# StorageTek™ Virtual Tape Control System

Software

## Beyond the Basics: VTCS Leading Edge Techniques

CRC Update Only
Revision C
Version 6.2

# VTCS™

## Beyond the Basics: VTCS Leading Edge Techniques

PN CRC Update Only

Version 6.2

Sun Microsystems, Inc.
www.sun.com

CRC Update Only
February 2008

Submit comments about this document to: SLSFS@Sun.com

---

We welcome your feedback. Please contact the Sun Learning Services Feedback System at:

SLSFS@Sun.com

or

Sun Learning Services
Sun Microsystems, Inc.
One StorageTek Drive
Louisville, CO 80028-3256
USA

Please include the publication name, part number, and edition number in your correspondence if they are available. This will expedite our response.

Please Recycle

Adobe PostScript

Revision C • CRC Update Only

# Revision History

| EC | Date | Description |
|---|---|---|
| 132864 | March 2007 | Revision A |
| --- | February 2008 | Revision C |

# Notices

Please read the following compliance and warning statements for this product.

**Caution –** *Potential equipment damage:* Cables that connect peripherals must be shielded and grounded; refer to descriptions in the cable instruction manuals. Operation of this equipment with cables that are not shielded and not correctly grounded might result in interference to radio and TV reception.

Changes or modifications to this equipment that are not expressly approved in advance by StorageTek will void the warranty. In addition, changes or modifications to this equipment might cause it to create harmful interference.

## United States FCC Compliance Statement

The following compliance statement pertains to Federal Communications Commission Rules 47 CFR 15.105:

**Note –** This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his or her own expense.

## CISPR 22 and EN55022 Warning

This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

## Japanese Compliance Statement

The following compliance statement in Japanese pertains to VCCI EMI regulations:

この装置は、情報処理装置等電波障害自主規制協議会（ＶＣＣＩ）の基準に基づくクラスＡ情報技術装置です。この装置を家庭環境で使用すると電波妨害を引き起こすことがあります。この場合には使用者が適切な対策を講ずるよう要求されることがあります。

**English translation**: This is a Class A product based on the Technical Requirement of the Voluntary Control Council for Interference by Information Technology (VCCI). In a domestic environment, this product may cause radio interference, in which case the user may be required to take corrective actions.

## Taiwan Warning Label Statement

The following warning label statement pertains to BSMI regulations in Taiwan, R.O.C.:

警告使用者：這是甲類的資訊產品，在居住的環境中使用時，可能會造成射頻干擾，在這種情況下，使用者會被要求採取某些適當的對策

**English translation**: This is a Class A product. In a domestic environment, this product may cause radio interference, in which case, the user may be required to take adequate measures.

## Internal Code License Statement

The following is the Internal Code License Agreement from StorageTek:

The following is the Internal Code License Agreement from StorageTek:

**NOTICE**

INTERNAL CODE LICENSE

PLEASE READ THIS NOTICE CAREFULLY BEFORE INSTALLING AND OPERATING THIS EQUIPMENT. THIS NOTICE IS A LEGAL AGREEMENT BETWEEN YOU (EITHER AN INDIVIDUAL OR ENTITY), THE END USER, AND STORAGE TECHNOLOGY CORPORATION ("STORAGETEK"), THE MANUFACTURER OF THE EQUIPMENT. BY OPENING THE PACKAGE AND ACCEPTING AND USING ANY UNIT OF EQUIPMENT DESCRIBED IN THIS DOCUMENT, YOU AGREE TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT. IF YOU DO NOT AGREE WITH THE TERMS OF THIS AGREEMENT, DO **NOT** OPEN THE PACKAGE AND USE THE EQUIPMENT. IF YOU DO NOT HAVE THE AUTHORITY TO BIND YOUR COMPANY, DO **NOT** OPEN THE PACKAGE AND USE THE EQUIPMENT. IF YOU HAVE ANY QUESTIONS, CONTACT THE AUTHORIZED STORAGETEK DISTRIBUTOR OR RESELLER FROM WHOM YOU ACQUIRED THIS EQUIPMENT. IF THE EQUIPMENT WAS OBTAINED BY YOU DIRECTLY FROM STORAGETEK, CONTACT YOUR STORAGETEK REPRESENTATIVE.

1. **Definitions:** The following terms are defined as follows:
   a. "Derivative works" are defined as works based upon one or more preexisting works, such as a translation or a musical arrangement, or any other form in which a work may be recast, transformed, or adapted. A work consisting of editorial revision, annotations, elaboration, or other modifications which, as a whole, represent an original work of authorship, is a Derivative work.
   b. "Internal Code" is Microcode that (i) is an integral part of Equipment, (ii) is required by such Equipment to perform its data storage and retrieval functions, and (iii) executes below the user interface of such Equipment. Internal code does not include other Microcode or software, including data files, which may reside or execute in or be used by or in connection with such Equipment, including, without limitation, Maintenance Code.
   c. "Maintenance Code" is defined as Microcode and other software, including data files, which may reside or execute in or be used by or in connection with Equipment, and which detects, records, displays, and/or analyzes malfunctions in the Equipment.
   d. "Microcode" is defined as a set of instructions (software) that is either imbedded into or is to be loaded into the Equipment and executes below the external user interface of such Equipment. Microcode includes both Internal Code and Maintenance Code, and may be in magnetic or other storage media, integrated circuitry, or other media.
2. The Equipment you have acquired by purchase or lease is manufactured by or for StorageTek and contains Microcode. By accepting and operating this Equipment, you acknowledge that StorageTek or its licensor(s) retain(s) ownership of all Microcode, as well as all copies thereof, that may execute in or be used in the operation or servicing of the Equipment and that such Microcode is copyrighted by StorageTek or its licensor(s).
3. StorageTek hereby grants you, the end user of the Equipment, a personal, nontransferable (except as permitted in the transfer terms below), nonexclusive license to use each copy of the Internal Code (or any replacement provided by StorageTek or your authorized StorageTek distributor or reseller) which license authorizes you, the end user, to execute the Internal Code solely to enable the specific unit of Equipment for which the copy of Internal Code is provided to perform its data storage and retrieval functions in accordance with StorageTek's (or its licensor's) official published specifications.
4. Your license is limited to the use of the Internal Code as set forth. You may not use the Internal Code for any other purpose. You may not, for example, do any of the following:

   (i) access, copy, display, print, adapt, alter, modify, patch, prepare Derivative works of, transfer, or distribute (electronically or otherwise) or otherwise use the Internal Code;

   (ii) reverse assemble, decode, translate, decompile, or otherwise reverse engineer the Internal Code (except as decompilation may be expressly permitted under applicable European law solely for the purpose of gaining information that will allow interoperability when such information is not otherwise readily available); or

   (iii) sublicense, assign, or lease the Internal Code or permit another person to use such Internal Code, or any copy of it.

5. Nothing in the license set forth above or in this entire Notice shall convey, in any manner, to you any license to or title to or other right to use any Maintenance code, or any copy of such Maintenance Code. Maintenance Code and StorageTek's service tools and manuals may be kept at your premises, or they may be supplied with a unit of Equipment sent to you and/or included on the same media as Internal Code, but they are to be used only by StorageTek's customer service personnel or those of an entity licensed by StorageTek, all rights in and to such Maintenance Code, service tools and manuals being reserved by StorageTek or its licensors. You agree that you shall not use or attempt to use the Maintenance Code or permit any other third party to use and access such Maintenance Code.
6. You, the end user, agree to take all appropriate steps to ensure that all of your obligations set forth in this Notice are extended to any third party having access to the Equipment.
7. You may transfer possession of the Internal Code to another party only with the transfer of the Equipment on which its use is authorized, and your license to use the Internal Code is discontinued when you are no longer an owner or a rightful possessor of the Equipment. You must give such transferee all copies of the Internal Code for the transferred Equipment that are in your possession, along with a copy of all provisions of this Notice.

   Any such transfer by you is automatically (without further action on the part of either party) expressly subject to all the terms and conditions of this Notice passing in full to the party to whom such Equipment is transferred, and such transferee accepts the provisions of this license by initial use of the Internal Code. You cannot pass to the transferee of the Equipment any greater rights than granted under this Notice, and shall hold StorageTek harmless from any claim to the contrary by your transferee or its successors or assigns. In addition, the terms and conditions of this Notice apply to any copies of Internal Code now in your possession or use or which you hereafter acquire from either StorageTek or another party.
8. You acknowledge that copies of both Internal Code and Maintenance Code may be installed on the Equipment before shipment or included with the Equipment and other material shipped to you, all for the convenience of StorageTek's service personnel or service providers licensed by StorageTek, and that during the warranty period, if any, associated with the Equipment, and during periods in which the Equipment is covered under a maintenance contract with StorageTek or service providers licensed by StorageTek, both Internal Code and Maintenance Code may reside and be executed in or used in connection with such Equipment, and you agree that no rights to Maintenance Code are conferred upon you by such facts.

   StorageTek or the licensed service provider may keep Maintenance Code and service tools and manuals on your premises but they are to be used only by StorageTek's customer service personnel or those of service providers licensed by StorageTek. You further agree that upon
   (i) any termination of such warranty period or maintenance contract period; or
   (ii) transfer of possession of the Equipment to another party, StorageTek and its authorized service providers shall have the right with respect to the affected Equipment to remove all service tools and manuals and to remove or disable all Maintenance Code and/or replace Microcode which includes both Internal Code and Maintenance Code with Microcode that consists only of Internal Code.

# Preface

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 installing configuring VTCS. See *VTCS Command and Utility Reference* for information about the following:

- VTCS and NCS (virtual) commands and utilities
- HSC SMF records for VTCS
- VTD commands

### 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 is a new book for VTCS 6.2, and is designed to take you beyond the basics of installation, configuration, and day-to-day operation. This is where your VSM system really shows its value add, to whit:

- "The Advanced Management Feature and VTCS CDS Levels" on page 1 is about the Advanced Management Feature, which you pretty much have to have to get things done in this book.
- "Using Management and Storage Classes" on page 3. Management and Storage Classes are the bread and butter of anything leading edge in VTCS land.
- "Using Clustered VTSS Configurations" on page 29. Clustered VTSS is a powerful tool, and here's why and how to use it.
- "Merging and Consolidating Data Centers" on page 61, which you do via the EXPORT and IMPORT utilities.

Enjoy...

# What's New in this Guide?

## VTCS 6.2.0, Revision C

The VTCS 6.2.0, Revision C of this guide contains information about the VTCS 6.2 enhancements described in TABLE P-1.

**TABLE P-1** VTCS 6.2.0 Updates to VTCS Leading Edge Techniques, Revision C

| This Enhancement... | ...is described in... |
|---|---|
| VTSS Synchronous Replication | "Another Variation: Synchronous or Asynchronous Replication?" on page 42 |

Synchronous replication, which applies to only VSM4s and VSM5s, has the requirements described in TABLE P-2.

**TABLE P-2** Sychronous Replication Requirements for VTCS/NCS 6.2

| Synchronous replication requires... | ..the following VSM4/VSM5 microcode... | ...and the following VTCS/NCS 6.2 PTFs... | ...and CDS level... |
|---|---|---|---|
| FICON ports for the CLINKs | D02.03.00.00 or higher | L1H13QL (SWS6200), L1A00L3 (SMC6200), and L1H13K8 (SOS6200) | "F" or higher |

## VTCS 6.2.0, Revision B

The VTCS 6.2.0, Revision B of this guide contains technical updates and corrections.

# Contents

CHAPTER 1

# The Advanced Management Feature and VTCS CDS Levels

This is a quick commercial for your toolkit, focusing on two essential items you need for advanced projects. Any discussion of VTCS Leading Edge work starts with the The Advanced Management Feature, which is an optional VTCS feature, but pretty much required if you want to do anything, well, *advanced*, as shown in TABLE 1-1.

**TABLE 1-1** Features Enabled by the Advanced Management Feature

| For this feature... | ...use this interface... | ...which is described in... |
|---|---|---|
| Management and Storage Classes | STORclas statement<br>MGMTclas statement MIGpol, RESTIME, CONSRC, CONTGT, REPLICAT, ARCHAGE, and ARCHPOL parameters | "Using Management and Storage Classes" on page 3 |
| Export/Import | EXPORT command/utility<br>IMPORT command/utility | "Merging and Consolidating Data Centers" on page 61 |
| Clustered VTSS configurations | MGMTclas statement REPLICAT parameter<br>CONFIG CLUSTER, CONFIG CLINK statements<br>MGMTclas and STORclas statements (typical but not required) | "Using Clustered VTSS Configurations" on page 29 |

So, the moral is, if you want to do fun stuff, sign up for the Advanced Management Feature and make sure you enable it via the HSC FEATURES control statement.

The next message is “know your CDS VTCS level, and what it can and cannot do.” In , we provided this handy chart of CDS levels and the features they enable, which we’ll repeat here.

**TABLE 1-2** CDS Levels for Supported VTCS Versions

| This VTCS CDS Level... | ...is valid for these VTCS/NCS versions... | ...and this VTSS hardware... | ...and provides these enhancements |
|---|---|---|---|
| E | 6.0, 6.1, 6.2 | ■ VSM2 and VSM3<br>■ VSM4 with up to 256 VTDs per VTSS and/or up to 16 RTDs per VTSS.<br>■ RTD sharing **except** for paired RTDs (a paired RTD shares a CIP with another Nearlink connection, either an RTD or a CLINK). | ■ 4 MVC copies<br>■ 800 Mb VTVs (see for additional requirements) |
| F | 6.1, 6.2 | | ■ Near Continuous Operations (NCO)<br>■ Bi-directional clustering |
| G | 6.2 | | ■ 400Mb/800Mb/2Gb/4Gb VTVs<br>■ Standard/Large VTV Pages<br>■ 65000 VTVs per MVC |

CHAPTER **2**

# Using Management and Storage Classes

VTCS Management and Storage Classes, which are the bread and butter of a lot of leading edge implementations, do the following:

- VTCS Management Classes specify *how* VTCS manages VTVs. The HSC MGMTclas control statement defines a Management Class and its attributes. For example, the DELSCR parameter of the MGMTclas statement specifies whether VTCS deletes scratched VTVs from the VTSS. Management Classes can also point to...
- ...VTCS Storage Classes, which specify *where* migrated VTVs reside. The HSC STORclas control statement defines a Storage Class and its attributes. For example:

```
MGMT NAME(PAYROLL) MIGPOL(LOCAC,REMAC)
STORCLAS NAME(LOCAC) ACS(00) MEDIA(STK1R)
STORCLAS NAME(REMAC) ACS(01) MEDIA(STK2P,ZCART)
```

  This combination of Management and Storage Classes says "For Management Class PAYROLL, migrate duplexed to separate MVCs in the local and remote ACSs. In the local ACS, put it on 9840 media so I can get it back in a hurry if I need it. In the remote ACS, prefer 9940 over ZCART media, but *definitely* put it in Deep Storage there."

See how easy that was? Okay, there *is* Some Assembly Required, so we'll get you walking before you try to run, starting with "Creating and Using VTCS Management and Storage Classes: The Basics" , which is a plain vanilla procedure you can cut to fit just about any of your shop's needs. Next, it's on to "Nine Cool Things You Can Do with Management and Storage Classes" . Think of this section as a sort of Parade of Homes where you get to see a variety of options, then pick out the one that works best for you...

---

**Tip –** Finally, there's one more topic that's a little off the subject, but fits in as well here as anywhere: "...And Now a Commercial Message: Named MVC Pools or Not?" .

---

# Creating and Using VTCS Management and Storage Classes: The Basics

You'll see this pattern a lot (TAPEREQ --> MGMTclas --> STORclas), so study it carefully, because it's the foundation for all the good stuff in "Nine Cool Things You Can Do with Management and Storage Classes" .

## ▼ To create and use VSM Management and Storage Classes:

1. **Enable the VSM Advanced Management Feature via the HSC FEATures control statement.**
2. **Determine the definition data set that contains the STORclas and MGMTclas statements.**

   MGMTclas and STORclas statements **must** reside in the same data set for cross-validation.
3. **Define Storage Classes via the STORclas control statement.**
4. **Define Management Classes with the MGMTclas control statement.**

   Note that the MGMTclas control statement specifies Storage Classes on the MIGpol, CONSRC, and CONTGT parameters.
5. **Load the MGMTclas and STORclas control statements with the HSC MGMTDEF command.**
6. **Specify the Management Class name to VTCS on any of the following:**
   - The SMC TAPEREQ statement.
   - SMS routines that you write to the StorageTek DFSMS interface.

## Maintaining Management and Storage Classes

**Note the following**:

- If you specify a Management Class on a TAPEREQ statement and an SMS routine, the Management Class on the SMS routine takes precedence.
- If VTCS receives a request to migrate a VTV that is assigned to an invalid Management Class, VTCS will dynamically create the !ERROR Storage Class and migrate the VTVs defined by the invalid Management Class to the !ERROR Storage Class. MVC reports show when a VTV is migrated to this Storage Class.
- You can use the VTVMAINT utility to change a VTV's Management Class. Also note that while you **cannot** use VTVMAINT to directly change a VTV's Storage Class, you **can** use VTVMAINT to change a VTV's Management Class, which can reference a different Storage Class.
- Use only the minimum Storage Classes required to define the policies you want to implement. Excessive Storage Classes can impact VSM performance due to the MVC mount/dismount overhead incurred. In addition, an MVC can only contain VTVs in a single Storage Class, so excessive Storage Classes can underuse MVC space.
- **If you decide to delete a Management Class definition**, run a VTV Report to make sure that the Management Class is no longer assigned to any VTVs, otherwise unpredictable results will occur!

