil job (SWSADMIN utility only), let's figure out *why* we want to change a VTV's Storage Class. There are basically three reasons:

- As above, you're 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 would 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 we're talking about using RECONcil to, um, 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 via 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 **don't** 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 (i. e., Do they need reconciliation or not?), you can specify one of the following RECONcil parameters...**

   ...and this is tricky, so we'll go slow:

   - 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....and we'll talk more about that in Step 2.

---

2. **Accept the default when you first run RECONcil...**

   ...which is to generate a report only, which, as you can imagine, does no data movement, but merely reports the VTVs that are candidates for reconciliation. You can guess the reason for this, but we'll explicitly state it anyway.

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

---

3. **If needed, fine tune the RECONcil job...**

   Okay, let's say you ran the report per Step 2, and it looks like you'll be reconciling until the next millennium. How can you cut the work down to size or parcel it out a piece at a time? Well, 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 "Let's say I changed some VTV Management Classes per "Changing VTV Management Class and Unlinking VTVs from MVCs" on page 41 (including their Storage Class specifications) and then ran RECONcil? What would *that* look like?" Well, 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. **Okay, you've done all the "what ifs," fine tuning, and off-peak schedule required...**

   ...now it's time to go for the gold. FIGURE 3-3 shows example JCL to run RECONcil as follows:

