StorageTek

Virtual Tape Control System

Installation and Configuration Guide

Version 6.1.0

CRC Update Only

Revision F - April 2007

CRC Update Only

This revision applies to Version 6.1.0 of the Virtual Tape Control System software. Information in this publication is subject to change. Send comments about this publication to:

E-mail us at: `slsfs@sun.com`

About this Book

Virtual Tape Control System 6.1.0 (VTCS 6.1.0, hereafter referred to as "VTCS") is MVS host software, which together the portions of NCS 6.1.0 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 and VSM. 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

Also see: *VTCS Administrator's Guide* for information about VTCS administration tasks.

Reader's Comments

If you have comments on this book, please e–mail us at **slsfs@sun.com** and include the document title and number with your comments.

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 the Software

This guide applies to VTCS 6.1.0 and NCS 6.1.0 and above. VTCS executes in the native MVS or OS390 environment and does not use or require OS390 OpenEdition services.

How this Guide is Organized

This guide contains the following sections:

What's New in This Guide?

Revision F

The Revision F of this reference contains the new information or updates described in Table 1.

Table 1. Updates to VTCS Command and Utility Reference, Revision F

This information...	...is described in...	...and is available via PTF...
T10000 encryption transports and media	• "Using T10000 Drives as RTDs" on page 217 • "VTCS Considerations to Correctly Specify MVC Media" on page 26	6.1 with the following PTFs: • L1H13AA (SOS6100) • L1H136K (SWS6100) 6.2 with the following PTFs: • L1H139D (SOS6200) • L1H139C (SWS6200)

Revision E

The Revision E of this reference contains the new information or updates described in Table 2.

Table 2. Updates to VTCS Command and Utility Reference, Revision E

This information...	...is described in...	...and is available via PTF...
VSM5 support	"VSM5 Configuration" on page 193	L1H12ZT (SWS6100). **Note that** this PTF changes `QUERY/DISPLAY VTSS` to report the VTSS capacity in gigabytes (Gb).
Swapping out a VTSS and replacing it with another VTSS	"Swapping VTSSs" on page 211	N/A
T10000 drives as RTDs	"Using T10000 Drives as RTDs" on page 217	• L1H12ZN (SOS6100) • L1H12ZO (SWS6100)

VTCS 6.1.0, Revision D

The VTCS 6.1.0, Revision D of this guide contains technical updates and corrections to "Converting the Formatted CDS to VSM Extended Format" on page 83.

VTCS 6.1.0, Revision C

The VTCS 6.1.0, Revision C of this guide contains technical updates and corrections.

VTCS 6.1.0, Revision B

The VTCS 6.1.0, Revision B of this guide contains information about the VTCS 6.1 enhancements described in Table 3.

Table 3. VTCS 6.1.0 Updates to VTCS Installation and Configuration Guide, Revision B

This SPE...	...is described in...	...and is available via the following PTFs...
VTCS Locks in a Coupling Facility	"Storing VTCS Locks in a Coupling Facility (Optional)" on page 114	L1H12J4 (SWS6100) and L1H12J3 (SOS6100) **Note:** If VTCS locks are held in a Coupling Facility structure, these PTFs must be installed on all hosts as described in "All Hosts PTFs" on page 2.

Revision A

The Revision A of this guide contains the updates described in Table 4.

Table 4. VTCS 6.1.0 Updates to VTCS Installation and Configuration Guide, Revision A

This Enhancement...	...is described in...
VSM4 FICON Back-End Support	"VSM4 FICON Front-End and Back-End Configuration" on page 175

VTCS 6.1.0, Initial Release

The VTCS 6.1.0, Initial Release of this guide contains information about the VTCS 6.1 enhancements described in Table 5.

Table 5. VTCS 6.1.0 Updates to VTCS Installation and Configuration Guide, Initial Release

This Enhancement...	...is described in...
Near Continuous Operations	• "HSC CDS DASD Space" on page 43 • "Converting the Formatted CDS to VSM Extended Format" on page 83
Bi-Directional Clustering	"Converting the Formatted CDS to VSM Extended Format" on page 83
SMC enhancements	• "Note" on page 14 • "Note" on page 14 • "Note" on page 15 • "VSM Esoterics and Esoteric Substitution" on page 17 • "Note" on page 24 • "Hint" on page 81
VSM4 FICON Front-End support	Replaced by "VSM4 FICON Front-End and Back-End Configuration" on page 175

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 lENTERm.

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

Symbols

The following symbols are used to highlight text 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.

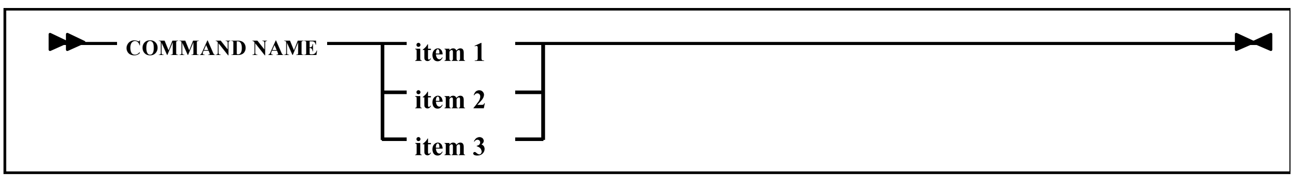
Hint: Information that can be used to shorten or simplify your task or they may simply be used as a reminder.