# Nine Cool Things You Can Do with Management and Storage Classes

This is a work in progress, because right now it's titled "Nine Cool Things..." but I'm sure I'll come up with a whole lot more, so watch this space...and the current lineup is:

- "Using the STORclas MEDIA Parameter for MVC Media Preferencing" . Yes, we give you defaults for MVC media preferencing, but you can adjust them any way you want...
- "Grouping Multiple Workloads on Shared MVCs" . This was our introductory example, and is a good one if your company owns the data center, and you want to optimize use of your available resources by:
  - Duplexing critical data to separate MVCs in the local and remote ACSs. In the local ACS, put it on 9840 media so I can get it back in a hurry if I need it. In the remote ACS, prefer 9940 over ZCART media for deep storage on high-capacity media.
  - Giving two critical jobstreams (payroll and accounting) access to these Management/Storage Classes. Result: all your payroll and accounting data is duplexed local and remote, grouped on the same set of MVCs of the appropriate media per the Storage Class specifications.
  - Production data is also critical, but we want it one a separate set of MVCs from the ones used for payroll and accounting data. No problem...just create another Management Class/Storage Class combo for production data, and you're home free...
- "Segregating Individual Workloads on Separate Sets of MVCs" . All you service groups, study this one carefully, because it's right up your alley. Ever want to give each of your clients his/her own set of resources for billing/security purposes? Look no further...
- "Using VTSS Preferencing" . An interesting use of Management and Storage Classes, plus VTSS preferencing...
- "Archiving Data" . In "Specifying VTV Residency Interval before Automatic Migration Candidacy" we show an example where you can enable *both* high availability and deep archive for the same data. This is a variation on that theme, where you can use VTCS to mimic HSM...in tapeland only. That is, you can use the ARCHAge and ARCHPol parameters of the MGMTclas statement to set an *Archive Policy* for VTVs in a Management Class.

  This one probably applies to everybody out there. Information Lifecycle Management (ILM), a StorageTek storage management strategy, has as its central concept the idea that data should be stored on media that matches its importance to the business enterprise and its pattern of reuse. Active, important data goes on fast-access media, and has multiple copies, while inactive, less important data is archived to high-capacity, cheaper media. Automating this process is the most cost-effective way to manage data storage. Archiving implements ILM by letting you archive inactive data. Using VTCS archiving, you can move VTVs to different media (for example, from fast-access 9840 media to high-capacity 9940 media) and a different location (for example, from a local ACS to a remote ACS for ejection/vaulting). For the details, see "Archiving Data" ...

- “Reconciling VTV Media and Location” . Think of archiving as a proactive move. You put the data on the right media at the beginning of the ILM cycle, then move it to other media as the data ages. What happens if the data ends up on the wrong media? Answer: use the RECONcil utility to move it from one Storage Class to another...
- “Deleting Scratch VTVs” . Deleting scratch VTVs from VTSS buffer space is generally a Good Thing because it frees VTSS buffer space. The VTVs are still defined to VTCS and are eligible for scratch mounts, but no longer take up needed space in the VTSS.

  There are, of course, some *caveats* with this feature...namely, the data is gone at scratch synchronization time, so read the fine print. This is the first of several features that you implement with Management Class only...we’re *only* specifying how the VTV data is managed. Storage Classes aren’t required, although there is nothing that inhibits the Management Class from specifying a Storage Class for other purposes, and we will, in fact, further fine tune our delete-on-scratch example using a Storage Class.
- “Immediately Migrating VTVs On Dismount” . This is a handy tool for getting VTVs on tape ASAP, and you have the option of either deleting the VTV from VTSS space after a successful migration, or leaving it resident for automatic space management to do its thing.
- “Specifying VTV Residency Interval before Automatic Migration Candidacy” (MGMTclas RESTIME) is mutually exclusive with (IMMEDmig DELETE), and you can see why. Specifying both would be one foot on the brake, one on the gas. Specifying a long RESTIME is a way of preferencing high availability of the data in a Management Class.

## Using the STORclas MEDIA Parameter for MVC Media Preferencing

By default, in mixed-media VSM systems, VTV automatic and demand migrations (and consolidations) attempt to go to MVCs by media type in this order:

1. Standard length 3480 cartridge
2. 3490E cartridge
3. 3490EE cartridge
4. T9840A/B cartridge
5. T9840C cartridge
6. T9940A cartridge
7. T10000 sport cartridge
8. T9940B cartridge
9. T10000 full capacity cartridge

By default, for automatic and demand space reclamations, VSM attempts to write VTVs to output MVCs by media type in this order:

1. T10000 full capacity cartridge
2. T9940B cartridge
3. T10000 sport cartridge
4. T9940A cartridge
5. T9840C cartridge
6. T9840A/B cartridge
7. 3490EE cartridge
8. 3490E cartridge
9. Standard length 3480 cartridge

TABLE 2-1 describes each cartridge type, its corresponding `STORclas MEDIA` value, and where this cartridge media type is specified on the `VOLATTR` statement.

**TABLE 2-1** Valid Values for the MVC Media Parameter

| This cartridge type... | ...is specified by this STORclas MEDIA value... | ...which is specified on this VOLATTR parameter |
|---|---|---|
| standard length 3480 cartridge | STANDARD | MEDIA |
| 3490E cartridge | ECART | MEDIA |
| 3490EE cartridge | ZCART | MEDIA |
| T9840A or T9840B cartridge | STK1RAB | RECTECH |
| T9840C cartridge | STK1RC | RECTECH |

**TABLE 2-1** Valid Values for the MVC Media Parameter

| This cartridge type... | ...is specified by this STORclas MEDIA value... | ...which is specified on this VOLATTR parameter |
|---|---|---|
| T9940A cartridge | STK2PA | RECTECH |
| T9940B cartridge | STK2PB | RECTECH |
| T10000 full capacity cartridge | T10000T1 | MEDIA |
| T10000 sport cartridge | T10000TS | MEDIA |

The MEDIA parameter of the STORclas statement specifies a preference list of MVC media types. This list supersedes the default media selection list. **Note that** for reclamation, VTVs are written back to MVCs in the **reverse** of the order specified on the MEDIA parameter.

For example, if you specify the following on the MEDIA parameter of the STORclas statement...

```
MEDIA(STK1RAB,STK1RC,STK2PB)
```

- ...to select an MVC for migration to this Storage Class, VTCS searches for a usable MVC in the order STK1RAB, STK1RC, STK2PB.
- ...to select an MVC for the output of reclaim to this Storage Class, VTCS searches for a usable MVC in the order STK2PB, STK1RC, STK1RAB.

You can specify the media and ACS preferencing via the Storage Class(es) specified on the MIGpol parameter of the MGMTclas control statement.

To optimize recall processing in mixed-media systems, ensure that your MVC pool has at least one media type compatible with each RTD type.

## Grouping Multiple Workloads on Shared MVCs

You can use Storage and Management Classes to group multiple workloads on a shared set of MVCs. For example, the STORclas statements in FIGURE 2-1 define Storage Classes LOC1, LOC2, REM1, and REM2.

```
STORCLAS NAME(LOC1) ACS(00) MEDIA(STK1R)

STORCLAS NAME(LOC2) ACS(00) MEDIA(STK1R)

STORCLAS NAME(REM1) ACS(01) MEDIA(STK2P,ZCART)

STORCLAS NAME(REM2) ACS(01) MEDIA(STK2P,ZCART)
```

**FIGURE 2-1** Storage Classes for Workload Grouping

FIGURE 2-2 defines the following Management Classes:

- Management Classes PAY and ACCOUNT both specify the LOC1 and REM1 Storage Classes on the MIGPOL parameter. The VTVs in PAY and ACCOUNT, therefore, are duplexed and grouped on the MVCs defined by Storage Classes LOC1 and REM1.
- Management Class PROD specifies the LOC2 and REM2 Storage Classes on the MIGPOL parameter. The VTVs in PROD, therefore, are duplexed and grouped on the MVCs defined by Storage Classes LOC2 and REM2, which are separate from those for PAY and ACCOUNT.

```
MGMT NAME(PAY) MIGPOL(LOC1,REM1)

MGMT NAME(ACCOUNT) MIGPOL(LOC1,REM1)

MGMT NAME(PROD) MIGPOL(LOC2,REM2)
```

**FIGURE 2-2** Management Classes for Workload Grouping

FIGURE 2-3 consists of TAPEREQ statements that do the following:

- Data sets with qualifiers of PAYROLL.** are routed to VSM, Management Class PAY, and data sets with qualifiers of ACCOUNTS.** are routed to VSM, Management Class ACCOUNT. As shown in FIGURE 2-2, these Management Classes specify identical Storage Classes on the MIGPOL parameter, so all data sets with qualifiers of PAYROLL.** and ACCOUNTS.** are duplexed and grouped on the MVCs defined by Storage Classes LOC1 and REM1.
- All other data sets are routed to VSM, Management Class PROD; these data sets, therefore, are duplexed and grouped on the MVCs defined by Storage Classes LOC2 and REM2.

```
TAPEREQ DSN(PAYROLL.**) MEDIA(VIRTUAL) MGMT(PAY)

TAPEREQ DSN(ACCOUNTS.**) MEDIA(VIRTUAL) MGMT(ACCOUNT)

TAPEREQ DSN(**) MEDIA(VIRTUAL) MGMT(PROD)
```

**FIGURE 2-3** TAPEREQ Statements for Workload Grouping

Once an MVC is used for a Storage Class it remains exclusively assigned to that Storage Class while it contains current VTV copies. This grouping of VTVs on MVCs will be retained even after the MVCs undergo reclamation processing.

---

**Caution –** You cannot use the default Storage Class (the name of the last VTSS that wrote to the MVC for reclamation or migration) to group workloads.

---

## Segregating Individual Workloads on Separate Sets of MVCs

You can use Storage and Management Classes to segregate individual workloads on separate sets of MVCs. For example, the STORclas statements in FIGURE 2-4 define Storage Classes LOC, CUSTA, and CUSTB1, and CUSTB2.

```
STORCLAS NAME(LOC) ACS(00) MEDIA(STK1R)

STORCLAS NAME(CUSTA) ACS(00) MEDIA(STK1R)

STORCLAS NAME(CUSTB1) ACS(00) MEDIA(STK1R)

STORCLAS NAME(CUSTB2) ACS(01) MEDIA(STK2P)
```

**FIGURE 2-4** Storage Classes for Workload Segregation

FIGURE 2-5 defines the following Management Classes:

- Management Class CUSTA specifies the CUSTA Storage Class on the MIGPOL parameter. VTCS simplexes VTVs in this Management Classes to **only** the CUSTA Storage Class (9840 media in the local ACS), because that's what this customer wants.
- Customer B wants more protection, namely duplexing to the local and remote ACSs, so Management Class CUSTB points to **both** the CUSTB1 and CUSTB2 Storage Classes.
- Finally, local ACS/9840 media is just fine for your own production data, so that's what Management Class PROD does. What I'll probably also do is set up an Archive Policy for this Management Class (see "Archiving Data" ) so I can move it to Deep Storage eventually.

```
MGMT NAME(CUSTA) MIGPOL(CUSTA)

MGMT NAME(CUSTB) MIGPOL(CUSTB1,CUSTB2)

MGMT NAME(PROD) MIGPOL(LOC)
```

**FIGURE 2-5** Management Classes for Workload Segregation

FIGURE 2-6 shows the corresponding TAPEREQ statements:

- As shown in FIGURE 2-6, data sets with qualifiers of CUSTA.** are routed to VSM, Management Class CUSTA, data sets with qualifiers of CUSTB.** are routed to VSM, Management Class CUSTB, and all other data sets are routed to VSM, Management Class PROD. Thus, workload from the CUSTA.**, CUSTB.**, and all other data sets are effectively segregated.

```
TAPEREQ DSN(CUSTA.**) MEDIA(VIRTUAL) MGMT(CUSTA)

TAPEREQ DSN(CUSTB.**) MEDIA(VIRTUAL) MGMT(CUSTB)

TAPEREQ DSN(**) MEDIA(VIRTUAL) MGMT(PROD)
```

**FIGURE 2-6** TAPEREQ Statements for Workload Segregation

---

**Caution –** You cannot use the default Storage Class (the name of the last VTSS that wrote to the MVC for reclamation or migration) to segregate workloads.

---

## Using VTSS Preferencing

VTSS preferencing is a method to prioritize VTSS use for specified operations: migration, recall, drain, and so forth. Is this useful? Probably, especially if you have remote and local sites. You want the VTSS(s) at the remote site to do whatever work needs to be done for that site, and the VTSS(s) at the local site to be working on local stuff, and not vice versa. VTSS preferencing isn't simply a performance optimizer, but it's probably its most eye-opening use.

So let's set the stage: You have cross-connected MVS hosts as shown in FIGURE 2-7.

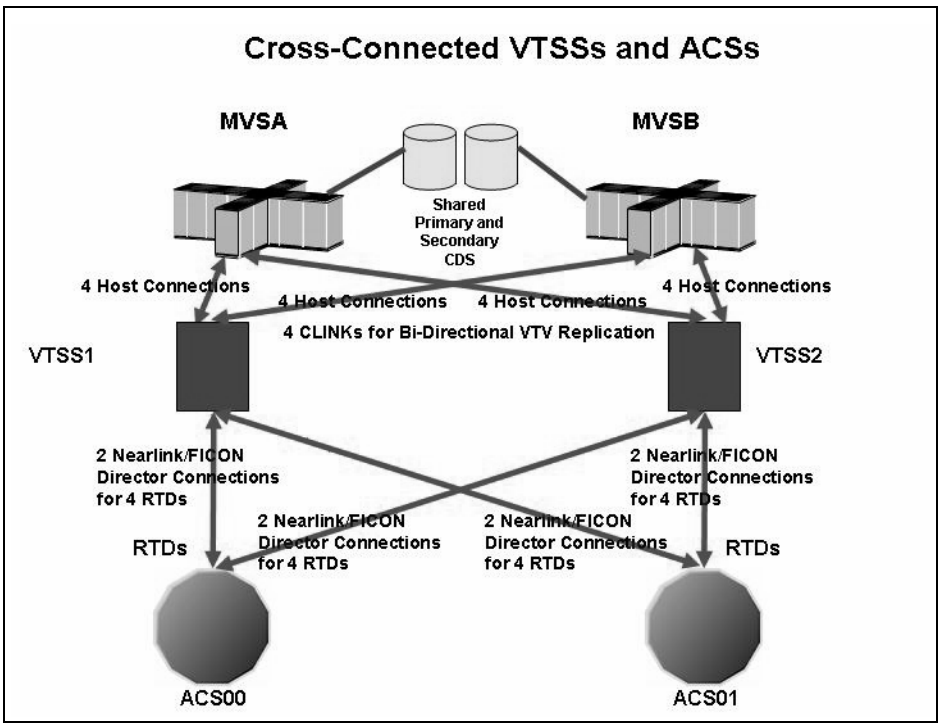



**FIGURE 2-7** Cross-Connected Hosts

Cross-connected is good, because anybody can get to anybody, if necessary. In general, however, you'd like VTSS2 to be doing things with ACS01 to optimize resource use in that site, and VTSS1 to be doing things with ACS00 for the same reason...especially for resource-intensive stuff like drain/reclaim. Here's how you do it:

```
STOR NAME(MVSA) ACS(00)
STOR NAME(MVSB) ACS(01)

VTSSLST NAME(LOCAL_A) VTSS(VTSS1)
VTSSLST NAME(LOCAL_B) VTSS(VTSS2)

VTSSEL FUNCTION(RECLAIM) STORCLAS(MVSA) VTSSLST(LOCAL_A)
VTSSEL FUNCTION(DRAIN) STORCLAS(MVSA) VTSSLST(LOCAL_A)
VTSSEL FUNCTION(RECLAIM) STORCLAS(MVSB) VTSSLST(LOCAL_B)
VTSSEL FUNCTION(DRAIN) STORCLAS(MVSB) VTSSLST(LOCAL_B)
```

...and you have appropriate Management Classes that point to the corresponding Storage Classes.

**Result:** Drain and reclaim are restricted to the VTSSs local to each site.

---

**Tip –** Before we get to the Usage Notes part of the show, there is one other example of VTSS preferencing that deserves mention in "Variation on a Theme: Uni-Directional or Bi-Directional?..." .

---

## VTSS Preferencing Usage Notes

- You use the MGMTDEF command to load the following statements, which must **all reside in the same data set for cross-validation**:
  - MGMTclas
  - STORclas
  - VTSSLST
  - VTSSSEL
  - STORLST
  - STORSEL
- The VTSSLST control statement is **only valid if** FEATures VSM(ADVMGMT) is specified.

# Archiving Data

You can use the ARCHAge and ARCHPol parameters of the MGMTclas statement to set an *Archive Policy* for VTVs in a Management Class. When the VTV's age exceeds the ARCHAge value, the VTV is eligible for archive per the Storage Class(es) specified on the ARCHPol parameter. The actual archive occurs in one of two ways:

- Automatically the next time the VTV is recalled and remigrated.
- On demand via the ARCHIve utility.