   - 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=SWSADMIN

//STEPLIB     DD DSN=hlq.SLSLINK,DISP=SHR

//SLSPRINT    DD SYSOUT=*

//SLSIN       DD *

  RECON MGMT (LOCALPROD1,LOCALPROD2) MAXMVC(60) CONMVC(8)
ELAPSE(360) MOVEVTV
```

FIGURE 3-3 Example JCL for the RECONcil utility

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...

CHAPTER 4

Finding and Fixing VTCS Problems

This section, as advertised, is about what to do when Things Go Bump in the Night. Okay, you've already done your dailies per"Using the VTCS Dashboard" on page 3, and your as-neededs per "Working the Must Do (Sometime) Chores List" on page 19, and things are *still* not going well. Unfortunately, things go wrong in spite of your best efforts...but here's the place where you find out how to get VTCS back on track when the lights go out, starting with the garden variety stuff you'll probably run across in "Fixing Common Problems" on page 48.

Fixing Common Problems

There's nothing common about a VTCS problem if it happens to you, which you already know. "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 we begin with VTV mount performance problems, let's emphasize that these are common problems that you can *generally* diagnose and fix on your own. After making a reasonable effort, however, or even in some cases if you already have, if things still aren't working out, it's time to call for help as described in "Customer-initiated Maintenance" on page xv. There are also some tools that we don't talk about here, like traces, because you basically only want to use them under the direction of StorageTek 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 will terminate but migration scheduler will start 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.

Note – VTCS **does not** attempt to resolve outstanding mounts on VTDs during HSC initialization. You must resubmit the jobs requesting these mounts.

Poor Migration Performance

If VTV migration occurs very slowly, check the following:

- Start with Display MIGrate, which will show 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 Nearline Tape Status (Daily)" on page 8. 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's only one thing worse than poor migration performance, and that's no migration at all. Fortunately, VTCS 6.2 is enhanced to provide more detailed information about migration failures as described in the following sections:

- "Messages Enhancements" on page 50
- "Display STORCLas" on page 51
- "Enhanced MVC Pool Validation" on page 51
- "Enhanced Storage Class Validation" on page 52

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 bottom line is that you're now getting greater granularity of 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 6.2 provides more 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.
- Does each MVC Pool volser range overlaps the range(s) defined by VTCS CONFIG? If not, message SLS6847W is issued.

Even better, VTCS now does some up-front work to ensure that you *don't* end up *in extremis*...

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're 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.

Using MVCATTR

When an error occurs while reading an MVC on an RTD, VTCS may swap the MVC to another RTD to retry the operation. If the swap is for an MVC audit then the selection of the "swap to" device type **excludes** the device type that the error occurred on. Otherwise, the "swap to" device type is restricted to the native device type (device type MVC was created on). However, **note that**:

- T9840D devices are the **preferred** device type for reading an MVC created on **any** T9840 device type (T9840A/T9840B/T9840C/T9840D).
- T10KB devices are the **preferred** device type for reading an MVC created on **any** T10K device type (T10KA/T10KB).

This enhancement provides the MVCATTR control statement, which lets you specify the "swap to" RTD device type for each MVC media name. For more information, see *VTCS Command and Utility Reference*.

Note – It is possible to get the SLS6949I message on a write swap. This will only occur if the MVCATTR requests a swap to the native device type. For example, if you have an MVCATTR for MEDIA(STK1RC) and request SWAPTO(STK1RC34,) then it will look like the swap was influenced because it matched the MVCATTR but in reality it did this by defaul.

Okay, It's a Bad MVC...

...but how bad? Let's say you ran through the check list for RTD problems, above, and that's not the problem. 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...no problem, the MVCs actually were in the in the ACS, otherwise you would have either reentered or replaced any MVCs not listed on the HSC Volume Report.

So it really *does* look like a media problem. You'll see what kind of media problem via the Wellness Meters described in "Checking Nearline Tape Status (Daily)" on page 8...and in that section, we've talked about some of the fixes for the most straightforward MVC anomalies. The following is an exhaustive list of the MVC statuses you **don't** want to see on your Wellness Meters, 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 will probably fail. 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" on page 29. 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 via 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 will 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 will 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 will recall from, but 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 will correct 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, you simply need to read the MVC to the end of the tape, which can be done via a VTCS audit. If the media is the problem, run an MVCDRAIN EJECT to recall the VTVs and cause the MVC to be removed from the MVC pool.

▼ Recovering an MVC with a Data Check

This is a very specific subspecies of the general “bad MVC” woes, and you know it’s required when you see an MVC data check error on your Wellness Meters.

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. HSC EJECT the defective MVC.

   b. HSC ENTER the replacement MVC.

   c. Initialize the tape as required.

   d. HSC AUDIT the new MVC.

   e. Run an MVCDRAIN (no EJECT) to return the MVC to the MVC pool.

   - Remove the MVC from the system:

   a. HSC EJECT 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.

Recovering the CDS

We're clearly out of the realm of fixing common problems. Losing some or all of the CDS is a *serious* occurrence, so it has its own private space. The are the standard recovery procedures, but if you really are in one of these situations, before doing anything, you probably should talk to support as described in "Customer Support" on page vii.

- If you lose the primary CDS and are recovering from a backup copy, make sure to use the most recent copy.
- If you have lost all copies of the CDS, use the procedure described in "Recovering from Losing All Copies of the CDS" on page 60.
- If the primary CDS is intact but you lost information about some VSM resources, use the procedure described in "Recovering from Losing Information about Some VSM Resources" on page 61.

  For example, you would use this procedure if VTCS and HSC abended while applications were writing VTVs to the VTSS and VTCS was doing migrate, recall, and/or reclaim operations. In this situation, VTV and MVC reports can help identify the lost information.

  You would also use this procedure if the VTSS lost power. In this situation, the SYSLOG can help identify any MVCs mounted at the time of the power outage.
- A VSM audit of all MVCs will audit MVCs defined with CONFIG MVCVOL statements even if they are *not* specified in the MVCPool statements. If these volumes are not in an ACS that VTCS accesses, HSC issues message SLS2126I to prompt the operator to enter these volumes. If the volume does not exist or is not available, the operator should reply "I" (ignore). If the operator does not reply to this message, VTCS times out with this MVC after 20 minutes and continues with the next MVC.
- **Note that** VSM does not support copies of the CDS at multiple sites (for example, Primary CDS at one site and Secondary at another). A link failure would allow the two sites to run independently, and VSM cannot enforce separation of all resources. This prevents reconciliation of the two divergent CDSs as can be accomplished in a pure NCS environment.
- If you audit an export MVC, VTCS processes the VTVs on the MVC as follows:
  - VTVs have an MVC1 and MVC2 field in the CDS. If the VTV's time and date stamp matches the CDS and the MVC1 and MVC2 fields are empty (the VTV has not been migrated), VTCS marks the audited VTV as current, fills in the empty MVC1 or MVC2 field, marks the MVC as read-only and as a consolidation MVC, and issues warning messages.
  - If the VTV's time and data stamp matches the CDS and the MVC1 and MVC2 fields are filled in, VTCS ignores this VTV copy. If all VTVs on the MVC are in this state, VTCS marks the MVC as empty and read-only, and issues warning messages.
  - If the VTV's time and date stamp does not match the CDS, VTCS ignores this VTV copy.
- When HSC and VSM audits complete, run the HSC scratch conversion utility (SLUCONDB) or the ExLM SYNCVTV function. Also ensure that the CDS is backed up at this point.
- *Do not* use VTVs to back up the CDS!

Note – A word about VTCS audits: When you run the AUDIT, VTCS splits the work into multiple audit subtasks which can use all available RTDs. The audit subtasks compete for RTDs with other VTCS tasks, such as recalls. When an audit subtask for an individual MVC completes, the RTD becomes available for other tasks, such as recalls or other audit subtasks.

Note, however, that VTCS will only run one AUDIT batch job at a time. If you submit multiple AUDIT batch jobs, all audit tasks for jobs after the first job submitted are queued behind tasks for the first job.

▼ Recovering from Losing All Copies of the CDS

Use the following procedure to recover a VSM system after losing all copies of the CDS.

To recover after losing all copies of the HSC CDS and update the MVC and VTV information, do the following:

1. **To recover the HSC CDS, audit the library.**

   For more information about the HSC AUDIT utility, see Chapter 5, “Utility Functions” in the *HSC System Programmer’s Guide for MVS.*

2. **Update your system’s MVC pool definition to load a null MVC pool.**

   A null MVC pool prevents migrations, consolidations, recalls, and space reclamations, which ensures that MVCs are not overwritten during the audit.

3. **Prevent applications from writing data to or reading data from existing VTVs during the audits.**

   Applications can write new VTVs to the VTSSs if the TMS is correct and sufficient VTSS space exists.

4. **To update the recovered CDS with your VSM system’s MVC and VTV information, do a full VSM audit.**

   For example, FIGURE 4-1 shows example JCL to run AUDIT for your entire VSM system.