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

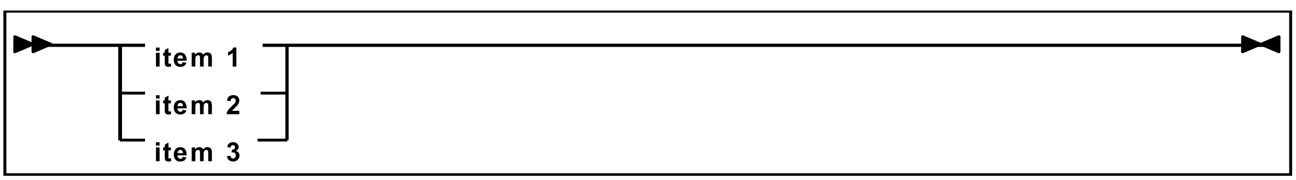
Syntax

Syntax flow diagram conventions include the following:

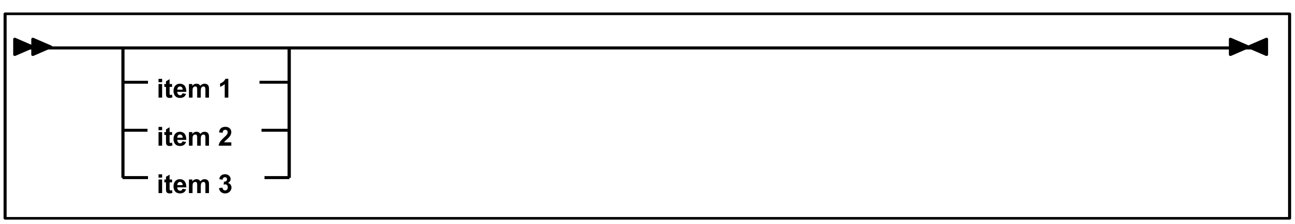
Flow Lines—Syntax diagrams consist of a horizontal baseline, horizontal and vertical branch lines and the command text. Diagrams are read left to right and top to bottom. Arrows show flow and direction.



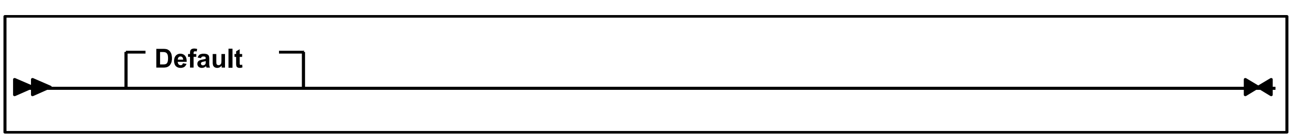
Single Required Choice—Branch lines (without repeat arrows) indicate that a single choice must be made. If one of the items to choose from is on the baseline of the diagram, one item must be selected.



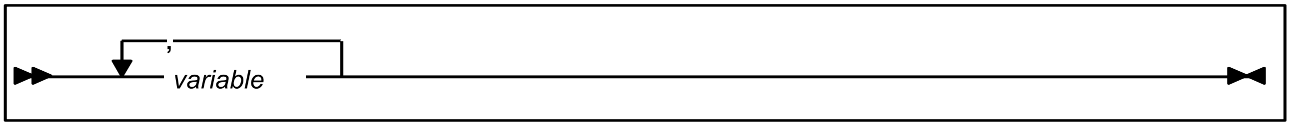
Single Optional Choice—If the first item is on the line below the baseline, one item may optionally be selected.



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.

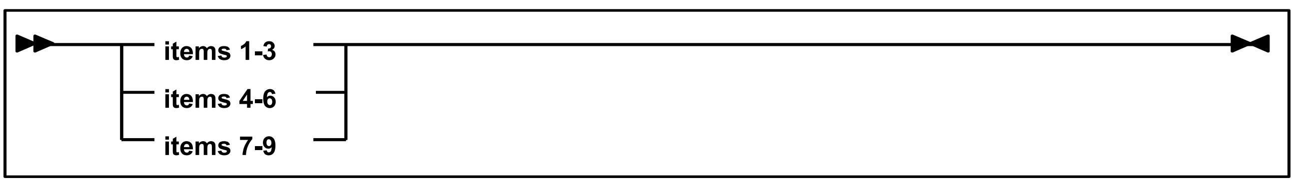
Variables—Italic type is used to indicate a variable.

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

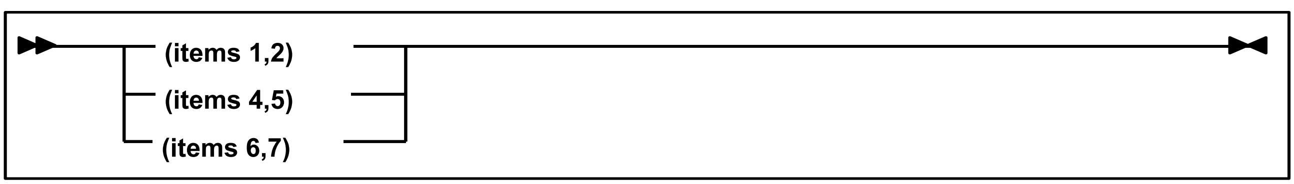
Optional—Brackets [ ] are used to indicate that a command parameter is optional.

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.



Related Publications

The following publications provide additional information about VSM and StorageTek's Automated Cartridge System software and hardware.

VTCS and VSM

The VTCS and VSM documentation set consists of the following:

- *Introduction to VSM*, which you can request from your StorageTek representative
- The VTCS Information CD-ROM, which contains PDF file formats of *Virtual Tape Control System Installation and Configuration Guide*, *Virtual Tape Control System Command and Utility Reference, Virtual Tape Control System Administrator's Guide, Virtual Tape Control System Messages,* and *Virtual Tape Control System XML Reference*
- *Virtual Tape Control System Installation and Configuration Guide* (this book)
- *Virtual Tape Control System Command and Utility Reference*
- *Virtual Tape Control System Administrator's Guide*
- *Virtual Tape Control System Messages*
- *Virtual Tape Control System Quick Reference*
- *Virtual Tape Control System XML Reference*
- *VSM Offsite Vault Disaster Recovery Guide* (supplied with the VSM Offsite Vault Disaster Recovery Feature)

VTSS

- *Virtual Storage Manager Planning, Implementation, and Usage Guide*
- *Virtual Storage Manager Physical Planning Guide*
- *VTSS Installation Guide*

NCS

- *NCS Installation Guide*
- *SMC Administration and Configuration Guide*

HSC-MVS Environment

- *Configuration Guide*
- *Operator's Guide*
- *System Programmer's Guide*
- *Messages and Codes*
- *System Programmer's Reference Summary*
- *Operator's Reference Summary*

LibraryStation

- *Configuration Guide*
- *Operator and System Programmer's Guide*
- *Messages and Codes*