So a "what if" for this might be to ensure compliance. It turns out that I have data that I have to retain for 7 years for external auditors, but my internal auditors might like to look at it once a year as well. Here's what that solution looks like:

```
TAPEREQ DSN(COMPLY.**) MEDIA(VIRTUAL) MGMT(COMPLY)

MGMT NAME(COMPLY) IMMMED(DELETE) MIGPOL(LOC1) -
            ARCHAGE(365) ARCHPOL(REMDEEP)
STOR NAME(LOC1) ACS(00) MEDIA(STK1R)
STOR NAME(REMDEEP) ACS(01) MEDIA(STK2P
```

**FIGURE 2-8** Storage Classes / Management Classes for Archiving Data

For this scenario, I did the following:

- All compliance data is immediately migrated to the local ACS and grouped on 9840 media. After the migration succeeds, the VTVs are deleted from the VTSS. The "archive age" for this data is 365 days, in case the internal auditors want to see it in the next year. After that...
- ...the data is eligible to be archived (moved to) 9940 media in the remote ACS.

**Result**: Compliance, at the best possible cost, while optimizing virtual resources.

## Archive Usage Notes

As noted above, you have two methods to use for the actual archive: wait until the VTV is recalled and migrated, or do it on demand via the ARCHIve utility. The problem with waiting for remigration is that we're talking about data that isn't likely to be accessed. The chances are, the best way to archive stuff is to run the ARCHIve utility periodically or as-needed.

So here's some hints 'n' tips for using the ARCHive utility:

- To select the VTVs to archive, you can specify one of the following parameters:
  - MGMTclas to archive the VTVs in Storage Class(es) specified by the ARCHAge / ARCHPol parameter of the specified Management Class(es).
  - VTV to archive a list or range of VTVs per the Management Class(es) for those VTVs.

---

**Note –** If you do not specify a value for MGMTclas or VTV, VTCS scans all VTVs. I'd probably do business via Management Class, but I can see situations where you'd want to do it by VTV volser or all VTVs.

---

- By not specifying the MOVEVTV parameter, you can get a report (only) that is a valuable "what if" picture of how many VTVs, MVCs, and total MBs you will process with an archive request. StorageTek **strongly recommends**, therefore, that you first run ARCHIve **without** MOVEVTV, then adjust the job as needed **before** specifying MOVEVTV!. For more information, see *VTCS Command and Utility Reference.*
- Because demand archive can be resource intensive, you typically run ARCHIve during non-peak processing periods. You can also use the ARCHIve utility to override the CONFIG RECLAIM THRESHLD, MAXMVC, and CONMVC settings to optimize archive performance. You can also specify the maximum time for the archive in minutes on the ELAPSE parameter. Note that there are several limiting factors that influence archives (for example, MAXMVC and ELAPSE). VTCS enforces the strictest limiting factor. For example, if you run ARCHIve and specify ELAPSE equal to 5 hours and MAXMVC equal to 10 *and* VTCS archives 10 MVCs in one hour, then VTCS terminates the archive before the ELAPSE value expires.
- VTCS and HSC must be active to process a ARCHIve request except when you specify the POLICYdd parameter. POLICYdd (which forces "report only" mode) also provides an enhanced "what if" capability. You can create one or more alternate MGMTclas statements with different Archive Policies (different ARCHAge and ARCHPol values), and use POLICYdd to view the Archive Policy and resource use for each scenario.
- The RECONcil utility looks for all the world like ARCHive because RECONcil also moves VTVs from one Storage Class to another (that is, moves them from one MVC media to another and/or moves them from one ACS to another). Think of ARCHive as proactive and RECONcil as reactive, and you'll see the difference, as described in"Reconciling VTV Media and Location" ...

So let's say the 365 days is up, and the internal auditors haven't shown up. Time to go to Deep Archiveland. FIGURE 2-9 shows example JCL to run ARCHive as follows:

- Archive VTVs in Management Classes COMPLY to 9940 media in the remote ACS.
- Set MAXMVC to 60, CONMVC to 8, and ELAPSE to 60 for the ARCHive job.

```
//ARCHIVE     EXEC PGM=SWSADMIN
//STEPLIB   DD DSN=hlq.SLSLINK,DISP=SHR //SLSPRINTDD SYSOUT=*
//SLSIN     DD *
            ARCH MGMT(COMPLY) MAXMVC(60) CONMVC(8) ELAPSE(360)
MOVEVTV
```

**FIGURE 2-9** Example JCL for the ARCHive utility

---

**Tip –** The MOVEVTV parameter also gives you a report, so you can see how well (or not) you did. If your tuning parameters didn't archive the whole bunch, then tune up your job and rerun...

---

# Reconciling VTV Media and Location

Using RECONcil to reconcile VTV media and location basically means moving VTVs from one Storage Class to another. Is this at all like archiving data with ARCHive? In terms of data movement, yes. In terms of the reasons why you're doing it, it's a reactive rather than proactive move. You typically reconcile VTVs when:

- The VTVs are on the wrong media, in the wrong ACS, or both.
- An ACS is unavailable for a considerable period of time, then is brought back online. In this case, you 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.

The actual reconciling process gets a little tricky, so let's drop immediately into "RECONcil Example" .

## RECONcil Example

Let's say you want to reconcile VTVs that are on the wrong media and in the wrong ACS. How would you know this, for starters? Well, you peruse your VTV reports on a weekly basis as described in *Managing VTCS*. This week, I notice that all the VTVs in my production (PROD) Management Class are on the wrong media *and* in the wrong ACS! And the Storage Class doesn't look right, either...

How could this have happened? I *thought* I did the following:

```
STORCLAS NAME(LOC) ACS(00) MEDIA(STK1R)
STORCLAS NAME(CUSTA) ACS(00) MEDIA(STK1R)
STORCLAS NAME(CUSTB1) ACS(00) MEDIA(STK1R)
STORCLAS NAME(CUSTB2) ACS(01) MEDIA(STK2P
MGMT NAME(CUSTA) MIGPOL(CUSTA)
MGMT NAME(CUSTB) MIGPOL(CUSTB1,CUSTB2)
MGMT NAME(PROD) MIGPOL(LOC)
```

**FIGURE 2-10** Storage Classes/Management Classes for Workload Segregation

According to FIGURE 2-10, everything in Management Class PROD should have ended up on 9840 media in the local ACS...but in fact, they're all on 9940 media in the remote ACS...almost as if they were in the wrong Storage Class...

On closer examination, my production Management Class actually looks like this:

```
MGMT NAME(PROD) MIGPOL(CUSTA)
```

...which is not good for another reason, because it means my production data is co-resident on the same MVCs that are supposed to be dedicated to one of my customers. Time to run RECONcil, right? Well, not quite. RECONcil only moves things out of the wrong Storage Class, and right now, according to the way I wrote my Management Class statement, CUSTA is the right Storage Class! So before I run RECONcil, I have to go back and fix the Management Class thusly:

```
MGMT NAME(PROD) MIGPOL(LOC)
```

*Now* I can run RECONcil as shown in FIGURE 2-9 as follows:

- Move VTVs in Management Class PROD to their (updated) correct location in Storage Class LOC.
- Set MAXMVC to 60, CONMVC to 8, and ELAPSE to 60 for the RECONcil job.

```
//RECONCIL   EXEC PGM=SWSADMIN

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

//SLSPRINT  DD SYSOUT=*

//SLSIN     DD *

  RECON MGMT(PROD) MAXMVC(60)
            CONMVC(8) ELAPSE(360) MOVEVTV
```

**FIGURE 2-11** Example JCL for the RECONcil utility

## RECONcil Usage Notes

To select the VTVs to reconcile, you can specify one of the following parameters:

- MGMTclas to move the VTVs to the Storage Class(es) specified by the MIGpol parameter. This is what we did in "RECONcil Example" . Management Class points to the wrong Storage Class, get it pointed toward the right Storage Class, run RECONcil against the updated Management Class.
- STORclas to move the VTVs to the specified Storage Class(es). I'd probably use this when an ACS is unavailable for a considerable period of time.
- MVC to reconcile the VTVs on a list or range of MVCs. The VTVs are moved to the Storage Class(es) specified by he MIGpol parameter of the MGMTclas statement(s) for the VTVs. I'd use this or the VTV option as onesies/twosies kind of fix-ups.
- VTV to reconcile a list or range of VTVs. The VTVs are moved to the Storage Class(es) specified by the MIGpol parameter of the Management Class(es) for the VTVs.

---

**Note –**

---

- If you do not specify a value for MGMTclas or VTV, VTCS scans all VTVs.
- Because reconciling VTVs can be resource intensive, you typically run RECONcil during non-peak processing periods. You can also use the RECONcil utility to override the CONFIG RECLAIM THRESHLD, MAXMVC, and CONMVC settings to optimize reconcile performance. You can also specify the maximum time for the reconcile in minutes on the ELAPSE parameter.

  Note that there are several limiting factors that influence reconciliations (for example, MAXMVC and ELAPSE). VTCS enforces the strictest limiting factor. For example, if you run RECONcil and specify ELAPSE equal to 5 hours and MAXMVC equal to 10 *and* VTCS reconciles 10 MVCs in one hour, then VTCS terminates the reconciliations before the ELAPSE value expires.
- By not specifying the MOVEVTV parameter, you can get a report (only) that is a valuable "what if" picture of how many VTVs, MVCs, and total MBs you will process with a reconcile request. StorageTek **strongly recommends**, therefore, that

you first run RECONcil **without** MOVEVTV, then adjust the job as needed **before** specifying MOVEVTV!. For more information, see *VTCS Command and Utility Reference*.

- VTCS and HSC must be active to process a RECONcil request except when you specify the POLICYdd parameter. POLICYdd (which forces “report only” mode) also provides an enhanced “what if” capability. You can create one or more alternate MGMTclas statements with different reconciliation scenarios (different MIGpol values), and use POLICYdd to view the VTVs reconciled and resource use for each scenario.
- VTCS and HSC must be active to process a RECONcil request.

## Deleting Scratch VTVs

Deleting scratch VTVs is easy...you just specify DELSCR(YES) on the VTV's Management Class. The default is DELSCR(NO), so don't worry...you have to explicitly request deletion of scratch VTVs. Back in "Segregating Individual Workloads on Separate Sets of MVCs" , we had a Management Class for production data:

```
MGMT NAME(PROD) MIGPOL(LOC)
```

Let's say that because a lot of the data sets that get fed into this job stream are regularly scratched via the TMS, we want to use this to free VTSS buffer space. No problem, just update the Management Class as shown:

```
MGMT NAME(PROD) MIGPOL(LOC) DELSCR(YES)
```

Note that this change to the Management Class has two interesting effects:

1. At scratch synchronization time, the VTV is deleted from the VTSS buffer.
2. Also at scratch synchronization time, the migrated VTV is marked non-current, which makes the MVC (more) eligible for reclamation.

Okay, it's time for the obligatory warnings about DELSCR(YES), so please **study this carefully**.

---

**Caution –**

---

- When you scratch a VTV with DELSCR YES attribute, **VSM erases the VTV data at scratch synchronization time**, which eliminates the ability to "unscratch" a VTV to recover data!
- **Also note** that when using previous releases of HSC SLUCONDB to perform scratch synchronization, SLUCONDB attempted to scratch everything that was marked scratch in the TMS database. For HSC 6.0 and above, however, SLUCONDB has been updated to scratch *only* those volumes that are not in scratch status in the HSC CDS. Therefore, for HSC 6.0 and above, the *only* possibilities of inadvertently scratching a VTV resulting in data loss at scratch synchronization time are as follows:
  - If you are running the HSC SLUADMIN Scratch Update Utility at the same time that SLUCONDB is running.
  - If you do not specify the current TMS database and/or the current HSC CDS when using SLUCONDB.

  For more information about HSC scratch synchronization with the Scratch Conversion Utility (SLUCONDB), see *HSC System Programmer's Guide for MVS*.

- **Also note** that for HSC and MVS/CSC, the DELDISP parameter has two values that affect how HSC manages the scratch status of VTVs and real volumes that were mounted scratch and the delete disposition on the dismount message is delete ('D').
- For more information about ExLM scratch synchronization with the SYNCVTV function, see the ExLM documentation.

## Immediately Migrating VTVs On Dismount

You have three possibilities with this one, which is implemented via the IMMEDmig parameter of the MGMTclas statement:

IMMEDmig(NO)

Specifies that VTCS does not immediately migrate the VTV, but migrates it according to standard VTCS migration criteria. This is the default, so if you don't want to specify immediate migration, don't do anything.

IMMEDmig(KEEP)

Specifies that VTCS immediately migrates a VTV and keeps a copy resident on the VTSS until the VTV become eligible for deletion. This is a nice one, because you can get the migration done quickly, but there's still a copy in VTSS space. Result: you safeguard your data **and** optimize accessibility.

IMMEDmig(DELETE)

Specifies that VTCS immediately migrates the VTV and then deletes it from the VTSS. This is the Bigger Hammer that says "Get it on tape ASAP...once that's happened, I'm probably not going to need it again for a while, if ever, so delete the copy in VTSS space."

So a "what if" for this might be that I have an incoming job stream that is basically compliance data that I have to retain for 7 years for the IRS, but I probably won't use it much in my day to day operations unless somebody asks...like the IRS. So here's what that TAPEREQ/MGMTclas/STORclas combo looks like:

```
TAPEREQ DSN(IRS.**) MEDIA(VIRTUAL) MGMT(IRS)
MGMT NAME(    IRS) IMMMED(DELETE) MIGPOL(REMDEEP)
STOR NAME(REMDEEP) ACS(01) MEDIA(STK2P)
```

So what have I accomplished with this setup? Well, the following:

- All compliance data of this flavor is immediately migrated to the remote ACS and grouped on 9940 media.
- After the migration succeeds, the VTVs are deleted from the VTSS.

**Result**: Once again, compliance, at the best possible cost, while optimizing virtual resources.

## Immediate Migrate Usage Notes

- With the IMMEDmig parameter of the MGMTclas statement, you can specify whether VTCS will immediately schedule a VTV for migration after dismounting it. When the migration actually occurs depends on RTD availability, Storage Classes for immediate migration, and the total number of immediate migrates scheduled.
- Specify NO (the default) if you *do not* want immediately migration and you *do* want other migration policies to determine your migration strategy.

---

**Caution –** IMMEDmig DELETE ensures that VTVs are immediately migrated and frees VTSS space; however, it preferences migration processing, may increase I/O to the RTDs, uses up MVC space more quickly, and may also increase the need for MVC space reclamation and VTV recalls.

---

- The RESTIME and IMMEDmig(DELETE) parameters are mutually exclusive; for more information, see *VTCS Command and Utility Reference* and *Installing and Configuring VTCS*.
- IMMEDmig KEEP and IMMEDmig DELETE are mutually exclusive with CONFIG HOST NOMIGRAT. If you specify both, the IMMEDmig value overrides NOMIGRAT (for only those VTVs with the IMMEDmig value), and VTCS does not issue a message about this override.

  If VTCS stops with pending immediate migrations, these migrations will resume when VTCS restarts.

# Specifying VTV Residency Interval before Automatic Migration Candidacy

Let's say you have some data that needs to be both high-access **and** deep archived. It'll cost, but it can be done, to whit:

```
TAPEREQ DSN(LOVEDONES.**) MEDIA(VIRTUAL) MGMT(FAVES)
MGMT NAME(    FAVES) RESTIME(9999) MIGPOL(REMDEEP2)
STOR NAME(REMDEEP2) ACS(01) MEDIA(STK2P)
```

## RESTIME Usage Notes

- The RESTIME value in a VTV's Management Class Specifies how long VTCS attempts to keep a VTV as VTSS-resident before becoming a preferred automatic migration candidate. Valid values are 1 - 9999 hours. The interval starts at the time that instance of the VTV is created. A new instance of the VTV is created whenever the VTV is updated. At automigration time, the creation date and time of the VTV instance plus the RESTIME value is compared to the TOD clock to determine if the VTV is an automatic migration candidate.
- A VTV's Management Class (and attributes, such as RESTIME) is set after a scratch mount or optionally after a specific mount if VTVattr = ALLmount.
- The RESTIME value is *only* a recommendation. VTCS can migrate a VTV before its residency interval expires if the DBU has not reached the LAMT or the specified migrate-to-threshold value and no VTVs have expired their residency intervals.
- You can do a demand migrate of a VTV and delete it from the VTSS even if its residency interval has not expired.
- The RESTIME and IMMEDmig(DELETE) parameters are mutually exclusive; for more information, see "Immediately Migrating VTVs On Dismount" on page 18.

## RESTIME Example

1. You create Management Class with a RESTIME of 10 hours.
2. A job requests a scratch mount for the Management Class you created in Step 1 VTCS selects and mounts a scratch VTV. The VTV is updated, so at dismount time, its RESTIME value is set to 10 hours (which began when VTCS mounted the VTV).
3. VTCS migrates the VTV after 3 hours, then recalls the VTV 2 hours later for a read. The RESTIME value is *not* reset, and there are now 5 hours of residency remaining.
4. 2 hours later, a job updates the VTV, which was 7 hours old. The update creates a new instance of the VTV and the residency interval will restart from the time the VTV was mounted for update.
5. 24 hours later, VTCS migrates the VTV, then recalls it 2 days later for a read. VTCS does *not* create a new instance of the VTV because it is not updated. The residency interval has expired and the VTV is therefore an automatic migration candidate based only on least-recently-used / size criteria.

   A week later, the VTV is scratched. VTCS eventually selects and mounts the VTV to satisfy a scratch mount request. If the VTV is updated, its residency interval is set to the RESTIME value of the Management Class being used.