```
//AUDIT    EXEC PGM=SWSADMIN
//STEPLIB    DD DSN=hlq.SLSLINK,DISP=SHR
//SLSPRINT   DD SYSOUT=*
//SLSIN      DD *
  AUDIT ALL
```

FIGURE 4-1 Example JCL for the AUDIT utility

   When VTCS encounters VTVs with the same volsers, VTCS determines which volume is the most current and uses that VTV. VTCS marks not current the other copies of the VTV.

5. **Update the scratch status of your VSM system’s MVCs and VTVs.**

   After the VTCS audit, VTCS assumes that all VTVs are non-scratch. After the VTCS audit completes, run the HSC scratch conversion utility or the ExLM SYNCVTV function.

6. **Rerun any consolidation jobs that may have been impacted by the loss of the CDS.**

Recovering from Losing Information about Some VSM Resources

Use the following procedure to recover after losing information about some VSM resources.

To recover after losing information about some VSM resources, do the following:

1. **If necessary, recover information about HSC resources by running an HSC audit.**

   For more information about the HSC AUDIT utility, see *HSC System Programmer's Guide for MVS.*

2. **Identify which VTVs and MVCs have incorrect information in the CDS.**

   If VTCS and HSC abended while applications were writing VTVs to the VTSS and VTCS was doing migrate, consolidate, recall, and/or reclaim operations, VTV and MVC reports can help identify the lost information. Compare these reports with the most current previous detailed reports.

   If the VTSS lost power, the SYSLOG can help identify any MVCs mounted at the time of the power outage.

3. **Update your system's MVC pool definition to load a pool that excludes any MVCs with incorrect information in the CDS.**

   The updated pool definition prevents migrations, recalls, and space reclamations for these MVCs to ensure they are not overwritten during the audit.

4. **Prevent applications from writing data to or reading data from existing VTVs during the audits.**

   Applications can write new VTVs to the VTSSs if the TMS is correct and sufficient VTSS space exists.

5. **To update the CDS, specify the MVCs and VTSSs for the VSM audit.**

6. **Update the scratch status of your VSM system's MVCs and VTVs.**

   After the VTCS audit, VTCS assumes that all VTVs are non-scratch. After the VTCS audit completes, run the HSC scratch conversion utility or the ExLM SYNCVTV function.

7. **Rerun any consolidation jobs that may have been impacted by the lost resource information.**

Using the RTV Utility

The RTV utility is another item you're probably only going to use after talking with StorageTek 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. RTV utility is included on the VTCS product tape and is also available for download from the StorageTek Customer Resource Center (CRC). Both versions are identical.

If you downloaded the RTV utility from the CRC for use at a site that does not have NCS/VTCS software installed, you must do the following:

- Install the downloaded version of RTV per instructions on the CRC.
- APF authorize the RTV load library you just created.
- Substitute the DSN you picked for the RTV load library for DSN hlq.SLSLINK on the STEPLIB DD statement(s).

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 will convert 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 BLOCKid or FILEnum parameter is specified, the entire MVC will be 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 will fail.
- 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

Listing the VTVs on an MVC

FIGURE 4-2 shows example JCL to lists the VTVs on MVC MVC001.