MVS/CSC

- *Configuration Guide*
- *Operator Guide*
- *System Programmer Guide*
- *Messages and Codes*

ExPR

- *Introduction to ExPR*
- *ExPR SMP/E Installation*
- *ExPR MVS Configuration*
- *ExPR MVS Reports*
- *ExPR MVS Reference*

ExLM 4.0.0

The ExLM 4.0.0 documentation set consists of the following:

- The ExLM 4.0.0 Information CD-ROM, which contains PDF file formats of *ExLM Installation Guide*, *ExLM System Administrator's Guide*, *ExLM System Administrator's Guide - Field Tables Supplement,* and *ExLM Messages and Codes*
- *ExLM Installation Guide*
- *ExLM System Administrator's Guide*
- *ExLM System Administrator's Guide - Field Tables Supplement*
- *ExLM Messages and Codes*
- *ExLM Quick Reference*

ExLM 5.0.0

The ExLM 5.0.0 documentation set consists of the following:

- The ExLM 5.0.0 Information CD-ROM, which contains PDF file formats of the ExLM publications
- *ExLM Installation Guide*
- *ExLM System Administrator's Guide*
- *ExLM Messages and Codes*
- *ExLM Quick Reference* (includes information formerly provided in the *ExLM 4.0.0 System Administrator's Guide - Field Tables Supplement*)

IBM Publications

- *IBM ESA/390 Common I/O-Device Commands and Self Description*
- *IBM 3490 Magnetic Tape Subsystem*
  Models A01, A02, A10, A20, B02, B04, B20, and B40
  Introduction
- *IBM 3490 Magnetic Tape Subsystem*
  Models A01, A02, A10, A20, B02, B04, B20, and B40
  Hardware Reference
  (Referred to in this book as the *IBM 3490 Hardware Reference*)
- *IBM 3490 Command Reference*
- *IBM 3480 Magnetic Tape Subsystem Reference*
- *IBM 3480 Installation Guide and Reference*
- *OS/390 V2R4.0 MVS Planning: Global Resource Serialization*
- *MVS Authorized Assembler Services Guide*

Online Documentation on the StorageTek CRC

The StorageTek Customer Resource Center (CRC) on the World Wide Web provides online versions in PDF format of this book, the related StorageTek publications listed on page x, and many other StorageTek software and hardware publications.

To access PDF documents on the StorageTek CRC:

1. **Using an Internet browser such as Netscape or Internet Explorer, go to the StorageTek CRC at:**

   `http://www.support.storagetek.com/`

2. **Click the** `Login` **link.**
3. **Fill in the login information.**

   If this is the first time you have used the CRC, click `Request a CRC password` and fill in the requested information. You should receive your account information within two business days.

4. **From the upper left bar, click** `Product Information` **and** `Current Products` **from the dropdown links.**
5. **Select** `Software` **from the Product Family dropdown menu and click** `Next`**.**

   Click the desired product link from the Product Categories and navigate to the documents you want to view.

Technical Support

Refer to *Requesting Help from Software Support* for information about contacting StorageTek for technical support and for requesting changes to software products.

Document Effectivity

EC Number	Date	Doc Kit Number	Type	Effectivity
132208	February 2005	---	Initial Release	This document applies to VTCS, Version 6.1.0.
---	June 2005	---	Revision A	This document applies to VTCS, Version 6.1.0.
---	August 2005	---	Revision B	This document applies to VTCS, Version 6.1.0.
---	August 2005	---	Revision C	This document applies to VTCS, Version 6.1.0.
---	August 2005	---	Revision D	This document applies to VTCS, Version 6.1.0.
---	September 2006	---	Revision E	This document applies to VTCS, Version 6.1.0.
---	April 2007	---	Revision F	This document applies to VTCS, Version 6.1.0.

Contents

List of Figures

List of Tables

Chapter 1. Planning for Installation

Before doing the tasks described in Chapter 2 “Preparing for Installation”, complete the planning tasks described in the following sections:

- “Verifying VSM Software and Hardware Prerequisites” on page 10.
- “Determining VSM Configuration Values” on page 14. Use Table 25 on page 125 to record these VSM configuration values, which you use to complete the tasks in this guide.

Use “VTCS Installation Summary and Checklist” on page 3 to help plan and verify completion of your system’s installation and configuration tasks.

Hint: Your StorageTek representatives will help you plan and install the VTSS hardware. If you are adding Nearline hardware, such as LSMs or transports to be used as RTDs, StorageTek will also help you install and configure this hardware.

To plan the VTSS configuration and install and configure your VTSS hardware, StorageTek will use the VTSS publications on page x and the VSM pre-sales planning tool. This tool produces data to help define the optimum VSM solution for your business needs and to identify VSM candidate data sets.

Your StorageTek representatives will also help coordinate the installation and configuration of the VSM hardware with the software installation and configuration tasks described in this guide. For example, before you can verify the updated HSC LIBGEN as described in “Verifying the LIBGEN” on page 82, if your system’s RTDs are new transports and you will share them between VSM and MVS, you must install them and define their MVS unit addresses via the `HCD` facility.

All Hosts PTFs

In the future, PTFs that change the CDS in such a way that it can not be processed by systems without the PTF applied will make use of the CDS "feature string".

The feature string:

- Is a byte within the CDS.
- Indicates which features are in use that require the PTF to be installed on all Hosts.

When PTFs are installed appropriately, this technique has no visible impact.

HSC/VTCS systems without an "all-Host" PTF applied will be unable to process the CDS successfully. The result will be the following messages:

- `SLS6664E CDS level is not compatible with VTCS`, or
- `SLS6818E The CDS contains an unrecognised feature string (X'`*`hh`*`')`

The feature string is set to the following value in the following situations:

- X'80': VTCS locks are implemented in a coupling facility structure (the VTCS configuration specifies `GLOBAL LOCKSTR=`*`structure-name`*)
- X'40': reserved

VTCS Installation Summary and Checklist

Use the checklist in Table 6 to help plan and verify completion of your system's installation and configuration tasks.