# ...And Now a Commercial Message: Named MVC Pools or Not?

Named MVC Pools are the right tool for the job in a couple of specific instances:

- They are one of the foundations of the Offsite Vault Feature, an optional DR feature of VTCS. If you want to know how they work there, see *VTCS Offsite Vault Feature User's Guide*.
- For all of you service groups, you can use Named MVC Pools to give an application ownership of the MVCs in the named pool. For example, a service group might elect to use Named MVC Pools where its customers have legal requirements to buy and own a group of MVCs.

However, if you do *not* have a specific requirement for Named MVC Pools but you *do* want to group or segregate client data on MVCs, StorageTek **strongly advises** that you do *not* use Named MVC Pools. Instead, use the methods described in the following sections:

- "Grouping Multiple Workloads on Shared MVCs"
- "Segregating Individual Workloads on Separate Sets of MVCs"

The above sections tell how to use Storage Classes to group or segregate data on MVCs that are selected from the system-wide MVC pool. In this case, you need to manage a only a single MVC pool.

If you create Named MVC Pools, you must **explicitly** manage each pool, which includes ensuring that each pool has sufficient free MVCs and available MVC space, and potentially includes setting different policies for each pool using the MVCPool MVCFREE, MAXMVC, THRESH, and START parameters.

If you **do** elect to use Named MVC Pools, go to "Creating and Using Named MVC Pools" .

## ▼ Creating and Using Named MVC Pools

To create and use Named MVC Pools, do the following:

1. **Modify existing MVCPool statements and/or add additional statements to define the Named MVC Pools.**

   If you do not specify the MVCPOOL NAME parameter, VTCS does not create a Named MVC Subpool and assigns the specified volumes to the default pool (DEFAULTPOOL). You cannot create Named MVC Pools with the reserved names DEFAULTPOOL and ALL.

   You can use the optional MVCFREE, MAXMVC, THRESH, and START parameters to specify values for the Named MVC Pool that override the global values specified on CONFIG.

   For example, the following MVCPool statement defines volsers 800000 - 804999 as MVCs in Named Pool CUST1POOL with reclamation parameter values that override the CONFIG global values.

```
MVCP V(800000 - 804999)NAME (CUST1POOL) MAXMVC=20 THRESH=70 START=35
```

---

**Caution –** A Named MVC Pool *must* contain media of the type specified in the corresponding STORCLAS MEDIA parameter (see Step 4); otherwise, VTCS will issue a "no MVCs available" message during MVC selection.

---

2. **Run the VT MVCDEF command to activate the updated data set.**

3. **Enable the VSM Advanced Management Feature via the HSC FEATURES control statement.**

   The Advanced Management Feature is required for the Storage Classes you define in Step 4.

4. **Define Storage Classes and associate them with Named MVC Pools.**

   For example, the following STORclas statement defines STORCL1 and associates this Storage Class with Named MVC Pool CUST1POOL. Requests to use MVCs for storage class STORC1 will result in MVCs being selected only from the named pool CUST1POOL.

   STOR NAME(STORCL1) MEDIA(ECART,ZCART,STK1R) MVCPOOL(CUST1POOL)

5. **Create Management Classes that specify the Storage Classes you defined in** Step 4 **and specify these Management Classes when you route data to the Named MVC Pool.**

   For more information, see "Creating and Using VTCS Management and Storage Classes: The Basics" .

6. **Specify the Management Class name to VTCS on any of the following:**
   - The SMC TAPEREQ statement.
   - SMS routines that you write to the StorageTek DFSMS interface; for more information, see *SMC Configuration and Administration Guide*.

---

**Note –** If you specify a Management Class on a TAPEREQ statement and an SMS routine, the Management Class on the SMS routine takes precedence.

---

If VTCS receives a request to migrate a VTV that is assigned to an invalid Management Class, VTCS will dynamically create the !ERROR Storage Class and migrate the VTVs defined by the invalid Management Class to the !ERROR Storage Class. MVC reports show when a VTV is migrated to this Storage Class.

CHAPTER 3

# Using Clustered VTSS Configurations

Ever wish you could copy VTVs from one VTSS to another? Well, you can, thanks to the magic of Clustered VTSSs. Clustered VTSS is a powerful tool for applications such as but not limited to DR (Disaster Recovery) solutions. As you've probably guessed, however, with Clustered configurations, Some Assembly *is* Required. So we'll start with a couple of real world examples to prevent Information Overload:


- "Example: Uni-Directional Clustered VTSS - Dual ACS Configuration, 8 VCF Cards, 4 CLINKs, FICON Directors for 8 RTDs" on page 30
- "Example: Bi-Directional Clustered VTSS - Dual ACS Configuration, 8 VCF Cards, 4 CLINKs, FICON Directors for 8 RTDs" on page 36
- ...and, as an added bonus, "Variation on a Theme: Uni-Directional or Bi-Directional?..." on page 41.


That's the introduction-by-example. The details, which can peruse at your leisure, are as follows:


- "Clustered VTSS Requirements" on page 43
- "How Clustered VTSS Configurations Work" on page 44
- "Managing Clustered VTSS Systems" on page 51

# Example: Uni-Directional Clustered VTSS - Dual ACS Configuration, 8 VCF Cards, 4 CLINKs, FICON Directors for 8 RTDs

FIGURE 3-1 shows an example of a Uni-Directional Clustered VTSS Dual ACS system. In this example, I only have one MVS host, but it's putting out a lot of critical data that I want protected to the max using two brand new VSM4s that I just purchased. No problem: VTSS1 is the Primary VTSS, and it's connected to the Secondary (VTSS2) via Cluster Links (CLINKs). If the Management Class for a VTV specifies replication, presto, when the VTV arrives in VTSS1, it is *replicated* (copied) to VTSS2, and also immediately migrated (with KEEP). Result: I have increased data availability (there's a copy of the VTV in each VTSS in case one fails) *and* data protection (the VTV is also on square tape in case both VTSSs go offline). Clustered VTSS is a great solution, therefore, for business continuity and business resumption.

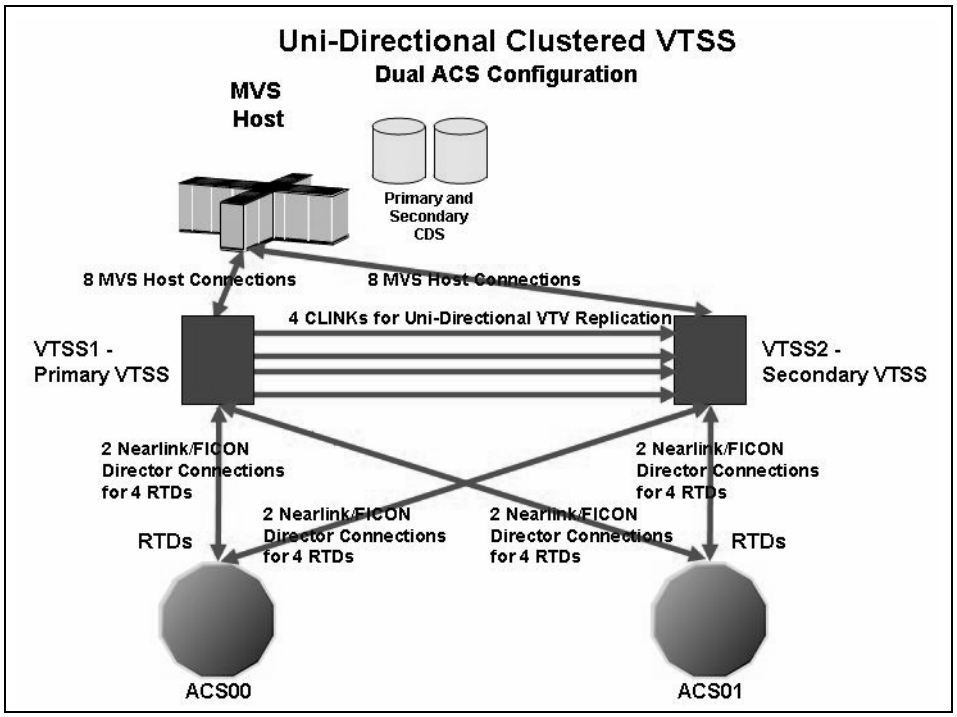



**FIGURE 3-1** Dual ACS Uni-Directional Clustered VTSS Configuration

Now it's time to take a look at the hardware for this Clustered configuration. FIGURE 3-2 shows CONFIG channel interface identifiers for a VSM4 with 8 VCF cards. In this configuration, we've allocated:

- 8 Host ports.
- 4 ports for RTDs. The RTD ports are all connected to FICON directors, each of which is attached to RTDs, so the CHANIF identifiers for both RTDs are shown on each port. This allows Back-End connection to 8 RTDs, although only one RTD per port/Director can be active at a time.

- 4 ports for CLINK connections to form a Uni-Directional VTSS Cluster, and 8 ports to host connections. To form the clustered VTSS, we'll have two VSM4s (VTSS1 and VTSS2) configured identically as shown in FIGURE 3-2.

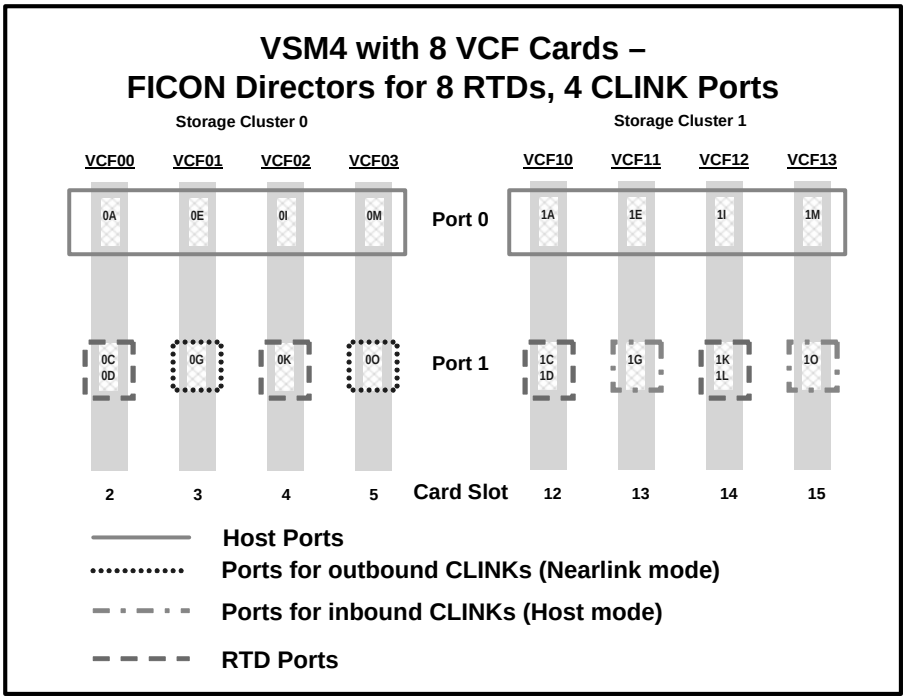



**FIGURE 3-2** VSM4 with 8 VCF Cards, 8 Host Ports, FICON Directors for 8 RTDs, 4 CLINK Ports

Okay, we've seen what our example Uni-Directional Cluster looks like, and we've seen the VCF card port configurations required. Now let's tie it all together in "Configuring and Managing a Uni-Directional Clustered VTSS System" on page 32.

## ▼ Configuring and Managing a Uni-Directional Clustered VTSS System

To configure and manage the Uni-Directional Clustered system shown in Figure 11. on page 26, do the following:

1. **Ensure that your system has the Clustered VTSS requirements.**
2. **Use CONFIG to create CLUSTER and CLINK statements to define the VTSS Cluster and its connections.**

   FIGURE 3-3 shows example CONFIG JCL to define a Uni-Directional Cluster of two VSM4s (VTSS1 and VTSS2) as shown in Figure 11 on page 26. **Note that**:

   - The CLUSTER statement defines the Cluster as consisting of VTSS1 and VTSS2.
   - There are CLINK statements using the sending (Nearlink Mode) ports of **only VTSS1** to enable the Cluster as Uni-Directional, where VTSS1 is the Primary and VTSS2 is the Secondary.

```
//CREATECFG                        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
//CFG22202                      DD DSN=FEDB.VSMLMULT.CFG22202,DISP=SHR
//SLSPRINT                      DD   SYSOUT=*
//SLSIN                         DD   *
 CONFIG RESET CDSLEVEL(V62ABOVE)
 GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
 LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE
 RECLAIM THRESHLD=70 MAXMVC=30 START=40 CONMVC=5
 VTVVOL LOW=905000 HIGH=999999 SCRATCH
 VTVVOL LOW=C00000 HIGH=C25000 SCRATCH
 VTVVOL LOW=RMM000 HIGH=RMM020 SCRATCH
 MVCVOL LOW=N25980 HIGH=N25989
 MVCVOL LOW=N35000 HIGH=N35999
 VTSS NAME=VSMPR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
 RTD  NAME=PR11A00 DEVNO=1A00 CHANIF=0C
 RTD  NAME=PR11A01 DEVNO=1A01 CHANIF=0D
 RTD  NAME=PR11A02 DEVNO=1A02 CHANIF=0K
 RTD  NAME=PR11A03 DEVNO=1A03 CHANIF=0L
 RTD  NAME=PR12A08 DEVNO=2A08 CHANIF=1C
 RTD  NAME=PR12A09 DEVNO=2A09 CHANIF=1D
 RTD  NAME=PR12A0A DEVNO=2A0A CHANIF=1K
 RTD  NAME=PR12A0B DEVNO=2A0B CHANIF=1L
 VTD LOW=9900 HIGH=99FF
 VTSS NAME=VSMPR2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
 RTD  NAME=PR23A00 DEVNO=3A00 CHANIF=0C
 RTD  NAME=PR23A01 DEVNO=3A01 CHANIF=0D
 RTD  NAME=PR23A02 DEVNO=3A02 CHANIF=0K
 RTD  NAME=PR23A03 DEVNO=3A03 CHANIF=0L
 RTD  NAME=PR24A08 DEVNO=4A08 CHANIF=1C
 RTD  NAME=PR24A09 DEVNO=4A09 CHANIF=1D
 RTD  NAME=PR24A0A DEVNO=4A0A CHANIF=1K
 RTD  NAME=PR24A0B DEVNO=4A0B CHANIF=1L
 VTD LOW=9900 HIGH=99FF
 CLUSTER NAME=CLUSTER1 VTSSs(VTSS1,VTSS2)
 CLINK VTSS=VTSS1 CHANIF=0G
 CLINK VTSS=VTSS1 CHANIF=0O
 CLINK VTSS=VTSS1 CHANIF=1G
 CLINK VTSS=VTSS1 CHANIF=1O
```

**FIGURE 3-3** CONFIG example: Dual ACS Uni-Directional Clustered VTSS System, VSM4 FICON Back-End

3. **Enable the Advanced Management Feature.**

   The Advanced Management Feature is required to enable the REPLICAT parameter of the MGMTclas statement.

4. **Specify the Conditional Replication setting on the CONFIG GLOBAL statement.**

```
CONFIG GLOBAL REPLICAT=CHANGED
```

**FIGURE 3-4** CONFIG GLOBAL Setting for VTV Replication

In FIGURE 3-5, CONFIG GLOBAL REPLICAT=CHANGED specifies:

- Replicate VTVs only if the VTV is updated and an identical copy does not exist in the Secondary.
- Via the MIGPOL parameter, migrate duplexed to ACSs 00 and 01 by Storage Classes you will create in Step 6 on page 31.

What if I wanted to unconditionally replicate VTVs? I would specify (you guessed it), CONFIG GLOBAL REPLICAT=ALWAYS.

5. **Create a Management Class that specifies VTV replication and two Storage Classes to migrate (duplexed) the replicated VTVs.**

```
MGMT NAME(VSMREPL) REPLICAT(YES) MIGPOL(REPLSTR1,REPLSTR2)
```

**FIGURE 3-5** Management Class for VTV Replication

---

**Note –**

---

- Note the subtle interaction between GLOBAL REPLICAT, which specifies *when* the replication can occur, and MGMTclas REPLICAT(YES), which says, “when the GLOBAL REPLICAT condition says it’s time, go ahead and replicate.”
- The Management Class VSMREPL **does not** specify an immediate migrate policy. VTV replication automatically enforces immediate migrate. The VTVs in this Management Class will be added to the immediate migration queue on VTSS once the replication has completed. Note that duplexing **is not** a requirement for replicate VTVs. For more information, see “How Clustered VTSS Configurations Work” on page 40.

6. **Create the Storage Classes for the MVCs that contain the replicated, migrated VTVs.**

```
STOR NAME(REPLSTR1) ACS(00) MEDIA(STK1R) MIRATE(RECEIVER)
STOR NAME(REPLSTR2) ACS(01) MEDIA(STK1R) MIGRATE(RECEIVER)
```

**FIGURE 3-6** Storage Classes for Replicated, Migrated VTVs

In FIGURE 3-6, the STORclas statement defines Storage Classes REPLSTR1 and REPLSTR2 referenced in the MIGPOL parameter in Step 5 on page 30. **Also note** that the MIGRATE parameters on the Storage Classes specify that the VTSS receiving the replicated VTV...in this case VTSS2, the Secondary, does the migration to both ACSs. This is a handy way of ensuring that the Secondary functions as the “migrate engine.”