```
//JOBVREC     JOB(account),programmer
//RUNRTV      EXEC PGM=SWSRTV,PARM='MIXED'
//STEPLIB     DD DSN=hlq.SLSLINK,DISP=SHR
//SLSPRINT    DD SYSOUT=A
//SLSIN       DD *
  RTV  MVC(MVC001)INUNIT(/1AB4) LISTONLY
/*
//
```

FIGURE 4-2 Example JCL to run the RTV utility: LISTONLY run

Converting a Single VTV by Specifying Its Volser

FIGURE 4-3 shows example JCL to run the RTV utility to convert VTV VTV200 on MVC MVC001, which will be 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.

```
//JOBVREC     JOB(account),programmer
//RUNRTV      EXEC PGM=SWSRTV,PARM='MIXED'
//STEPLIB     DD DSN=hlq.SLSLINK,DISP=SHR
//SLSPRINT    DD SYSOUT=A
//SLSIN       DD *
  RTV  MVC(MVC001) INUNIT(3490E) VTV(VTV200) CPYVOLID
OUTUNIT(280)
/*
//
```

FIGURE 4-3 Example JCL to run the RTV utility: single VTV by volser

Converting a Single VTV by Specifying Its Volser and Block ID

FIGURE 4-4 shows example JCL to run the RTV utility to convert VTV VTV200 at block ID x’8EA484AB’ on MVC MVC001, which will be mounted on a 3490E transport. The output (converted VTV VTV200) goes to the output volume mounted on transport 480.