Table 6. VTCS Installation Summary and Checklist

<table>
<tr><th>Task</th><th>Notes</th><th>✓ to Verify Completion</th></tr>
<tr><td>“Verifying VSM Software and Hardware Prerequisites” on page 10</td><td>Ensure that you have the prerequisites. for VTCS 6.1 and the features and hardware you intended to use.</td><td></td></tr>
<tr><td>“Determining VSM Configuration Values” on page 14

“HSC CDS DASD Space” on page 43</td><td>Plan configuration values, such as but not limited to your implementation of 4 VTV Copies and 800 Mb VTVs.

Note especially! The VSM Extended Format CDS requires additional DASD space!</td><td></td></tr>
<tr><td colspan="3">“Preparing for Installation” on page 65</td></tr>
<tr><td>“Defining A Security System User ID for HSC, SMC, and VTCS” on page 66</td><td rowspan="4">None of these tasks is difficult, but they are all critical. For example, Depending on the default settings of your security system, VSM may not be able to mount and to write to MVCs until you have defined a security system user ID for HSC and TAPEVOL profiles for the MVCs!</td><td></td></tr>
<tr><td>“Configuring MVS Device Numbers and Esoterics” on page 66</td><td></td></tr>
<tr><td>“Setting the MVS Missing Interrupt Handler (MIH) Value” on page 67</td><td></td></tr>
<tr><td>“Specifying the Region Size” on page 67</td><td></td></tr>
</table>

Table 6. VTCS Installation Summary and Checklist

Task	Notes	✓ to Verify Completion
"Installing HSC and VTCS" on page 69		
"Reviewing Coexistence Requirements" on page 70	Basic SMP/E installation. **Note that** order is important, and you have to coordinate VTCS installation with NCS (including SMC) installation.	
"Verifying Installation Materials" on page 70		
"Installing HSC and SMC" on page 72		
"Receiving the VTCS 6.1.0 FMID" on page 73		
"Receiving the VTCS 6.1.0 Service" on page 73		
"Creating the VTCS 6.1.0 LINKLIB Data Sets and Defining Libraries to the HSC Target Zone" on page 74		
"Applying the VTCS 6.1.0 FMID" on page 75		
"Accepting the VTCS 6.1.0 FMID" on page 76		
"Applying the VTCS 6.1.0 Service" on page 77		
"Adding SWSLINK to the Authorized Program List" on page 78		
"Modifying the HSC Startup Procedure to include the VTCS 6.1.0 LINKLIB" on page 78		

Table 6. VTCS Installation Summary and Checklist

Task	Notes	✓ to Verify Completion
"Reconfiguring NCS" on page 79		
"Creating or Updating the HSC LIBGEN" on page 80	**Note that**: • If your system's RTDs are new transports, you must update the HSC `LIBGEN` by adding a `SLIDRIVS` macro to define the device addresses you determined. • If you are converting the CDS to VSM Extended Format as described in "Converting the Formatted CDS to VSM Extended Format", you **must** create a new CDS.	
"Verifying the LIBGEN" on page 82	...as with any new or updated LIBGEN...	
"Formatting the New CDS" on page 82	This is where you use the HSC `SLICREAT` macro to format the new VSM Extended Format CDS to the larger size you determined in the planning section.	
"Converting the Formatted CDS to VSM Extended Format" on page 83	**This is the key to successful conversion**. The heart of this section is the VTCS 4.0 to VTCS 6.1 conversion matrix that tells you how to go from VTCS/NCS 4.0, 5.0, 5.1, or 6.0 to VTCS 6.1.	

Table 6. VTCS Installation Summary and Checklist

Task	Notes	✓ to Verify Completion
"Updating the HSC Definition Data Sets" on page 95	You might not have to update any of these data sets...but if you're doing anything with Management and Storage Classes (4 VTV copies, for example), you'll need to do some work.	
"Creating and Using VSM Management and Storage Classes" on page 96	The "how to", with examples of the basics of Storage and Management Classes.	
"Updating the HSC PARMLIB Member (SLSSYSxx)" on page 104	This is where your "DEF" statements reside, plus other critical items such as the COMMPATH and FEATURES statements. As above, don't overlook this step...	
"Adding SMF Parameters for VTCS to SYS1.PARMLIB" on page 105	Technically speaking, this is optional...but **highly recommended**, because you need the SMF information to know how your system is performing.	

Table 6. VTCS Installation Summary and Checklist

Task	Notes	✓ to Verify Completion
“Connecting MVS/CSC Clients to VSM” on page 106	You only need to do this if you are using MVS/CSC on remote hosts. **Note that**, for VTCS/NCS 6.1, you can simply install SMC 6.1 in your client MVS system, and SMC will route virtual allocation and mount requests to HSC running in a remote server HSC system. For more information, see the *SMC 6.1 Configuration and Administration Guide*.	
“Connecting Non-MVS/CSC Clients to VSM” on page 108	You only need to do this if you want to connect non-MVS/CSC 4.0 and above clients to VSM and define LibraryStation subpools that contain VTVs. Contact StorageTek Software Support for information on the supported clients.	
“Restarting NCS” on page 109	...to make the NCS reconfiguration take effect...	

Table 6. VTCS Installation Summary and Checklist

Task	Notes	✓ to Verify Completion
"Configuring VSM" on page 111		
"Defining VSM Security" on page 112	Required to ensure that the correct personnel and applications have access to the VSM resources required.	
"Configuring VTCS" on page 114	This is where you run VTCS `CONFIG`. This section in the ICG is really just a reminder note; the guts of what you need to know about `CONFIG` is in the *VTCS 6.1 Command and Utility Reference*.	
"Running the HSC MERGECDS Utility" on page 120	If you are converting to VSM Extended Format, after you run the VTCS `CONFIG` utility, run the HSC `MERGEcds` Utility to transfer volume information from the old CDS to the new CDS.	
"Updating the Tape Management System" on page 120	You might not have to do anything here, unless you're adding VTV ranges or MVCs...but read through this section to make sure everything's set up correctly with your TMS(s)...	
"Updating HSM" on page 121	...if you're an HSM user, and are routing HSM jobs to VSM...	