7. **Load the MGMTclas and STORclas control statements with the MGMTDEF command.**

```
MGMTDEF DSN(hsc.parms)
```

**FIGURE 3-7** MGMTDEF Command to Load Statements

8. **Create a TAPEREQ statement to route the critical data to VSM and assign Management Class VSMREPL to the data.**

```
TAPEREQ DSN(*.PAYROLL.**) MEDIA(VIRTUAL) MGMT(VSMREPL)
```

**FIGURE 3-8** TAPEREQ Statement to Route Critical Data, Assign Management Class VSMREPL

In FIGURE 3-8, the TAPEREQ statement specifies:

- Route data sets with HLQ mask *.PAYROLL.** to VSM...
- .....and assign Management Class VSMREPL that you created in Step 5 on page 30.

---

**Caution –** To replicate VTVs, **both VTSS1** and VTSS2 must be varied online to VTCS so that VTCS can send control commands to both VTSSs. See "How Clustered VTSS Configurations Work" on page 44 for more information.

---

---

**Note – Also note** the following:

---

- You can also use esoteric substitution via SMC TAPEREQ statement or SMC DFSMS ACS routines to route replication jobs to VSM. For more information, see *SMC Configuration and Administration Guide*.

9. **Check your HSC PARMLIB options to ensure that subtype 28 records are enabled.**

   If enabled, VTSS clustering writes a subtype 28 record for each replication performed.

   ...and you're home free!

# Example: Bi-Directional Clustered VTSS - Dual ACS Configuration, 8 VCF Cards, 4 CLINKs, FICON Directors for 8 RTDs

FIGURE 3-9 shows an example of a Bi-Directional Clustered VTSS Dual ACS system. This system is very similar to the uni-directional example, but goes one step further: There are two MVS hosts sharing a CDS, and everything in the picture is cross-connected. I basically have sites mirroring each other for the ultimate in data availability and protection. To make this happen...that is, to make it bi-directional...I have to configure the two VTSSs as peers via the CLINK statements.

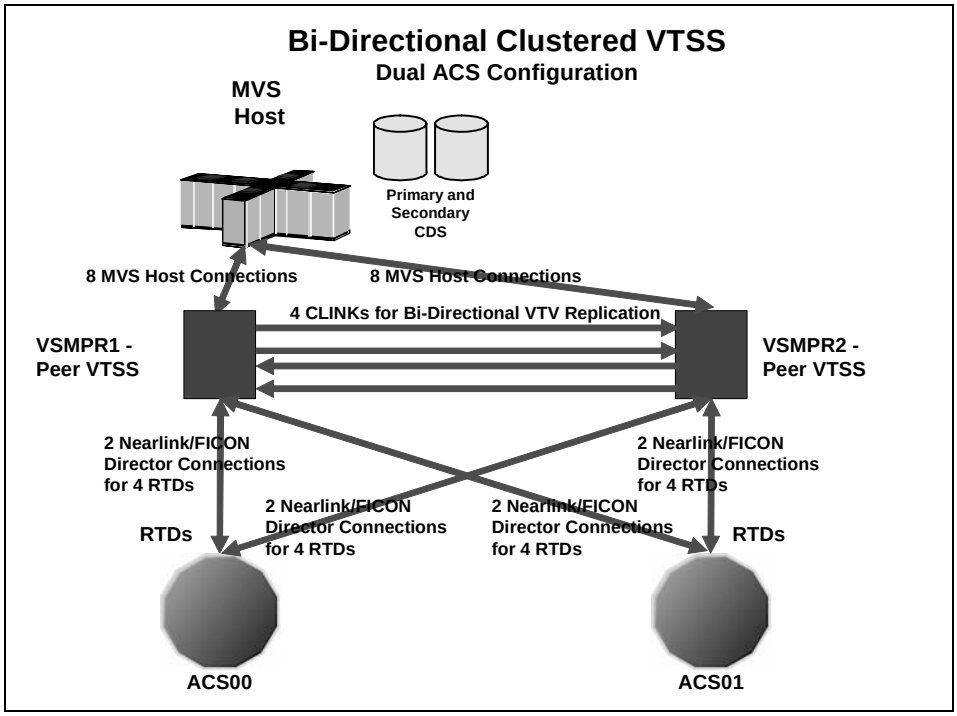



**FIGURE 3-9** Dual ACS Bi-Directional Clustered VTSS Configuration

FIGURE 3-10 should look pretty familiar, because the port configuration is identical to what we described for the uni-directional system in FIGURE 3-1. Again, the difference between uni-directional and bi-directional Clusters is in the software, which we'll get to next...

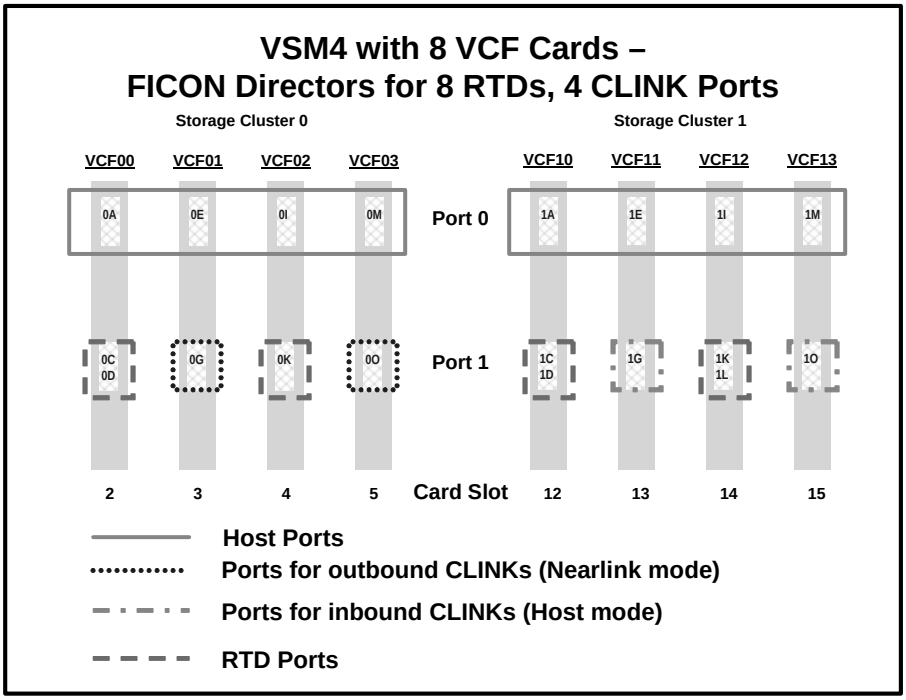



**FIGURE 3-10** VSM4 with 8 VCF Cards, 8 Host Ports, FICON Directors for 8 RTDs, 4 CLINK Ports

## ▼ Configuring and Managing a Bi-Directional Clustered System

**To configure and manage the Bi-Directional Clustered system shown in** FIGURE 3-9**, do the following**:

1. **Ensure that your system has the Clustered VTSS requirements described in *Installing and Configuring VTCS*.**
2. **Use CONFIG to create CLUSTER and CLINK statements to define the VTSS Cluster and its connections.**

FIGURE 3-11 shows example CONFIG JCL to define a Bi-Directional Cluster of two VSM4s (VSMPR1 and VSMPR2) as shown in Figure 11 on page 26. **Note that**:

- The CLUSTER statement defines the Cluster as consisting of VSMPR1 and VSMPR2.

- There are CLINK statements using the sending (Nearlink Mode) ports of **both VTSSs** to enable the Cluster as Bi-Directional. As described on page 27, the Nearlink ports are 0G and 1G on both VTSSs. So the trick to Bi-Directional is to ensure that you have data pipes going in both directions, and you do it by writing your CLINK statements in this fashion.

```
//CREATECFG                          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
//CFG22202                       DD DSN=FEDB.VSMLMULT.CFG22202,DISP=SHR
//SLSPRINT                       DD   SYSOUT=*
//SLSIN                          DD   *
 CONFIG RESET CDSLEVEL(V62ABOVE)
 GLOBAL MAXVTV=65000 MVCFREE=60 VTVATTR=SCRATCH RECALWER=YES
 LOCKSTR=STK_VTCS_LOCKS VTVPAGE=LARGE
 RECLAIM THRESHLD=70 MAXMVC=30 START=40 CONMVC=5
 VTVVOL LOW=905000 HIGH=999999 SCRATCH
 VTVVOL LOW=C00000 HIGH=C25000 SCRATCH
 VTVVOL LOW=RMM000 HIGH=RMM020 SCRATCH
 MVCVOL LOW=N25980 HIGH=N25989
 MVCVOL LOW=N35000 HIGH=N35999
 VTSS NAME=VSMPR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
 RTD  NAME=PR11A00 DEVNO=1A00 CHANIF=0C
 RTD  NAME=PR11A01 DEVNO=1A01 CHANIF=0D
 RTD  NAME=PR11A02 DEVNO=1A02 CHANIF=0K
 RTD  NAME=PR11A03 DEVNO=1A03 CHANIF=0L
 RTD  NAME=PR12A08 DEVNO=2A08 CHANIF=1C
 RTD  NAME=PR12A09 DEVNO=2A09 CHANIF=1D
 RTD  NAME=PR12A0A DEVNO=2A0A CHANIF=1K
 RTD  NAME=PR12A0B DEVNO=2A0B CHANIF=1L
 VTD LOW=9900 HIGH=99FF
 VTSS NAME=VSMPR2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
 RTD  NAME=PR23A00 DEVNO=3A00 CHANIF=0C
 RTD  NAME=PR23A01 DEVNO=3A01 CHANIF=0D
 RTD  NAME=PR23A02 DEVNO=3A02 CHANIF=0K
 RTD  NAME=PR23A03 DEVNO=3A03 CHANIF=0L
 RTD  NAME=PR24A08 DEVNO=4A08 CHANIF=1C
 RTD  NAME=PR24A09 DEVNO=4A09 CHANIF=1D
 RTD  NAME=PR24A0A DEVNO=4A0A CHANIF=1K
 RTD  NAME=PR24A0B DEVNO=4A0B CHANIF=1L
 VTD LOW=9900 HIGH=99FF
 CLUSTER NAME=CLUSTER1 VTSSs(VSMPR1,VSMPR2)
 CLINK VTSS=VSMPR1 CHANIF=0G
 CLINK VTSS=VSMPR1 CHANIF=1G
 CLINK VTSS=VSMPR2 CHANIF=0G
 CLINK VTSS=VSMPR2 CHANIF=1G
```

**FIGURE 3-11** CONFIG example: Dual ACS Bi-Directional Clustered VTSS System, VSM4 FICON Back-End

3. **Enable the Advanced Management Feature.**

   The Advanced Management Feature is required to enable the REPLICAT parameter of the MGMTclas statement.

4. **Specify the Conditional Replication setting on the CONFIG GLOBAL statement.**

```
CONFIG GLOBAL REPLICAT=CHANGED
```

**FIGURE 3-12** CONFIG GLOBAL Setting for VTV Replication

As with the uni-directional example, in FIGURE 3-12, we use CONFIG GLOBAL REPLICAT=CHANGED.

5. **Create a Management Class that specifies VTV replication and two Storage Class to migrate (duplexed) the replicated VTVs.**

```
MGMT NAME(VSMREPL) REPLICAT(YES) MIGPOL(REPLSTR1,REPLSTR2)
```

**FIGURE 3-13** Management Class for VTV Replication

FIGURE 3-13 should look familiar...replicate VTVs only if changed and not in the other VTSS in the Cluster, migrate duplexed to ACSs 01 and 00 by Storage Classes you will create in Step 6.

6. **Create the Storage Classes for the MVCs that contain the replicated, migrated VTVs.**

```
STOR NAME(REPLSTR1) ACS(01) MEDIA(STK1R) MIRATE(EITHER)
STOR NAME(REPLSTR2) ACS(00) MEDIA(STK1R) MIGRATE(EITHER)
```

**FIGURE 3-14** Storage Classes for Replicated, Migrated VTVs

In FIGURE 3-14, the STORclas statement defines Storage Classes REPLSTR1 and REPLSTR2 referenced in the MIGPOL parameter in Step 5. **Also note** that, to optimize VTSS and RTD resources, the MIGRATE parameters on the Storage Classes allow migrates to come from either VTSS. This is a typical strategy for Bi-Directional, or Peer to Peer VTSS Clusters.

7. **Load the MGMTclas and STORclas control statements with the MGMTDEF command.**

```
MGMTDEF DSN(hsc.parms)
```

**FIGURE 3-15** MGMTDEF Command to Load Statements

8. **Create a TAPEREQ statement to route the critical data to VSM and assign Management Class VSMREPL to the data.**

```
TAPEREQ DSN(*.PAYROLL.**) MEDIA(VIRTUAL) MGMT(VSMREPL)
```

**FIGURE 3-16** TAPEREQ Statement to Route Critical Data, Assign Management Class VSMREPL

In FIGURE 3-16, the TAPEREQ statement specifies:

- Route data sets with HLQ mask *.PAYROLL.** to VSM...
- .....and assign Management Class VSMREPL that you enabled in Step 5.

---

**Caution –** To replicate VTVs, **both VSMPR1** and VSMPR2 must be varied online to VTCS so that VTCS can send control commands to both VTSSs. See "How Clustered VTSS Configurations Work" on page 44 for more information.

---

**Note – Also note** the following:

---

- You can also use esoteric substitution via SMC TAPEREQ statement or NCS User Exits to route replication jobs to VSM. If an esoteric is substituted that spans all VTDs in **all** Peer VTSSs, then VTCS can continue to correctly influence allocation if a one of the Peer VTSSs in a Cluster is taken offline.
- For SMC, a Management Class name, if it is assigned in the StorageTek DFSMS Interface, is available at allocation time. Therefore the esoteric assigned in the interface no longer needs to contain only VTSSs that are part of clusters. As long as the esoteric contains some drives located on the Primary of a full function cluster, SMC has sufficient information to direct allocation to a drive on a Primary VTSS if the Management Class specifies replication enabled.

9. **Check your HSC PARMLIB options to ensure that subtype 28 records are enabled.**

   If enabled, VTSS clustering writes a subtype 28 record for each replication performed.

   ...and chalk up another success story using Clustered VTSS.

# Variation on a Theme: Uni-Directional or Bi-Directional?...

...the choice is yours! This is a Way Cool variation on the theme we showed with Bi-Directional Clustering. The pictures are the same...see FIGURE 3-9 and FIGURE 3-10. The CONFIG deck is also the same one shown in FIGURE 3-11. We're going to use VTSSLST and VTSSSEL statements, however, to make a Bi-Directional Cluster Uni-Directional. Why would I want to do this? What if I wanted to switch the roles of the Primary and Secondary VTSSs? Easy, you just start with the same setup as described in the procedure beginning on "Configuring and Managing a Bi-Directional Clustered System" on page 37. After you complete Step 6, you throw in a subtle change with the following VTSSLST and VTSSSEL statements.

```
VTSSLST NAME(SITEA) VTSS(VSMPR1)
VTSSSEL FUNCTION(SCRATCH) HOST(MVSA) VTSSLST(SITEA)
VTSSSEL FUNCTION(SPECIFIC) HOST(MVSA) VTSSLST(SITEA)
```

**FIGURE 3-17** VTSSLST/VTSSSEL Statements - VSMPR1 Primary, VSMPR2 Secondary

In FIGURE 3-17:

- The VTSSLST statement defines VTSS list SITEA that contains **only** VSMPR1.
- The VTSSSEL statements direct scratch and specific VTV mounts from MVSA to SITEA, which contains **only** VSMPR1...thus effectively making it the Primary.

So this Cluster is actually Bi-Directional, but VTSSLST and VTSSSEL statements give us the flexibility to effectively make either VTSS the Primary and the other the Secondary by simply loading the corresponding MGMTclas, STORclas, VTSSLST, and VTSSSEL control statements with the MGMTDEF command.

What if we wanted to switch the Primary and Secondary? No problem, just rewrite the VTSSLST and VTSSSEL statements to make VSMPR2 the Primary and VSMPR1 the Secondary.

```
VTSSLST NAME(SITEB) VTSS(VSMPR2)
VTSSSEL FUNCTION(SCRATCH) HOST(MVSB) VTSSLST(SITEB)
VTSSSEL FUNCTION(SPECIFIC) HOST(MVSB) VTSSLST(SITEB)
```

**FIGURE 3-18** VTSSLST/VTSSSEL Statements - VSMPR2 Primary, VSMPR1 Secondary

In FIGURE 3-18:

- The VTSSLST statement defines list SITEB that contains **only** VSMPR2.
- The VTSSSEL statements direct scratch and specific VTV mounts from MVSB to SITEB, which contains **only** VSMPR2...thus effectively making it the Primary.

Finally, what if the time came that things worked better with this Cluster as a true Bi-Directional Cluster? Easy...just delete the VTSSLST and VTSSSEL statements, reload your defs, and you're all set...

Now *that's* flexibility!

# Another Variation: Synchronous or Asynchronous Replication?

You have a choice: you can either replicate synchronously or asynchronously, depending on your site's policies. **Please note** the following, however:

---

**Caution –** With synchronous replication the time required to replicate a virtual volume will delay the completion of any job creating data that has a synchronous replication policy.

---

Okay, you've read the fine, print, and you've decided to implement Synchronous Replication. Here's how to do it...