```
//JOBVREC     JOB(account),programmer
//RUNRTV      EXEC PGM=SWSRTV,PARM=’MIXED’
//STEPLIB     DD DSN=hlq.SLSLINK,DISP=SHR
//SLSPRINT    DD SYSOUT=A
//SLSIN       DD *
  RTV  MVC(MVC001) INUNIT(3490E) VTV(VTV200) BLOCK(8EA484AB)
OUTUNIT(480)
/*
```

FIGURE 4-4 Example JCL to run the RTV utility: single VTV by volser and block ID

APPENDIX **A**

ExLM 6.2 Control Statements

TABLE A-1 ExLM 6.2 Control Statements

ExLM Control Statement	6.2 Status	Corresponding HSC or VTCS Utility
ACTION CONSOLIDATE	Enhanced	SWSADMIN CONSOLIDATE
ACTION EJECT	Enhanced	SLUADMIN EJECT
ACTION EXPORT	Enhanced	SWSADMIN EXPORT
ACTION MIGRATE	Enhanced	SWSADMIN MIGRATE
ACTION MOVE	Enhanced	SLUADMIN MOVE
ACTION RECALL	Enhanced	SWSADMIN RECALL
ACTION RECLAIM	New	SWSADMIN RECLAIM
ACTION SCRATCH	New	SLUADMIN SCRATCH
ACTION UNSCRATCH	New	SLUADMIN UNSCRATCH, SLUADMIN REPLACEALL
DATASET	Enhanced	
LOCATION	Unchanged	
MANAGE PHYSICAL	Unchanged	
MANAGE VIRTUAL	Unchanged	
METHOD	Enhanced	
OPTIONS	Enhanced	
OPTIONS SYNC and SYNCVTV MANAGE PHYSICAL and MANAGE VIRTUAL	Unchanged	SLUCONDB (and SMC equivalent)
PULLLIST	Unchanged	
REPORT CELLCNT	Unchanged	
REPORT CONSOLIDATE	Unchanged	

TABLE A-1 ExLM 6.2 Control Statements

ExLM Control Statement	6.2 Status	Corresponding HSC or VTCS Utility
REPORT DATASET	Unchanged	
REPORT EJECT	Unchanged	
REPORT ENTER	Unchanged	
REPORT EXPORT	Unchanged	
REPORT LSM	Enhanced	
REPORT MIGRATE	Unchanged	
REPORT MULTIPLE	New	SWSADMIN MVCRPT
REPORT NONSCRCNT	Unchanged	
REPORT OPERATOR	Unchanged	
REPORT PHYSICAL	New	SLUADMIN VOLRPT
REPORT RECALL	Unchanged	
REPORT SCRCNT	Unchanged	
REPORT SUMMARY	Unchanged	
REPORT VIRTUAL	New	SWSADMIN VTVRPT
REPORT VOLUME	Enhanced	
SET METHOD	Enhanced	
SUBPOOL	Enhanced	
SUBPOOL and MANAGE PHYSICAL BALSCR	Unchanged	SLUADMIN SCRATCH REDISTRIBUTION
TMS CA1	Enhanced	
TMS COMMON	Enhanced	
TMS CTT	Enhanced	
TMS CUSTOM	Enhanced	
TMS OPEN	Enhanced	
TMS RMM	Enhanced	
TMS TLMS	Enhanced	
UNMANAGED	Enhanced	

APPENDIX **B**

Additional Information

Oracle Corporation (Oracle) offers several methods for you to obtain additional information.

Oracle’s External Web Site

Oracle’s external Web site provides marketing, product, event, corporate, and service information. The external Web site is accessible to anyone with a Web browser and an Internet connection. The URL for the Oracle external Web site is: http://www.oracle.com/us/index.html

The URL for Oracle/StorageTek/Sun storage information for is:
http://www.oracle.com/us/products/servers-storage/storage/index.html

Sun/StorageTek Documentation

The URL for Sun/StorageTek documentation is:

http://docs.sun.com/app/docs

Oracle Global Partners

The Oracle Global Partners site provides information about solutions available with Oracle’s partners:

http://www.oracle.com/us/partnerships/index.html

Third-Party Web Sites