Table 6. VTCS Installation Summary and Checklist

Task	Notes	✓ to Verify Completion
“Routing Data Sets to VSM” on page 121	You might not have to do anything here, but read through this in case you have some new or changed jobs coming to VSM. There are some changes in the way things work (hint: it’s simpler), which you can probably put to good use...	
“Starting VTCS” on page 124	Just a reminder that if you modified the HSC procedure as recommended, HSC initialization automatically starts VTCS, and HSC termination automatically terminates VTCS.	

Verifying VSM Software and Hardware Prerequisites

VTCS System Software Requirements

Verify the software prerequisites for VTCS 6.1.0 listed in Table 7.

Table 7. ***VTCS 6.1.0 Minimum Software Requirements***

Software Description	Minimum Version/Release
Operating System	MVS 5.2.2 and above **Note:** NCS SMC JES3 requires JES3 5.1.1 or higher All versions of OS390
Nearline Control Solution	NCS 6.1 **Note:** • VTCS 6.1.0 requires HSC 6.1.0 and will not run with previous versions of HSC. • You can use the HSC 5.0 and above MERGEcds utility to convert the CDS to one of the VSM Extended Formats. For more information, see "Converting the Formatted CDS to VSM Extended Format" on page 83. • If you are using RMM in an MVS/CSC environment, MVS/CSC can share the tape management catalog with the host(s) running HSC if you have the following installed: • RMM APAR OA03368 • VSM3 microcode N01.00.65 or later or VSM4 microcode D01.00.03 or later. Otherwise, the tape management catalog cannot be shared or VTV scratch mounts will fail.
Expert Performance Reporter (optional software)	ExPR 4.0
Expert Library Manager (optional software)	To use Expert Library Manager (ExLM) with VSM for VTV consolidation using ExLM, ExLM 4.0, HSC 4.0.0, and VTCS 4.0.0. For more information about using ExLM with VSM, see "Using ExLM to Manage VSM Resources" in Chapter 2 of the ExLM *System Administrator's Guide*.

Caution: In a VSM configuration with multiple hosts that share the same HSC CDS, StorageTek strongly recommends that you:

- Install VTCS on all MVS hosts by completing all tasks described in “Installing HSC and VTCS” on page 69. Installing VTCS on all MVS hosts ensures that these hosts cannot scratch an MVC.

  Note that `CONFIG` lets you define MVS hosts that are not connected to a VTSS.

- Do *not* use a host that does not have VTCS installed to enter MVCs into an ACS, otherwise these MVCs will be eligible for selection as scratch volumes by any host in the configuration with HSC installed.

Third Party Tape Copy Software for Migrating Data to VSM

Table 8 lists Third Party tape copy software for migrating data to VSM.

Table 8. Third Party Tape Copy Software for Migrating Data to VSM

Product Name	Vendor
Beta55	Beta Systems Software AG
TelTape/390	Cartagena Software Limited
CA-1®/Copycat	Computer Associates International
CA-Dynam®/TLMS/ Copycat	Computer Associates International
MediaMerge	eMag Solutions
FATSCopy	Innovation Data Processing
Tape/Copy	OpenTech Systems, Inc.
Zela	Software Engineering of America
CARTS-TS TapeSaver	UNICOM Systems, Inc.

Nearline Hardware Requirements

Verify the minimum Nearline hardware requirements listed in Table 9.

Table 9. ***VSM Nearline Hardware Requirements***

Hardware	Requirement
LSMs	Any of the following but 9310, 9740, or SL8500 recommended by StorageTek: 4410, 9310, 9740, 9360, and SL8500
Transports and media	VSM RTDs can be a mixture of 9490 (Timberline), 9490EE (Timberline EE), T9840A, T9840B,T9840C, T9940A, T9940B, and T10000 transports. Each VTSS must have a minimum of two library-attached transports for each media type used for MVCs. For example, if your MVCs are STANDARD and ECART, you need a minimum of *either* two 9490s *or* two 9490EEs as RTDs. If your MVCs are STANDARD, ECART, ZCART, and STKIR, you need a minimum of two 9490EEs *and* two 9840s as RTDs. **Note:** Using T9940s as RTDs requires HSC 4.0.0 and VTCS 4.0.0 with PTF L1H1043 applied. Valid media types for the supported RTDs are: 9490: STANDARD, ECART 9490EE: STANDARD, ECART, ZCART T9840A, T9840B, T9840C: See “If you are adding T10000 encrypting transports and new media to encrypt, create VOLATTRs for the new media and STORCLAS statements to allow VTCS to select this media. For example:” on page 27. T9940A, T9940B: See “VTCS Considerations to Correctly Specify MVC Media” on page 26. T10000: T10000T1, T10000TS

Clustered VTSS Requirements

Table 10. describes the requirements for implementing Clustered VTSS configurations.

***Table 10.* Clustered VTSS Requirements**

Component	Requirement
Primary and Secondary **VTSSs within a cluster**	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: • Primary VSM4, Secondary VSM3 • Primary VSM4, Secondary VSM4 • Primary VSM3, Secondary VSM3 • Primary VSM3, Secondary VSM4 (not recommended)
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. See your StorageTek hardware service representative for details.
VTCS software	• VTCS 5.1.0 (for enhanced clustered support) • The Advanced Management Feature (to enable the `REPLICAT` parameter of the `MGMTclas` statement)

Determining VSM Configuration Values

The following sections tell how to determine configuration values for your VSM system. Use Table 25. on page 125 to record these values. This table also provides a record of your site's VSM configuration, which can help you and StorageTek service troubleshoot problems with your VSM system.

Note: Unless otherwise noted, in each of the following sections, the values you determine must match wherever you use them. For example, the unit addresses described in "RTD Unit Addresses and Identifiers" on page 24 must match on the following:

- The HSC `SLIDRIVS` macro.
- If you will share these transports with MVS, when you assign MVS device addresses to these transports via the `HCD` facility.

Note: For NCS 6.1, The `UNITATTR` statement has been moved from HSC to SMC is required **only** to set the real transport model type for non-library transports (which are not supported for VSM). For more information, see *SMC Configuration and Administration Guide*.