## ▼ Implementing Synchronous Replication

---

**Caution –** With synchronous replication the time required to replicate a virtual volume will delay the completion of any job creating data that has a synchronous replication policy.

---

1. **Ensure that your system has the Synchronous Replication requirements described in** TABLE P-2 on page xiv**.**
2. **With all HSC/VTCS systems down, use CONFIG GLOBAL to enable Synchronous Replication:**

   ```
   CONFIG GLOBAL SYNCHREP=YES
   ```

3. **Ensure that the CONFIG GLOBAL REPLICAT parameter is set as desired:**

   ALWAYS

   The replicate request is added to the VTCS replication queue every time the VTV is dismounted, regardless of whether the VTV was changed while it was mounted (the default).

   CHANGED

   The replicate request is added to the VTCS replication queue if the VTV:

   - Was changed while it was mounted **or**
   - Was only read while mounted but less than the expected number of MVC copies of the VTV exist.

4. **Specify Synchronous Replication on the desired MGMTClas statements:**

   ```
   MGMT (name) ..... REP(YES_SYNC)
   ```

# Clustered VTSS Requirements

**TABLE 3-1** Clustered VTSS Requirements

| Component | Requirement |
|---|---|
| 2 VTSSs within a cluster (ESCON Interfaces) | The Primary and Secondary VTSSs can be any combination of VSM3 and VSM4 where the Secondary can be of any capacity. All hosts **must** be at VTCS 5.1.0 or above to enable this feature. For example, all of the following are valid:<br>■ Primary VSM4, Secondary VSM3<br>■ Primary VSM4, Secondary VSM4<br>■ Primary VSM3, Secondary VSM3<br>■ Primary VSM3, Secondary VSM4 (not recommended)<br>For Bi-Directional Clustering, all hosts must be at VTCS 6.1.0 or above to enable this feature. |
| 2 VTSSs within a cluster (FICON Interfaces) | The Primary and Secondary VTSSs can be any combination of VSM4 and VSM5 where the Secondary can be of any capacity. All hosts **must** be at VTCS 5.1.0 or above to enable this feature. For example, all of the following are valid:<br>■ Primary VSM5, Secondary VSM4<br>■ Primary VSM5, Secondary VSM5<br>■ Primary VSM4, Secondary VSM5<br>■ Primary VSM4, Secondary VSM5 (not recommended)<br>For Bi-Directional Clustering, all hosts must be at VTCS 6.1.0 or above to enable this feature. |
| Primary and Secondary VTSS microcode | The Primary VTSS microcode must be at a level that supports sending replicated VTVS. The Secondary VTSS microcode must be at a level that supports receiving replicated VTVS and supports the use of the Secondary as a production VTSS. After the microcode is installed, the Clustering feature must be enabled at **both** the Primary and Secondary VTSS via an options floppy disk.<br>See your StorageTek hardware service representative for details. |
| VTCS software | ■ VTCS 5.1.0 (for enhanced clustered support)<br>■ For Bi-Directional Clustering, all hosts must be at VTCS 6.1.0 or above to enable this feature.<br>■ The Advanced Management Feature (to enable the REPLICAT parameter of the MGMTclas statement) |

# How Clustered VTSS Configurations Work

You can use VSM to connect two VTSSs by Cluster Links (CLINKs) to form a *Clustered VTSS configuration*. You use the following statements to implement a Clustered Configuration:

- Clusters can be either Uni-Directional (Primary/Secondary VTSSs) or Bi-Directional (two Peer VTSSs).
- The Secondary VTSS (or the second Peer) can either be at the same physical location as the Primary (or first Peer) or at a remote location.
- The CONFIG CLUSTER statement specifies the two VTSSs that form the Cluster.
- The CONFIG CLINK statement defines the CLINKs that connect the two VTSSs. The way you write the CLINK statements determines whether the CLINKs are uni-directional or bi-directional. For examples, see Figure 13 on page 29 and Figure 21 on page 36.
- The MGMTclas REPLICAT parameter (which requires the Advanced Management Feature) identifies the Management Class that contains the VTVs that VSM *replicates* (copies) from one VTSS in the Cluster to the other.

---

**Note –** For VTCS 6.2 and above, the new CONFIG GLOBAL REPLicat parameter now specifies when to replicate a VTV as follows:

---

REPLicat

specifies when VSM replicates the VTV.

ALWAYS

The replicate request is added to the VTCS replication queue every time the VTV is dismounted, regardless of whether the VTV was changed while it was mounted (the default).

CHANGED

The replicate request is added to the VTCS replication queue if the VTV:

- Was changed while it was mounted **or**
- Was only read while mounted but less than the expected number of MVC copies of the VTV exist.

**Regardless** of the CONFIG GLOBAL REPLicat setting, replication **also** requires that:

- The VTV must be dismounted in a VTSS that supports replication **and** there cannot be an identical copy of the VTV in the other VTSS in the Cluster.
- In addition to the CONFIG GLOBAL REPLicat value, you **must** specify REPLICAT(YES) on a VTV's Management Class for replication to occur.

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

- VTCS immediately migrates (with KEEP) replicated VTVs. You can specify the source VTSS for migration of replicated VTVs on the MIGRATE parameter of the STORclas statement. **Also note** that you **must** specify replication on a Management Class **that points** to a Storage Class **with** a MIGRATE parameter value to migrate from the desired VTSS. Otherwise, migration from the desired VTSS does not occur.

Because VTCS immediately migrates (with KEEP) replicated VTVs regardless of the MGMTclas IMMEDmig setting, StorageTek **strongly recommends** that you **do not** explicitly set a MGMTclas IMMEDmig policy for replicated VTVs. If you do, VTCS honors the explicit immediate migrate request, and immediately migrates the affected VTV from whichever VTSS is first capable of performing the migration (that is, the first VTSS that has a resident VTV copy and an available RTD to satisfy the migrate). Setting an explicit MGMTclas IMMEDmig policy, therefore, is redundant and may interfere with optimal VTV replication and migration.

**Also note that** the immediate migrate (KEEP) following replication is **not the same** as automigration. That is, during the implicit immediate migrate, no VTVs are deleted from either VTSS to manage the DBU. Instead, the VTVs are simply "pre-staged" via migration to an MVC from the receiving VTSS, leaving both VTSS buffer contents unchanged. For space management in a VTSS cluster, VTCS automigrates VTVs according to the space management/migration cycle of **either** VTSS. If the capacity of the receiving VTSS is greater than or equal to that of the sending VTSS, automigration on the sending VTSS deletes a replicate VTV from **both** the VTSSs. If the capacity of the receiving VTSS is less than that of the sending VTSS, automigration may start on the receiving VTSS. In this case, automigration deletes a replicate VTV from only the receiving VTSS, leaving the copy on the sending VTSS still resident.

- **Note that** the replication requirements of data is determined following a dismount, **not** a recall. Merely recalling a VTV will not cause a replicate – so demand recall, MVCdrain and reclaim will not cause a replicate. However, if the VTV is recalled and mounted on a VTD, at dismount time it will be replicated to the Secondary or Peer VTSS.
- In dual-ACS environments, the same device types must be represented in the RTDs attached to each ACS so that data migrated by one VTSS can be recalled by the other VTSS. The number of MVCs, the media type and location used for the migration is determined by the MIGPOL parameter of the MGMTclas statement.
- Each CLINK uses a VTD address on the Secondary VTSS in a Uni-Directional Cluster (or both Peers in a Bi-Directional Cluster). These VTD addresses **are not** available to MVS; if you try to vary the VTSS online, you get an "assigned to another system" condition as follows:
  - **For VTCS 6.0 and below**, VSM3 uses the highest addresses of the first 8 VTDs defined for CLINKs; VSM4 uses the highest addresses of the first 16 VTDs defined. For example, in a **VSM3** you define four CLINKs and the following VTD addresses:

    VTD LOW=5800 HIGH=583F

    You **cannot** vary online to MVS the four highest of the first eight VTD addresses (5804 through 5807).

    In a **VSM4** you also define four CLINKs with the same VTD addresses:

    VTD LOW=5800 HIGH=583F

    In this case, you **cannot** vary online to MVS the four highest order of the first 16 VTD addresses (580C through 580F).
  - **For VTCS 6.1 and above**, VSM3s use 8 VTD addresses for CLINKs and VSM4s use 16 VTD addresses for CLINKs...but these addresses are not static. VTCS assigns addresses dynamically to prevent conflicts and errors. Do not, therefore, vary online to MVS any of the first 16 VTD addresses!.

- As described in "Clustered VTSS Operating Modes" on page 52, a Cluster can support different workloads in each of four operating modes. For example, only Full-Function Clusters can support active replication, but in Degraded Primary Mode, you can vary the Secondary's VTDs online to MVS to take over the workload. You can use Query to display Cluster, Cluster link, VTV replication, and VTSS status. You can use VARY VTSS to change VTSS states and VARY CLink to change CLINK states.

## Uni-Directional and Bi-Directional CLINKs

**For clustering**, you need a port in Host mode on one VTSS connected via a CLINK to a port in Nearlink mode on the other VTSS.

For example, FIGURE 3-19 shows 2 CLINK ports on each VTSS configured for Bi-Directional Clustering. On the Primary VTSS (VTSS1), the CLINK CIPs/FIPs are configured in **Nearlink Mode**, while on the Secondary VTSS (VTSS2), the CIPs/FIPs are configured in **Host Mode**. Figure 11 on page 26 shows a Uni-Directional Cluster attached to a dual ACS. In a Uni-Directional Cluster, the Primary VTSS replicates VTVs to the Secondary VTSS.

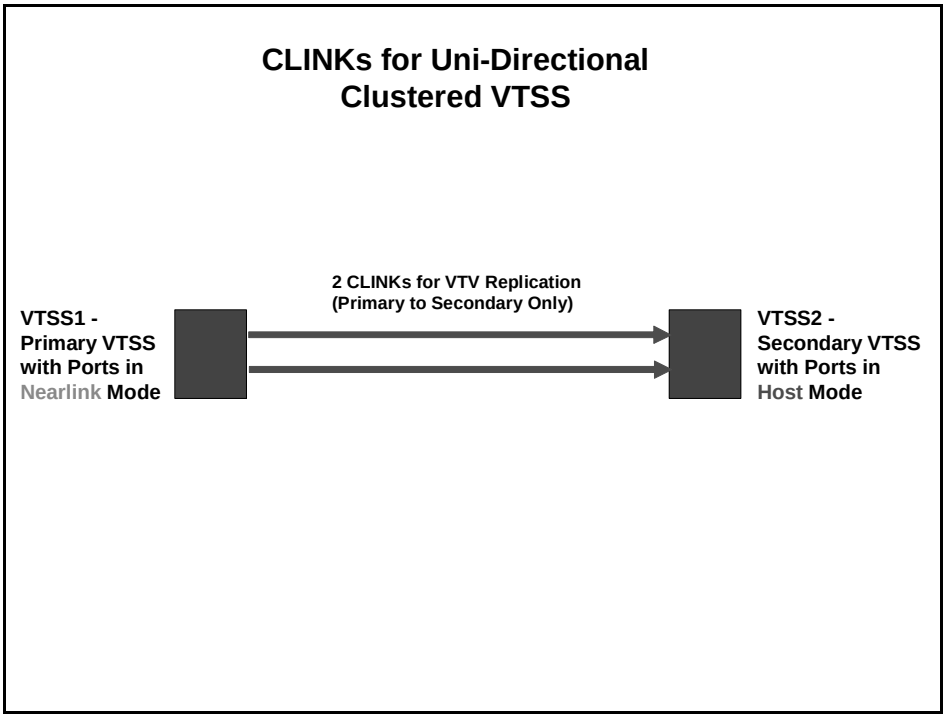



**FIGURE 3-19** CLINKs for Uni-Directional Clustered VTSS

FIGURE 3-20 shows 2 CLINK ports on each VTSS configured for Bi-Directional Clustering. **Each** Peer VTSS (VSMPR1 and VSMPR2), must have **both** of the following:

- **One** CLINK CIP/FIP configured in **Nearlink Mode** for replicating to the Peer.
- **One** CLINK CIP/FIP configured in **Host Mode** for receiving replicated VTVs from the Peer.

Bi-Directional Clustering, therefore, requires pairs of Uni-Directional CLINKs with the CIPs/FIPs configured so that the data flows in **opposite directions** on the CLINKs.

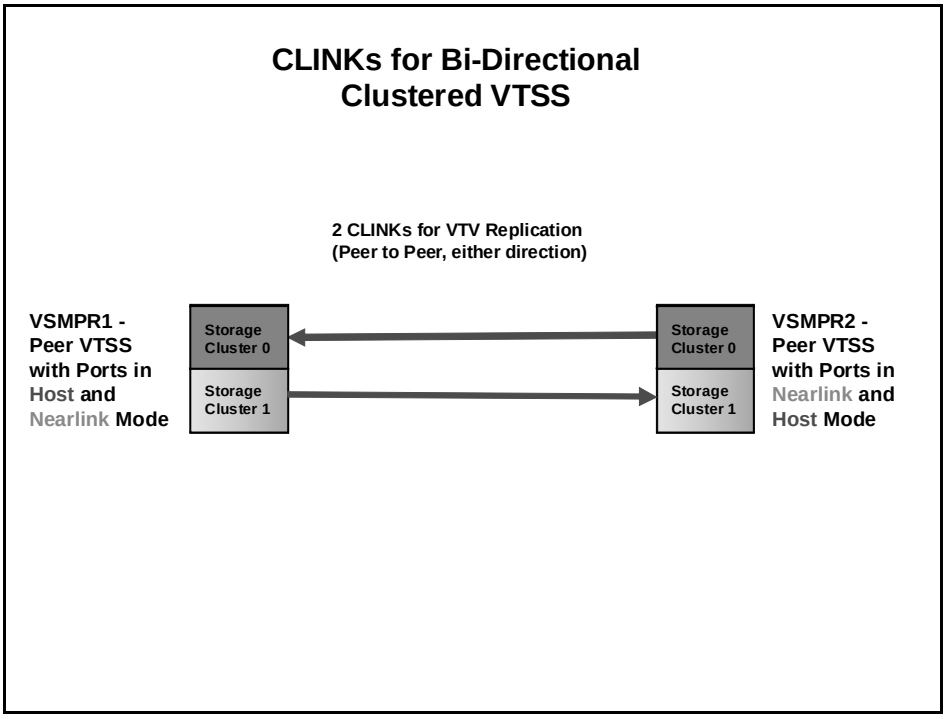



**FIGURE 3-20** CLINKs for Bi-Directional Clustered VTSS

## How Uni-Directional VTSS Clusters Work

**In a Uni-Directional Cluster**:

- The two VTSSs can be any combination of VSM3 and VSM4 where the Secondary can be of any capacity. All hosts **must** be at VTCS 5.1.0 or above to enable this feature. For example, all of the following are valid:
  - Primary VSM4, Secondary VSM3
  - Primary VSM4, Secondary VSM4
  - Primary VSM3, Secondary VSM3
  - Primary VSM3, Secondary VSM4 (not recommended)
- The Secondary can receive both replicated VTVs from the Primary and non-replicate production workload by any of the standard routing methods (for example, TAPEREQs). You need to vary the VTDs in the Secondary online to MVS so that the Secondary can accept production work. You **cannot** vary online to MVS the VTD addresses used by the CLINK terminations as described in "How Clustered VTSS Configurations Work" on page 44.
- A VTV with replication enabled is allocated to an online Primary VTSS unless none are available; in that case, the VTV is allocated to an online Secondary VTSS. If no online Secondary VTSSs are available, the VTV is allocated to a non-cluster VTSS. A VTV without replication can be allocated to any online VTSS including the Secondary of a Full-Function Cluster.
- At dismount time, a VTV with replication enabled that resides on a Full-Function Cluster is queued for replication to the Secondary VTSS. If a VTV with replication enabled is dismounted from a VTD in a VTSS that is not part of a Full-Function Cluster, the VTV is queued for immediate migration.

  When the Secondary VTSS receives a replicated VTV from the Primary VTSS, the VTV is then immediately migrated (with the KEEP option) regardless of Immediate Migrate Management Class settings for this VTV.
- **Both the Primary and the Secondary VTSS** can manage all space reclamations.

## How Bi-Directional VTSS Clusters Work

**In a Bi-Directional Cluster**, in normal operation, both VTSSs are online to VTCS as follows:

- The two VTSSs can be any combination of VSM3 and VSM4 and of any capacity. All hosts **must** be at VTCS 6.1.0 or above to enable this feature. For example, all of the following are valid:
  - Two VSM4s
  - Two VSM3s (not recommended)
  - VSM3 and VSM4 (not recommended)
- In a Bi-Directional Cluster, each of the Peer VTSSs can receive production work via the standard routing methods (for example, TAPEREQs). You need to vary the VTDs in both VTSSs online to MVS so that each can accept production work. However, **note that** you **cannot** vary online to MVS the VTD addresses used by the CLINK connections as described in "How Clustered VTSS Configurations Work" on page 44.
- **In a Bi-Directional Cluster**, a VTV with replication enabled is allocated to either of the Peer VTSSs. If one of the two Peer VTSSs is either offline or quiesced, production workload can run on the remaining online VTSS. VTVs requiring replication, however, are allocated to the remaining VTSS only if no other Full-Function clusters are available and suitable. In this case, replicate VTVs are migrated immediately with keep and queued for replication when the other VTSS comes online.
- **In a Bi-Directional Cluster**, at dismount time, a VTV with replication enabled that resides on a Full-Function Cluster is queued for replication to the other Peer VTSS. If a VTV with replication enabled is dismounted from a VTD in a VTSS that is not part of a Full-Function Cluster, the VTV is queued for immediate migration. **Note that** the replication requirements of data is determined following a dismount, **not** a recall. Merely recalling a VTV will not cause a replicate – so demand recall, MVCdrain and reclaim will not cause a replicate. However, if the VTV is recalled and mounted on a VTD, at dismount time it will be replicated to the Secondary VTSS
- **Both Peer VTSSs** can manage all space reclamations.