Oracle is not responsible for the availability of third-party web sites mentioned in this document. Oracle does not endorse and is not responsible or liable for any content, advertising, products, or other materials that are available on or through such sites or resources. Oracle will not be responsible or liable for any actual or alleged damage or loss caused by or in connection with the use of or reliance on any such content, goods, or services that are available on or through such sites or resources.

Oracle’s Global Offices

You may contact any of Oracle’s worldwide offices to discuss complete storage, service, and support solutions for your organization. You can find contact information at:
http://www.oracle.com/corporate/contact/global.html

Customer Support

For more information about Oracle support (including for Sun/StorageTek branded products) see:
http://www.oracle.com/us/support/index.htm?origref=
http://www.oracle.com/us/products/index.html

Conventions for Reader Usability

Conventions are used to shorten and clarify explanations and examples within this book.

Typographic

The following typographical conventions are used in this book:

- **Bold** is used to introduce new or unfamiliar terminology.
- Letter Gothic is used to indicate command names, filenames, and literal output by the computer.
- Letter Gothic Bold is used to indicate literal input to the computer.
- *Letter Gothic Italic* is used to indicate that you must substitute the actual value for a command parameter. In the following example, you would substitute your name for the "username" parameter.
- Logon *username*
- A bar (|) is used to separate alternative parameter values. In the example shown below either username or systemname must be entered.
- Logon *username|systemname*
- Brackets [ ] are used to indicate that a command parameter is optional.
- Ellipses (...) are used to indicate that a command may be repeated multiple times.
- The use of mixed upper and lower case characters (for non–case sensitive commands) indicates that lower case letters may be omitted to form abbreviations. For example, you may simply enter **Q** when executing the **Quit** command.

Keys

Single keystrokes are represented by double brackets [[ ]] surrounding the key name. For example, press [[ESC]] indicates that you should press only the escape key.

Combined keystrokes use double brackets and the plus sign (+). The double brackets surround the key names and the plus sign is used to add the second keystroke. For example, press [[AL]] + [[C]] indicates that you should press the alternate key and the C key simultaneously.

Enter Command

The instruction to "press the [[ENTER]] key" is omitted from most examples, definitions, and explanations in this book.

For example, if the instructions asked you to "enter" **Logon pat**, you would type in **Logon pat** *and* press [[ENTER]].

However, if the instructions asked you to "type" **Logon pat**, you would type in **Logon pat** and you would *not* press [[ENTER]].

Warnings, Cautions, and Notes - Software

The following are used in software documentation.

Caution – Information necessary to keep you from corrupting your data.

Tip – Information that can be used to shorten or simplify your task or they may simply be used as a reminder.