HSC and SMC Definition Data Set Names

Determine the names of the HSC data sets that will contain your VSM system's `VOLATTR`, `MVCPool`, `MGMTclas` and `STORclas` statements. `MGMTclas` and `STORclas` statements must reside in the same file (sequential data set or PDS member) for cross-validation.

Note: For NCS 6.1, the `TAPEREQ` statement (and the accompanying `TREQDEF` command) has been moved from HSC and MVS/CSC to SMC, and now resides in an SMC definition data set. For more information, *SMC Configuration and Administration Guide*.

VTSS Identifiers

Determine your system's 1 to 8 character VTSS identifiers, which you specify when you run VTCS `CONFIG` to initially install and configure your VSM system as described in "Configuring VTCS" on page 114.

Caution: Note the following:

- You specify the VTSS identifier *only* via the `NAME` parameter of the `VTSS` statement of `CONFIG`, which sets the VTSS identifier in both the VTSS microcode (as displayed in the Subsystem Name field in the LOP) and in the configuration area of the HSC CDS. After VSM is put into operation, the VTSS identifier is also stored in each VTV record in the CDS. Each VTV record contains the VTSS identifier on which that VTV is resident or, if the VTV is migrated, the VTV record contains the VTSS identifier from which the VTV was migrated.
- Once you set the VTSS identifier via the `NAME` parameter, you *cannot* change this identifier in the HSC CDS with `CONFIG`. `CONFIG` *will not* let you change the `NAME` parameter after an initial setting and changing the VTSS identifier using the Subsystem Name field of the LOP *cannot* change the VTSS identifier in the HSC CDS.
- You can, however, change a VTSS identifier when merging CDS.
- It is especially critical that you *do not* attempt to rename a VTSS that contains data on VTVs, which includes VTSS-resident VTVs and migrated VTVs!
- For an initial setting *only* (not a change), you can set the VTSS identifier in the `NAME` parameter only if the VTSS identifier value in the VTSS microcode is:
- The factory setting (all blanks).
- A value of `99999999` (eight 9s).
- Therefore, for an initial setting *only,* if the name in the VTSS microcode *is not* all blanks or `99999999`, your StorageTek hardware representative must use the VTSS LOP to set the VTSS identifier to `99999999` so you can set the VTSS identifier to the value you want via the `NAME` parameter.

Note: For NCS 6.1, the `UNITATTR` statement has been moved from HSC to SMC and is **no longer required** for VTDs.

VTD Unit Addresses

Determine MVS unit addresses for your system's VTDs as follows:

- For each VTSS in your VSM configuration, determine a unique unit address range for the VTDs in that VTSS. Do not use duplicate addresses or overlapping address ranges, either within the VTDs in a VTSS or across VTSSs.
- For each VTSS in your VSM configuration, you must define its VTD unit addresses to VTCS via `CONFIG`.

  In a multi-host, multi-VTSS configuration, you can configure your VTD unit addresses to restrict host access to VTSSs. Note that the VTVs created and MVCs initially written to from a VTSS are considered that VTSS's resources, so only hosts with access to a VTSS also have access to its VTVs and MVCs.
- For each HSC host, use the `HCD` facility to define to MVS the VTDs that host can access as described in "Assigning MVS Device Numbers to VTDs" on page 66. The unit addresses you specify via the `HCD` facility *must* match the unit address range you specified for that host via `CONFIG`.
- If you use MIM or GRS, add VTDs to the list of managed devices.

VSM Esoterics and Esoteric Substitution

Default SMC Device Allocation for VTDs in Multi-VTSS Systems

Where multiple VTSSs are eligible for allocation (from the `UNITNAME` in the JCL, SMS unit substitution, or from SMC `TAPEREQ` statements), the **default** SMC device allocation for VTDs is as follows:

- For *specific* volume requests, if the VTV is VTSS-resident, SMC will restrict allocation to the VTDs in that VTSS.

  If the VTV is *not* resident, SMC will restrict allocation to the VTDs in the VTSSs that have physical access to the MVC(s) except in JES3 with tape setup, where allocation will be restricted to a the VTDs of a single VTSS selected at random from the all the VTSSs that have physical access to the MVC(s).

- For *scratch* volume requests, if a Management Class is assigned to the VTV, then SMC allocates a VTD that can best satisfy the Management Class policies. For example, if the Management Class specifies a migration policy via the `MIGPOL` parameter (ACS ID, MVC media, and number of migration copies), then SMC allocates a VTD by selecting a VTSS attached to ACS(s) that best satisfy the policy. Similarly, if the Management Class specifies VTV replication via the `REPLICAT(YES)` parameter, SMC attempts to select a VTD on the Primary VTSS of a full-function Cluster.

If a Management Class is **not** assigned to the VTV, then SMC attempts to allocate the VTD in a VTSS with the greatest difference between the DBU and HAMT.

The following sections describe the requirements for defining and using VSM esoterics to influence VTD allocation for the following interfaces:

- The StorageTek DFSMS interface
- SMC `TAPEREQ` statements and User Exits for the following environments:
  - JES2 and JES3 without tape SETUP
  - JES3 with tape SETUP

Note:

- To ensure that virtual requests, like, actually go to virtual, StorageTek recommends that esoterics contain **only** VTDs.
- For any esoteric that you design, you must define the esoteric and associate it with the MVS device numbers for the VTDs you have chosen for that esoteric; see “Associating VTD MVS Device Numbers and Esoterics” on page 66.

- Assigning a Management Class to a VTV **requires** allocating a VTD in a VTSS that can satisfy the requirements of the assigned Management Class. For any jobs that route data to VSM via esoteric substitution **and** assign a Management Class to the data, **ensure** that you specify an esoteric that includes VTSSs that can satisfy the requirements of the Management Class!
- As described in "Routing Data Sets to VSM" on page 121, you can also route data sets to VSM by:
  - Using SMC `TAPEREQ` statements or HSC User Exits that do *not* use esoteric substitution
  - Changing existing JCL or writing new JCL

VSM Esoterics and Esoteric Substitution for the StorageTek DFSMS Interface

If you use these interfaces, you can use any valid esoteric defined to MVS. For these interfaces, begin your esoteric names with an alphabetic character to meet SMS requirements.