# Managing Clustered VTSS Systems

The following sections tell how to switch from Full-Function Mode to Degraded Primary or Degraded Secondary Mode, then return to Full-Function Mode:


- "Clustered VTSS Operating Modes" on page 52
- "Taking a Failed Primary VTSS Offline, Switching to the Secondary, Then Returning to Full-Function Mode" on page 55
- "Taking the Primary VTSS Offline, Switching to the Secondary, Then Returning to Full-Function Mode" on page 57
- "Taking a Failed Secondary VTSS Offline, Then Returning to Full-Function Mode" on page 59
- "Taking the Secondary VTSS Offline, Then Returning to Full-Function Mode" on page 60


---

**Note –** You can also use VARY CLink to do the following:

---

- Vary a CLINK offline if it has failed or requires service.
- Vary a CLINK online.

# Clustered VTSS Operating Modes

Active replication can only occur in a “Full-Function” Cluster where both VTSSs are online to VTCS. TABLE 3-2 describes Cluster operating modes and TABLE 3-3 on page 53 describes how VTCS manages VTVs in each of these modes.

**TABLE 3-2** Clustered VTSS Operating Modes

| Cluster Operating Mode | Primary VTSS State | Secondary VTSS State | Replication Workload Possible |
|---|---|---|---|
| **FULL-FUNCTION** | Online | Online | Workload goes to the Primary for VTV replication to the Secondary. |
| **DEGRADED** | n/a (both are Peer VTSSs) | n/a (both are Peer VTSSs) | One of the two peer VTSSs in a Bi-Directional Cluster is either offline or quiesced. Workload can run on the remaining online VTSS. VTVs requiring replication, however, are allocated to the remaining VTSS only if no other Full-Function clusters are available and suitable. In this case, replicate VTVs are migrated immediately with keep and queued for replication when the other VTSS comes online. |

**TABLE 3-2** Clustered VTSS Operating Modes

| | | | |
|---|---|---|---|
| **DEGRADED SECONDARY** | Online | Offline or Quiesced | Workload can run on the Primary. VTVs requiring replication, however, are allocated to the Primary only if no other Full-Function Clusters are available. If no Full-Function Clusters are available, replicate requests are allocated to non-clustered VTSSs and the replicate VTVs are migrated immediately with KEEP as described in Table 4 on page 50.<br>When the Secondary comes back online, VTCS replicates any VTVs requiring replication that are resident on the Primary but not resident on the Secondary. VTCS **does not** recall and replicate any VTVs already migrated and deleted by automigration.<br>Reconciliation also includes the migrate/delete actions described in Table 4 on page 50. The Cluster is now Full-Function again. |
| **DEGRADED PRIMARY** | Offline or Quiesced | Online | Workload can run on the Secondary. VTVs requiring replication, however, are allocated to the Secondary only if no other Full Function Clusters are available.<br>**Note:** When the Secondary takes over the workload, one VTD per Cluster link on the Secondary is reserved for replication and will not come online to MVS.<br>When the Primary comes back online, VTCS replicates any VTVs requiring replication that are resident on the Primary but not resident on the Secondary (that is, VTVs that were queued for replication but not yet replicated). VTCS **does not** recall and replicate any VTVs already migrated and deleted by automigration.<br>Reconciliation also includes the migrate/delete actions described in Table 4 on page 50. The Cluster is now Full-Function again. |
| **NON-OPERATIONAL** | Offline or Quiesced | Offline or Quiesced | No workload is possible on this Cluster. |

.

**TABLE 3-3** Clustered VTSS Operating Modes, VTV Allocations, REPLICAT Settings, and VTCS Actions

| If the Cluster Operating Mode is... | And the VTV is allocated to a VTSS of type... | And the REPLICAT setting is... | Then VTCS does the following at VTV dismount |
|---|---|---|---|
| **FULL-FUNCTION** | Primary | Yes | Replicates to Secondary then immediately migrates with KEEP from Secondary |
| **FULL-FUNCTION** | Primary | No | The VTV is managed by any other assigned Management Class attributes (for example, IMMEDMIG(KEEP)) |

**TABLE 3-3** Clustered VTSS Operating Modes, VTV Allocations, REPLICAT Settings, and VTCS Actions

| | | | |
|---|---|---|---|
| **FULL-FUNCTION** | Secondary | No | Immediately migrates with DELETE |
| **FULL-FUNCTION** | Non-cluster | No | Nothing |
| **FULL-FUNCTION** | Peer | Yes | Replicates to Peer VTSS then immediately migrates with KEEP from the VTSS designated by the STORCLASS MIGRATE policy |
| **DEGRADED SECONDARY** | Primary | Yes | Immediately migrates with KEEP |
| **DEGRADED SECONDARY** | Secondary | Yes | Not possible |
| **DEGRADED SECONDARY** | Non-cluster | Yes | Immediately migrates with KEEP |
| **DEGRADED SECONDARY** | Primary | No | Nothing |
| **DEGRADED SECONDARY** | Primary | Yes | Immediately migrates with KEEP and queues the VTV for replication. |
| **DEGRADED SECONDARY** | Secondary | No | Not possible |
| **DEGRADED SECONDARY** | Non-cluster | No | Nothing |
| **DEGRADED PRIMARY** | Primary | Yes | Not possible |
| **DEGRADED PRIMARY** | Secondary | Yes | Immediately migrates with KEEP |
| **DEGRADED PRIMARY** | Non-cluster | Yes | Immediately migrates with KEEP |
| **DEGRADED PRIMARY** | Primary | No | Not possible |
| **DEGRADED PRIMARY** | Secondary | No | Nothing |
| **DEGRADED PRIMARY** | Non-cluster | No | Nothing |
| **DEGRADED** | Peer | Yes | Immediately migrates with KEEP and queues the VTV for replication. It will be replicated when the Peer VTSS comes online. |

## ▼ Taking a Failed Primary VTSS Offline, Switching to the Secondary, Then Returning to Full-Function Mode

The following procedure tells how to take a failed Primary VTSS offline, switch to the Secondary, then return to Full-Function Mode.

**To take a failed Primary offline, switch to the Secondary, then return to Full-Function Mode, do the following**:

1. **Vary the Primary VTSS to Offline mode.**

   Message SLS6742I indicates when the VTSS is offline to all hosts.

---

**Note –** In a client/server environment (either MVS/CSC and LibraryStation or SMC client/server), VTCS cannot determine if long running jobs are active on the client hosts. Therefore, in addition to varying the Primary offline, you must also either explicitly vary the Primary's VTDs offline to MVS or ensure that there are no tape jobs active on the client hosts.

---

2. **Vary the VTDs in the Secondary online to MVS.**
3. **Ensure that MINMIG=1 on the Secondary.**
4. **Identify and recreate any unreplicated VTVs.**
   - An HSC SLS0906E Unable to mount message is issued by VTCS in response to an attempt to mount a VTV that is not available.
   - You can also run a VTV report with the UNAVAIL option to identify those VTVs that are now unavailable; that is, VTVs that are only resident and left mounted in VTSS1.
5. **When the Primary is available again, dismount any VTVs still mounted in the VTSS (MVS perspective), by doing either of the following:**
   - Use the MVS UNLOAD command to dismount the VTVs.
   - Use the VARY OFFLINE command to vary offline the VTD where the VTV is mounted, which will also dismount the VTV.
6. **Clear any boxed VTD conditions in the Primary.**
7. **Vary the Primary VTSS to Quiesced mode.**
8. **Audit the Primary VTSS.**

   When a VTSS is offline, a VTV can become "orphaned". That is, the VTV is resident in the VTSS but the VTV record in the CDS no longer has knowledge of this VTV image on the offline VTSS.

   When the VTSS goes to Quiesced mode, in Step 7, any orphaned VTVs occupy VTSS buffer space and automigrate does not remove them. The audit reconciles the orphaned VTVs as follows:

   - If the orphaned VTV is a current copy, the CDS is updated, and VTCS now manage it via automigration, recall, and so forth.
   - If the orphaned VTV is no longer current, the audit deletes it from the VTSS.

9. **Vary the Primary back online to VTCS and its VTDs back online to MVS.**

   Work will resume on the Primary. Note for a short period, work will also continue on the Secondary. No new allocations will use the Secondary VTSS but existing work will continue to completion (VTV mounts and dismounts will complete). Also note that VTCS will start to reconcile the Primary and Secondary VTSSs via migrates and deletes from the Secondary VTSS.

10. **Reset MINMIG=MAXMIG on the Secondary.**

11. **Vary the VTDs in the Secondary offline to MVS.**

    The VTDs will only go offline to MVS once all jobs that are allocated to VTDs in the secondary complete.

12. **Use Query CLINK to verify that the Cluster link is online.**

13. **Use Query CLUSTER to verify that the Cluster is in Full-Function Mode.**

## ▼ Taking the Primary VTSS Offline, Switching to the Secondary, Then Returning to Full-Function Mode

The following procedure tells how to take the Primary VTSS offline and switch to the Secondary (for service, replacement, and so forth, of the Primary), then return to Full-Function Mode when the Primary is available again.

**To explicitly take the Primary offline, switch to the Secondary, then return to Full-Function Mode, do the following**:

1. **Vary the Primary to Quiesced mode.**

   Message SLS6742I indicates when the VTSS is quiesced to all hosts.

2. **Do a demand migrate to 0.**

3. **Audit the Primary VTSS to ensure that all VTVs were successfully migrated.**

4. **Vary the Primary to Offline mode.**

   Message SLS6742I indicates when the VTSS is offline to all hosts.

---

**Note –** In a client/server environment (either MVS/CSC and LibraryStation or SMC client/server), VTCS cannot determine if long running jobs are active on the client hosts. Therefore, in addition to varying the Primary offline, you must also either explicitly vary the Primary's VTDs offline to MVS or ensure that there are no tape jobs active on the client hosts.

---

5. **Use Query CLUSTER to verify that the Cluster is in Degraded Primary Mode.**

6. **Vary the VTDs in the Secondary online to MVS.**

7. **Ensure that MINMIG=1 on the Secondary.**

8. **When the Primary is available again, vary it to Quiesced mode.**

9. **Audit the Primary VTSS.**

   When a VTSS is offline, a VTV can become "orphaned". That is, the VTV is resident in the VTSS but the VTV record in the CDS no longer has knowledge of this VTV image on the offline VTSS.

   When the VTSS goes to Quiesced mode, in Step 8, any orphaned VTVs occupy VTSS buffer space and automigrate does not remove them. The audit reconciles the orphaned VTVs as follows:

   - If the orphaned VTV is a current copy, the CDS is updated, and VTCS now manage it via automigration, recall, and so forth.
   - If the orphaned VTV is no longer current, the audit deletes it from the VTSS.

10. **Vary the Primary VTSS online to VTCS and vary the VTDs in the Primary VTSS online to MVS.**

    Work will resume on the Primary. Note for a short period, work will also continue on the Secondary. No new allocations will use the Secondary VTSS but existing work will continue to completion (VTV mounts and dismounts will complete). Also note that VTCS will start to reconcile the Primary and Secondary VTSSs via migrates and deletes from the secondary VTSS.

11. **Reset MINMIG=MAXMIG on the Secondary.**

12. **Vary the VTDs in the Secondary offline to MVS.**

    The VTDs will only go offline to MVS once all jobs that are allocated to VTDs in the secondary complete.

13. **Use Query CLINK to verify that the Cluster link is online.**

14. **Use Query CLUSTER to verify that the Cluster is in Full-Function Mode.**

## ▼ Taking a Failed Secondary VTSS Offline, Then Returning to Full-Function Mode

The following procedure tells how to take a failed Secondary VTSS offline, then return to Full-Function Mode.

**To take a failed Secondary offline, then return to Full-Function Mode, do the following**:

1. **Vary the Secondary to Offline mode.**

   Message SLS6742I indicates when the VTSS is offline to all hosts.

---

**Note –** In a client/server environment (either MVS/CSC and LibraryStation or SMC client/server), VTCS cannot determine if long running jobs are active on the client hosts. Therefore, if the Secondary is also accepting production work, in addition to varying the Secondary offline, you must also either explicitly vary the Secondary's VTDs offline to MVS or ensure that there are no tape jobs active on the client hosts.

---

2. **When the Secondary is available again, vary it to Quiesced mode.**
3. **Audit the Secondary VTSS.**
4. **Vary the Secondary to Online mode.**
5. **If you varied the Secondary's VTDs offline in** Step 1**, vary them online to MVS.**
6. **Use Query CLINK to verify that the Cluster link is online.**
7. **Use Query CLUSTER to verify that the Cluster is in Full-Function Mode.**

## ▼ Taking the Secondary VTSS Offline, Then Returning to Full-Function Mode

The following procedure tells how to take the Secondary VTSS offline (for service, replacement, and so forth), then return to Full-Function Mode when the Secondary is available again.

**To explicitly take the Secondary offline, then return to Full-Function Mode, do the following**:

1. **Vary the Secondary to Quiesced mode.**

   Message SLS6742I indicates when the VTSS is quiesced to all hosts.

2. **Audit the Secondary VTSS to ensure that all VTVs were successfully migrated.**

3. **Vary the Secondary to Offline mode.**

   Message SLS6742I indicates when the VTSS is offline to all hosts.

---

**Note –** In a client/server environment (either MVS/CSC and LibraryStation or SMC client/server), VTCS cannot determine if long running jobs are active on the client hosts. Therefore, if the Secondary is also accepting production work, in addition to varying the Secondary offline, you must also either explicitly vary the Secondary's VTDs offline to MVS or ensure that there are no tape jobs active on the client hosts.

---

4. **When the Secondary is available again, vary it to Quiesced mode.**

5. **Audit the Secondary VTSS.**

6. **Vary the Secondary to Online mode.**

7. **If you varied the Secondary's VTDs offline in** Step 1**, vary them online to MVS.**

8. **Use Query CLINK to verify that the Cluster link is online.**

9. **Use Query CLUSTER to verify that the Cluster is in Full-Function Mode.**

CHAPTER 4

# Merging and Consolidating Data Centers

Merging and consolidating data centers, a challenging task for sure. Fortunately, we've got the goods, namely EXPORT and IMPORT, which, thanks to the manifest file that EXPORT creates, let you create portable MVCs to move data from one VSM system to another.

## ▼ Example: Exporting by Management Class from the Source VSM System

This is the "send" phase of export/import, where we get the desired data packaged up and moved out of the source VSM system.

**To export from a source VSM system, do the following:**

1. **Identify the VTVs and/or MVCs that you want to export.**

   On this one, you have lots of choices, because you can export by:

   - **VTV volsers** - Use a TMS, ExLM, or VTVRPT report to identify the required VTVs.
   - **MVCs volsers** - Use a TMS, ExLM, or MVCRPT report to identify the required MVCs.
   - **Management Class** - Review your Management Class definitions to identify the required Management Classes.
   - **Storage Classes** - Review your Storage Class definitions to identify the required Storage Classes.

   Lots of granularity, but to me, the one that makes the most sense is Management Classes. Merging data centers is kind of like moving to a new house...you want to bring along all the valuable stuff, but the junk gets put out on the street for the trash service.

   So, accordingly, I'm going to decide that I don't need my production data (because I'm going to suspend production before the export/move), but I **do** want payroll and accounting information (Management Classes PAY and ACCOUNT).

2. **Run EXPORT to create a manifest file that lists the VTVs and MVCs available for export from the source VSM system:**

```
//EXPORT  EXEC PGM=SWSADMIN,PARM='MIXED' REGION=6M
//STEPLIBDD DSN=hlq.SLSLINK,DISP=SHR
//MOVE1DD DSN=FEDB.VSMLMULT.REMOTE2,DISP=(,CATLG,DELETE),
//             UNIT=SYSDA,SPACE=(CYL,(1,1),RLSE),
//             DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)
//SLSPRINTDD SYSOUT=*
//SLSINDD *
 EXPORT MGMT (PAY,ACCOUNT) MANIFEST(MOVE1)
```

**FIGURE 4-1** EXPORT by Management Class

In FIGURE 4-1, the output manifest file is MOVE1, which we'll need for the import. Just so you know what's going on under the covers, because we exported by Management Class, EXPORT consolidates (makes copies of) the selected VTVs on *export MVCs*. The export MVCs are marked as read-only and as exported in the CDS, and are now available for ejection from a source system LSM.

These copies are additional copies and are not recorded in the CDS. For example, if the VTV was duplexed before the export, the CDS records both duplexed copies, but the third additional copy used for consolidation is *not* recorded in the CDS. The original VTVs, therefore, are still available to the source system. You can use the data on the original VTVs or scratch and reuse them.

---

**Caution –** It pretty much goes without saying, but schedule the export for a time when the exported data is not being updated!

---

3. **Remove the MVCs for export from the MVC pool.**

   For more information, see *Managing VTCS*.