Note – Information that may be of special interest to you. Notes are also used to point out exceptions to rules or procedures.

Warnings, Cautions, and Notes - Hardware

The following are used in hardware documentation.

Note – A note provides additional information that is of special interest. A note might point out exceptions to rules or procedures. A note usually, but not always, follows the information to which it pertains.

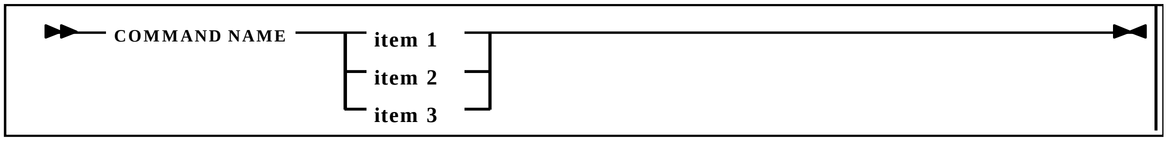
⚠ **Caution –** A caution informs you of conditions that might result in damage to hardware, corruption of data, or corruption of application software. A caution always precedes the information to which it pertains.

⚠ **Warning – Possible Physical Injury. A warning alerts you to conditions that might result in long-term health problems, injury, or death. A warning always precedes the information to which it pertains.**

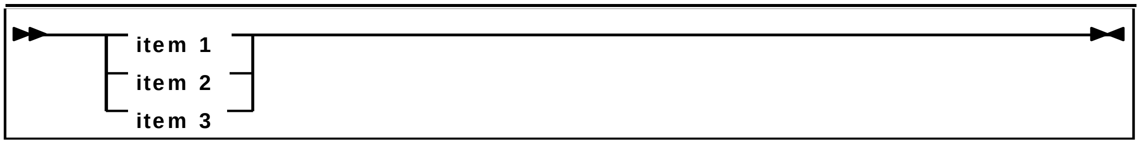
Syntax

Syntax flow diagram conventions include the following:

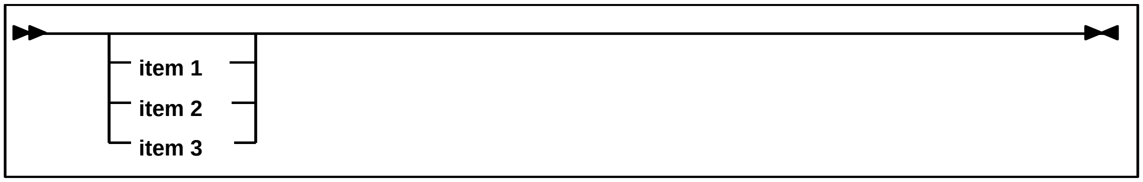
Flow Lines—Syntax diagrams consist of a horizontal baseline, horizontal and vertical branch lines and the command text. Diagrams are read left to right and top to bottom. Arrows show flow and direction.



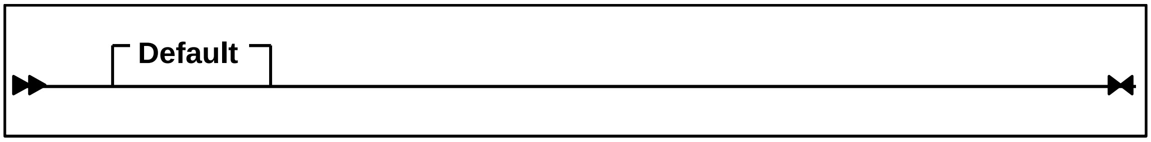
Single Required Choice—Branch lines (without repeat arrows) indicate that a single choice must be made. If one of the items to choose from is on the baseline of the diagram, one item must be selected.



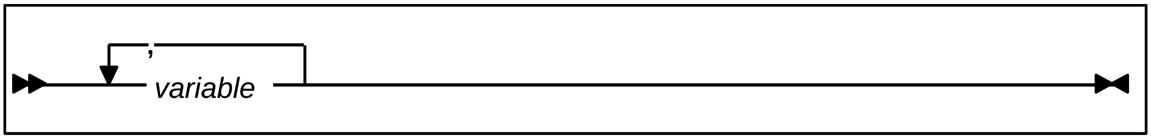
Single Optional Choice—If the first item is on the line below the baseline, one item may optionally be selected.



Defaults—Default values and parameters appear above the baseline.



Repeat Symbol—A repeat symbol indicates that more than one choice can be made or that a single choice can be made more than once. The repeat symbol shown in the following example indicates that a comma is required as the repeat separator.



Keywords—All command keywords are shown in all upper case or in mixed case. When commands are not case sensitive, mixed case implies that the lowercase letters may be omitted to form an abbreviation.

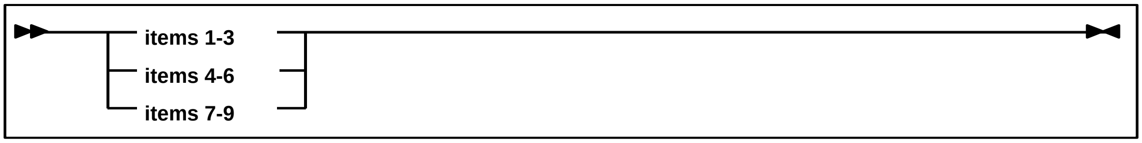
Variables—Italic type is used to indicate a variable.

Alternatives—A bar (|) is used to separate alternative parameter values.

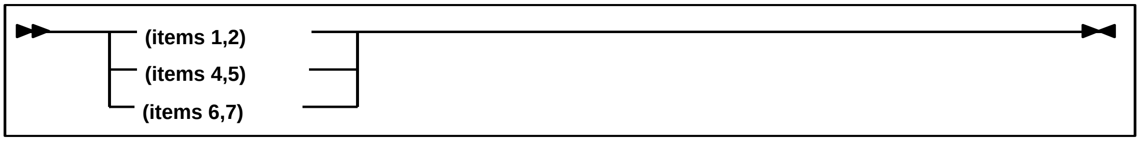
Optional—Brackets [ ] are used to indicate that a command parameter is optional.

Delimiters—If a comma (,), a semicolon (;), or other delimiter is shown with an element of the syntax diagram, it must be entered as part of the statement or command.

Ranges—An inclusive range is indicated by a pair of elements of the same length and data type, joined by a dash. The first element must be strictly less than the second element.



Lists—A list consists of one or more elements. If more than one element is specified, the elements must be separated by a comma or a blank and the entire line must be enclosed by parentheses.



Index