With these interfaces, for example, you can define and use any or all of the following:

- An esoteric that represents all the VTDs in your VSM system
- An esoteric for each VTSS that represents all the VTDs in that VTSS
- An esoteric that represents a subset of the VTDs in a single VTSS
- An esoteric that spans VTSSs

Note: The StorageTek DFSMS interface cannot assign Management Class to a VTV at VTD allocation.

VSM Esoterics and Esoteric Substitution for SMC TAPEREQ Statements and NCS User Exits

If you use the SMC TAPEREQ statement or NCS User Exits, see the following sections for information on how to define and use VSM esoterics for these environments:

- JES2 and JES3 without tape SETUP
- JES3 with tape SETUP

VSM Esoterics for JES2 and JES3 without Tape SETUP. In JES2 and JES3 without tape SETUP, for TAPEREQ statements or User Exits, esoteric definition and substitution is as follows:

- Esoteric definition is **optional** with these interfaces in these environments. That is, if you do not use esoteric substitution for VSM, you do not have to define any VSM esoterics.
- As long as the esoteric is a valid esoteric defined to MVS, esoteric substitution works as follows:
  - Allocation determines the common drives between the specified esoteric and the Eligible Device List (EDL).
  - As long as there are sufficient drives in the list of common drives, then allocation continues using this list of common drives.
- If you use esoteric substitution a multi-VTSS environment, StorageTek recommends that you**:**
  - Define an esoteric for each VTSS.
  - Ensure that each VTSS esoteric represents exactly the entire range of devices for only that VTSS.
  - Ensure that the VTSS esoteric name matches the VTSS identifier defined via the CONFIG utility.

  This approach allows you to do esoteric substitution at the VTSS level and also allows you to use the same VTSS identifier defined via the CONFIG utility.

  - You can also define other virtual esoterics. For example, you can define and substitute an esoteric that represents all of the VTDs in all VTSSs or an esoteric that represents all VTSSs that comprise VTSS Clusters. For more information on how VTCS allocates VTDs in a multi-VTSS system, see "Default SMC Device Allocation for VTDs in Multi-VTSS Systems" on page 17.
- For consistency, especially in installations that run both JES2 and JES3, you may want to define an esoteric structure for JES2 such as shown for JES3 with tape SETUP in Figure 1 on page 20.

VSM Esoterics for JES3 with Tape SETUP

Figure 1 is an example of a VSM esoteric structure for JES3 with tape SETUP.

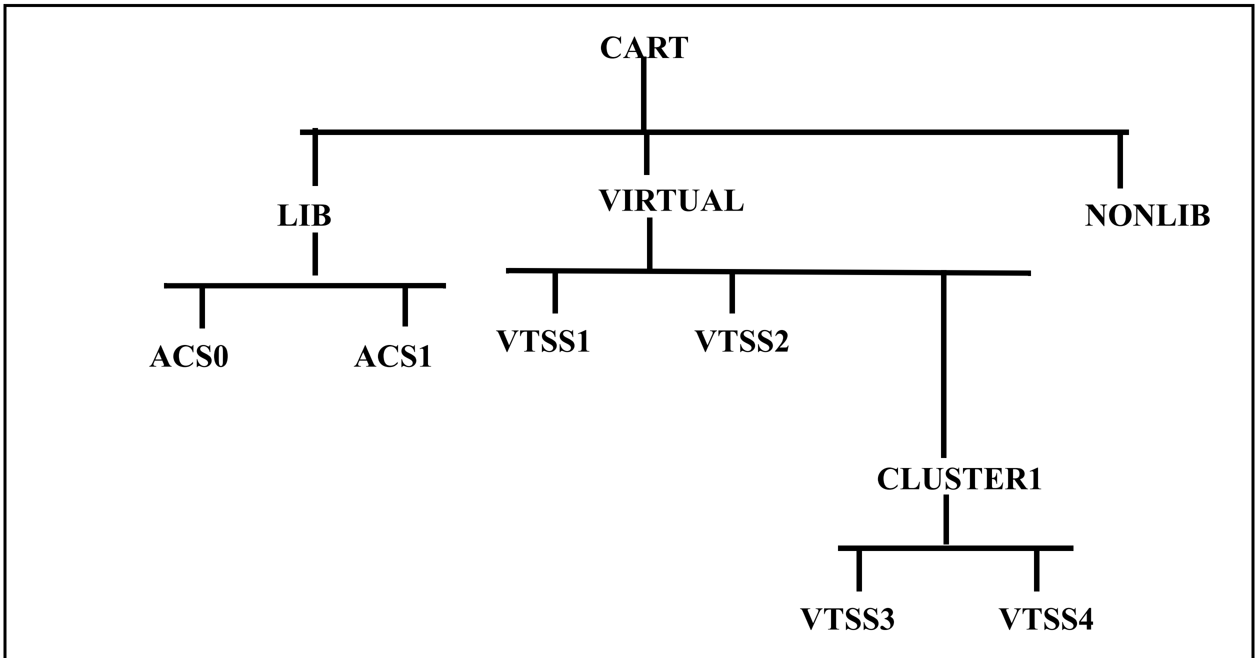


Figure 1. Example VSM Esoteric Structure for JES3 with Tape SETUP

Defining VSM Esoterics for JES3 with Tape SETUP . Figure 1 illustrates the VSM requirements and recommendations for defining esoterics in JES3 with tape SETUP as follows:

Requirement

Esoteric definition is **required** with these interfaces in this environment. That is, even if you do **not** use esoteric substitution for VSM, you must define VSM esoterics as described in the following sections.

Recommendation

StorageTek recommends that you begin the esoteric structure with a high-level esoteric that allocates all tape jobs. Your system typically has this esoteric previously defined, which is **CART** in Figure 1.