4. **Eject the MVCs for export from a source VSM system LSM.**

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

5. **If desired, at the source system, scratch or make unavailable the exported VTVs or reuse the data they contain.**

   After the export, the source system retains the CDS records of the exported VTVs and MVCs. The export MVCs are marked as exported and readonly in the CDS. Because you chose to export by Management Class, however, you can use the data on the original VTVs, scratch and reuse them, or make them unavailable. Use the HSC scratch utilities or the ExLM SYNCVTV function to scratch exported VTVs.

   Why would you want to do any of this? Let's say the actual move/import isn't going to happen for another month. There are other applications that, in the meantime, would like to take advantage of the virtual world in the source system. No problem...just scratch/reuse the "exported" VTVs.

   There is one other item you need to check, and that is What if one or more VTV volsers from the exporting system *already exist* in the importing system? We can't delete volser ranges, so this'll require some creative thinking.

   Probably the most straightforward approach is to look at the duplicate volsers in the TMS of the source system. Are any of them on the list to be scratched? Hopefully, they all are. Once the TMS scratches them, run scratch synchronization, then *don't write* to the VTVs again! A way of doing this would be to use MVCMAINT to temporarily mark the VTVs as readonly.

## ▼ Example: Importing by Management Class into the Target VSM System

It's a month later, and we're finally ready for the "receive" (import) portion of the export/import operation.

**To import into a target VSM system, do the following:**

1. **If the VTVs and MVCs you are importing are not in the target system CDS, rerun CONFIG to add these volsers...**

   ...as described in *Managing VTCS*. If you're actually merging or consolidating data centers, adding volsers is usually a gimme. If necessary, also increase the CDS size on the target VSM system.

   What if there were duplicate VTV volsers across the source and target systems? Back in Step 5 on  on page 63, we did some work to make the duplicates non-current, so we **won't** specify REPLACE in Step 5 on  on page 65.

2. **Prevent applications from creating or updating VTVs (and migrating them to MVCs) on the target VSM system.**

   In other words, quiesce/suspend virtual activity on the target system for all the good reasons in the world.

3. **Enter the MVCs for import into a target VSM system LSM.**

   For more information, see *HSC/MVS Operator's Guide*. Can you see what's going on here? You actually want to get the MVCs physically in place before you use IMPORT to tell the CDS that it has some new MVCs and VTVs.

4. **Optionally, do a "validate" run of IMPORT:**

```
//IMPORT  EXEC PGM=SWSADMIN,PARM='MIXED' REGION=6M
//STEPLIBDD DSN=hlq.SLSLINK,DISP=SHR
//REMOTE1DD DSN=FEDB.VSMLMULT.REMOTE1,DISP=SHR
//SLSPRINTDD SYSOUT=*
//SLSINDD *
 IMPORT MANIFEST(MOVEE1) NOUPDATE
```

**FIGURE 4-2** IMPORT utility example: validation import run, HSC is active

FIGURE 4-2 shows example JCL to run the IMPORT utility where:

- The Manifest File is the export Manifest specified in Step 2 on  on page 62.
- REPLACE(NONE) (the default) specifies that VTCS does not overwrite duplicate VTVs.
- IMMDRAIN(NO) (the default) specifies that VTCS does not drain all imported VTVs to VTSS space.
- NOUPDATE specifies that the CDS is not updated (validate run only).
- INACTCDS is not specified, so HSC is active.

Doing a validate run optional but **highly recommended**, because you really want to see what's going to happen before you push the button for real. Study carefully the Import Report. Like what you see? Continue with Step 5 on  on page 65.

---

**Note –**

- IMPORT is valid only if FEATures VSM(ADVMGMT) is specified; for more information, see “FEATURES Control Statement” on page 203.
- Ensure that the "to" CDS has the same features (enabled by CDS level) as the "from" CDS. For example, if the “from” CDS has Large VTV page sizes enabled and 2/4 Gb VTVs have been created, then the "to CDS" must have the same capabilities, otherwise the import fails.

---

5. **Do an actual run of IMPORT**:

```
//IMPORT  EXEC PGM=SWSADMIN,PARM='MIXED' REGION=6M
//STEPLIBDD DSN=hlq.SLSLINK,DISP=SHR
//REMOTE1DD DSN=FEDB.VSMLMULT.REMOTE1,DISP=SHR
//SLSPRINTDD SYSOUT=*
//SLSINDD *
  IMPORT MANIFEST(MOVEE1)
```

**FIGURE 4-3** IMPORT utility example: validation import run, HSC is active

FIGURE 4-3 shows example JCL to run the IMPORT utility where, per Step 1 on on page 64, we’re not overwriting the duplicate VTVs, so we do **not** specify REPLACE(ALL).

---

**Note –** What if you want to return the MVCs to the source system? Maybe you some or all of them there for whatever reason. If so, you can specify IMMDRAIN(YES) to drain the import MVCs.

---

6. **Adjust your VTV definitions as needed.**

   For example, you need to define any new VTVs to the target system’s TMS, and re the duplicate VTVs, you need to run VTVMAINT READONLY OFF to ensure that they are writable.

7. **Do one of the following**:
   - Optionally, run MVCMAINT to make imported MVCs writable. VTCS imports MVCs as readonly. To make them writable, you run MVCMAINT, specifying READONLY OFF. The chances are that you are going to want to use the new MVCs at the target system, and this is the first step.

     Next, add the imported MVCs to the MVC pool as described in . At this point, the MVCs can be reclaimed, drained, migrated to, recalled from, and so forth.
   - If you specified IMMDRAIN(YES) in Step 5, you can return the MVCs to the source system.

   That’s it! Take a break before you fire up your new system for production...

APPENDIX **A**

# Additional Information

# 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

The following are used in this book.

**Warning.** Information necessary to keep you from damaging your hardware or software.

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

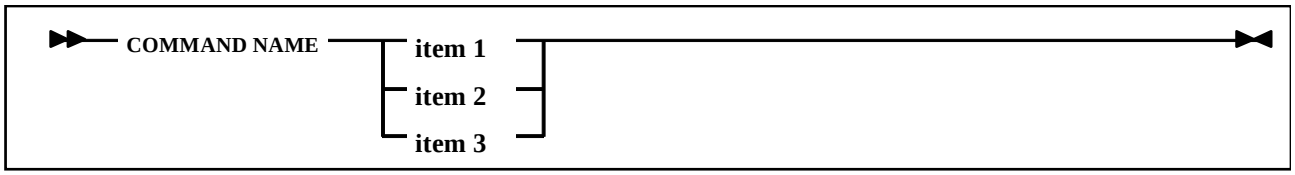
**Tip –** Information that can be used to shorten or simplify your task or they may simply be used as a reminder.

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

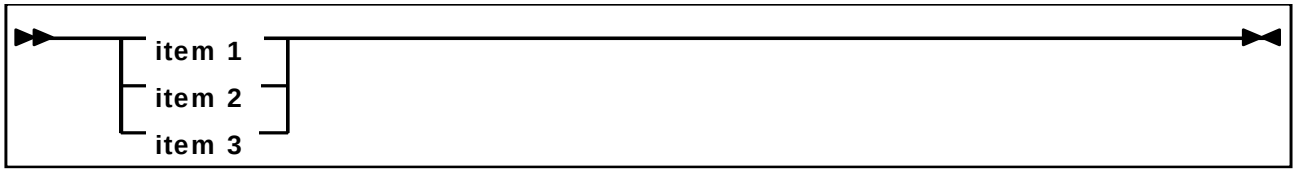
## Syntax

Syntax flow diagram conventions include the following:

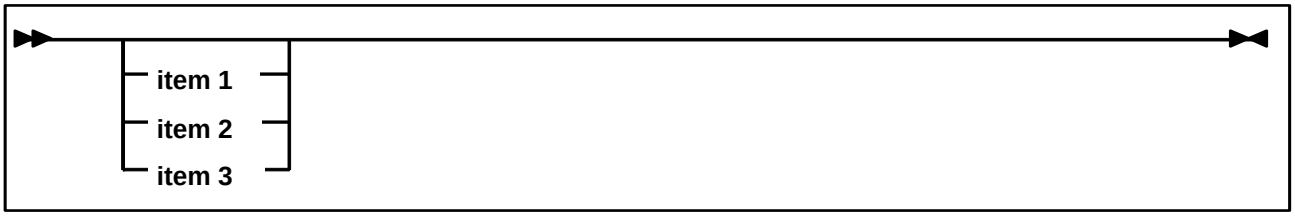
**Flow Lines—**Syntax diagrams consist of a horizontal baseline, horizontal and vertical branch lines and the command text. Diagrams are read left to right and top to bottom. Arrows show flow and direction.



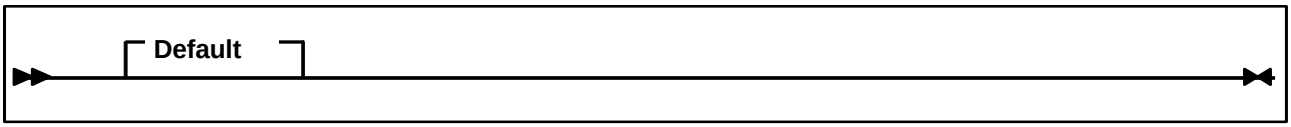

**Single Required Choice—**Branch lines (without repeat arrows) indicate that a single choice must be made. If one of the items to choose from is on the baseline of the diagram, one item must be selected.



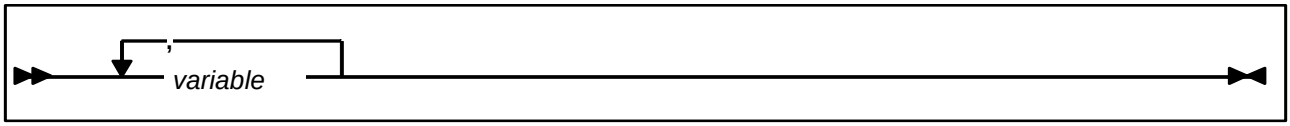

**Single Optional Choice—**If the first item is on the line below the baseline, one item may optionally be selected.



**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.

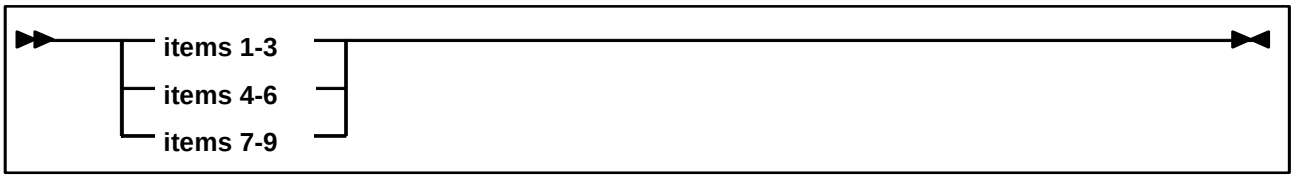
**Variables—**Italic type is used to indicate a variable.

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

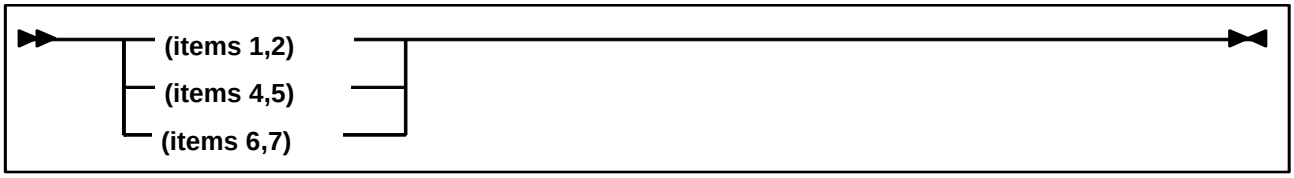
**Optional—**Brackets [ ] are used to indicate that a command parameter is optional.

**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.



# Additional Information for Sun Products

Sun Microsystems, Inc. (Sun) offers several methods for you to obtain additional information.

## Sun's External Web Site

Sun'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 Sun external Web site is: http://www.sun.com

The URL for Sun StorageTek™ brand-specific information is:
http://www.storagetek.com

## Customer Resource Center

The Sun StorageTek product Customer Resource Center (CRC) is a Web site that enables members to resolve technical issues by searching code fixes and technical documentation for StorageTek brand products. CRC membership entitles you to other proactive services, such as HIPER subscriptions, technical tips, answers to frequently asked questions, addenda to product documentation books, and online product support contact information. Customers who have a current warranty or a current maintenance service agreement may apply for membership by clicking on the Request Password button on the CRC home page. Sun employees may enter the CRC through the SunWeb PowerPort.

The URL for the CRC is http://www.support.storagetek.com

## Partners Site

The StorageTek Partners site is a Web site for partners with a StorageTek Partner Agreement. This site provides information about products, services, customer support, upcoming events, training programs, and sales tools to support StorageTek Partners. Access to this site, beyond the Partners Login page, is restricted. On the Partners Login page, Sun employees and current partners who do not have access can request a login ID and password and prospective partners can apply to become StorageTek resellers.

The URL for the StorageTek Partners site is:
http://members.storagetek.com

The URL for partners with a Sun Partner Agreement is:
http://www.sun.com/partners/

## Third-Party Web Sites

Sun is not responsible for the availability of third-party web sites mentioned in this document. Sun 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. Sun 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.

## Hardcopy Publications

Contact a Sun sales or marketing representative to order additional paper copies of this publication or to order other StorageTek brand product customer publications in paper format.

## Customer Support

Customer support is available 24 hours a day, seven days a week, to customers with Sun or StorageTek maintenance contracts and to Sun employees. You can find additional information about customer support on the Customer Resource Center (CRC) Web site at:
http://www.support.storagetek.com

## Customer-initiated Maintenance

Customer-initiated maintenance begins with a telephone call from you to Sun Microsystems StorageTek Support. You receive immediate attention from qualified personnel, who record problem information and respond with the appropriate level of support.

To contact Sun Microsystems StorageTek Support about a problem:

1. Use the telephone and call:

   **800.872.4786** (1.800.USA.4Sun)

   **800.722.4786** (Canada)

   For international locations, go to
   http://www.sun.com/service/contacting/solution.html
   for the appropriate telephone number

2. Describe the problem to the call taker. The call taker will ask several questions and will either route your call to or dispatch a support representative.

   If you have the following information when you place a service call, the process will be much easier:

| | |
|---|---|
| Account name | |
| Site location number | |
| Contact name | |

| | |
|---|---|
| Telephone number | |
| Equipment model number | |
| Device address | |
| Device serial number (if known) | |
| Urgency of problem | |
| Fault Symptom Code (FSC) | |
| Problem description | |
| | |
| | |
| | |

## Sun's Worldwide Offices

You may contact any of Sun's worldwide offices to discuss complete storage, service, and support solutions for your organization. You can find address and telephone number information on 's external Web site at:
http://www.sun.com/worldwide/

**Sun Microsystems, Inc.** 4150 Network Circle, Santa Clara, CA 95054 USA **Phone** 1-650-960-1300 or 1-800-555-9SUN **Web** sun.com

**Argentina:** 5411-4317-5636 • **Australia:** 1-800-550-786 • **Austria:** 43-1-601-26-0 • **Balkans:** 301-6188-111 • **Belgium:** 32 2-704 89 83 • **Brazil:** 55-11-51872100 • **Brunei:** 65-216-8333 • **Canada:** 1-800-422-8020 **(General);** 416-964-2001 **(Learning Management System Sales,Toronto)** • **Chile:** 562-372-4500 • **Colombia:** 571-629-2323 **Czech Republic:** 420 2 33009311 • **Denmark:** 45 4556 5040 • **Egypt:** 00 202 570 9442 • **Finland:** 358-9-525-561 • **France:** 33-1-41-33-17-17 • **Germany:** 49-89-460-08-2788 • **Greece:** 30-01-6188101 • **Hong Kong:** 852-2877-7077 • **Hungary:** 361-202-4415 • **India:** 91-80-229-8989 • **Indonesia:** 65-216-8333 • **Ireland:** 353-1-668-4377 **Israel:** 972-9-9710500 • **Italy:** 39-02-9259511 • **Japan:** 81-3-5779-1820 • **Korea:** 82-2-3453-6602 • **Malaysia:** 603-2116-1887 • **Middle East:** 00 9714 3366333 • **Mexico:** 525-261-0344 • **Netherlands:** 31-33-4515200 • **New Zealand:** 0800-786-338 • **North West Africa:** 00 9714 3366333 • **Norway: From Norway:** 47-22023950, **To Norway:** 47-23369650 • **Pakistan:** 00-9714-3366333 • **People's Republic of China:** 8610-6803-5588 • **Philippines:** 632-885-7867 • **Poland:** 48-22-8747848 • **Portugal:** 351-21-413-4000 • **Russia:** 7-095-935-8411 • **Saudi Arabia:** 00 9714 3366333 • **Singapore:** 65-216-8300 • **South Africa:** 27-11-256-6300 • **Spain:** 34-902-210-412 • **Sri Lanka:** 65-2168333 • **Sweden:** 46-8-631 22 00 • **Switzerland:** 41-1-908-90-50 **(German)** 41-22-999-0444 **(French)** • **Taiwan:** 886-2-25185735 • **Thailand:** 662-344-6855 • **Turkey:** 90 212 335 22 00 • **United Kingdom:** 44-1276-416-520 • **United States:** 1-800-422-8020 • **Venezuela:** 582-905-3800 • **Vietnam:** 65-216-8333 • **Worldwide Headquarters:** 1-650-960-1300