Requirement

For Nearline systems, under the high-level esoteric, you must have defined:

- A subesoteric for "library" (Nearline managed) real tape jobs (**LIB** in Figure 1). Under **LIB** are the subesoterics **ACSO** and **ACS1** that represent drives in two different ACSs. You specify **ACSO** and **ACS1** in the HSC LIBGEN on the `ACSDRV` parameter of the `SLIACS` macro.
- A second subesoteric for "non-library" (not Nearline managed) tape jobs (**NONLIB** in Figure 1). You specify this esoteric in the HSC LIBGEN on the `NNLBDRV` parameter of the `SLILIBRY` macro.

Requirement

Any esoteric that represents VTDs **must** contain only VTDs and must be defined in the JES3 initialization deck.

Requirement

For multi-VTSS environments, you **must** define an esoteric that represents all VTDs in your system (**VIRTUAL** in Figure 1. on page 20).

Requirement

StorageTek recommends that you place the esoteric that represents all VTDs in your system (**VIRTUAL** in Figure 1. on page 20) under the high-level esoteric. The benefit of placing the virtual subesoteric under the high-level esoteric is that this esoteric structure prevents JES3 HWS device allocation optimization from allocating a VTV to a non-virtual device.

Recommendation

For multi-VTSS environments, StorageTek recommends that you define an esoteric for each VTSS **(VTSS1**, **VTSS2**, **VTSS3**, and **VTSS4** in Figure 1. on page 20). Each esoteric *must* represent the entire range of devices for only that VTSS and the esoteric name *must* match the VTSS name on the VTCS `CONFIG` statement.

☞ **Note:** In JES3 with tape SETUP, you can also define other esoterics, such as an esoteric that represents all VTDs in all VTSSs or all VTDs in all VTSSs that comprise a VTSS Cluster (for example, **CLUSTER1** in Figure 1. on page 20). You can use these esoterics for esoteric substitution and JCL reference.

VTV Definitions

You define your system's VTV volsers to VTCS by volser ranges with the following:

- VTCS `CONFIG`.
- HSC `VOLATTR` statement.

Determine your system's VTV volser ranges as follows:

- For an initial `CONFIG` definition, consider defining only enough VTVs for reasonable growth. This method allows for growth without rerunning `CONFIG` but does not reserve unnecessary space in the CDS, which can impact VTV processing performance. Note that if the CDS does not contain sufficient space to run VTCS `CONFIG`, you will also have to run HSC `RECONFIG`. For more information about sizing the CDS for VSM, see "HSC CDS DASD Space" on page 43.

  If your VTV requirements expand beyond initial definition, then rerun `CONFIG` to define additional VTVs.
- You can only add new VTV ranges. A range can consist of a single volume. You cannot delete or modify existing ranges.
- If you are currently writing files to non–standard length tapes and will route these files to VTVs, you may need additional VTVs because VTVs emulate standard–length cartridges. This may require a change to the JCL volume count parameter.
- NCS/VTCS **does not allow** allocation of unlabeled tapes to VTVs. Unlabeled VTVs can cause the following for scratch VTV allocation requests:
  - If your JCL specifies a virtual esoteric, the NCS Storage Management Component (SMC) fails the allocation.
  - If you have a default esoteric such as CART and specify allocation to virtual (via SMC `TAPEREQ` or HSC User Exit), the allocation will go to a non-virtual device.
- You must define VTV volsers to your tape management system; for more information, see "Updating the Tape Management System" on page 120.
- Ensure that VTV volser ranges do not duplicate or overlap existing TMS ranges or volsers of real tape volumes, including Nearline volumes, *including* MVCs and Nearline volumes that are regularly entered and ejected from the ACS!
- If VTDs are being used across multiple MVS images and VTV volsers are unique, add a generic entry for SYSZVOLS to the SYSTEM inclusion RNL to insure that a VTV is used by only one job at a time. If you are using automatic tape switching, also add a generic entry for SYSZVOLS to the

SYSTEM inclusion RNL to prevent a system from holding a tape device while it waits for a mount of a volume that is being used by another system.

For more information, see the IBM publication *OS/390 MVS Planning: Global Resource Serialization.*

- If you specify scratch subpools for scratch mounts of VTVs (for example, with the SMC `TAPEREQ SUBPOOL` parameter or SMC User Exit 01), use the following guidelines:
  - If you need to define new subpools, add `SCRPOOL` statements to HSC PARMLIB for the VTV volsers.
  - HSC mixed-media support lets you mix VTV and real volume types in the same scratch pool. In this case, ensure that the mount request specifies a VTD as transport type (for example, via `TAPEREQ MEDIA (VIRTUAL)`. In addition, if you are routing data to a specific VTSS (for example, by using esoteric substitution as described in “VSM Esoterics and Esoteric Substitution” on page 17) and the request specifies a subpool, ensure that the subpool contains scratch VTVs.

Hint: Note the following:

- You can use `Query` to display the available scratch count of a subpool.
- You can dynamically reload `SCRPOOL` statements via the `SCRPDEF` command. For more information, see *HSC System Programmer’s Guide for MVS.*
- The `Warn SCRatch`, `Display SCRatch`, and `Display THReshld` commands are enhanced to let you manage and monitor scratch VTVs. For more information, see Chapter 2, “Commands, Control Statements, and Utilities,” in *HSC Operator’s Guide for MVS.*

- By default, VTCS assigns a Management Class to VTVs only on scratch mounts. You can, however, specify that VTCS assigns a Management Class whenever VTCS mounts a VTV (for read or write).

Caution: If you specify that VTCS assigns a Management Class whenever VTCS mounts a VTV, these attributes can change, which can cause undesirable or unpredictable results.

For example, if an application writes data set `PROD.DATA` to VTV100 with a Management Class of `PROD`, then writes data set `TEST.DATA` to VTV100 with a Management Class of `TEST`, then the VTV (and both data sets) has a Management Class of `TEST`. Similarly, it is possible to write SMC `TAPEREQ` statements or SMS routines that assign different Management Classes to the same data set (for example, based on jobname), which can also cause a VTV’s Management Class to change.

RTD Unit Addresses and Identifiers

RTDs, which are Nearline transports, require LIBGEN definitions.

Note: For NCS 6.1, The UNITATTR statement has been moved from HSC to SMC is required **only** to set the real transport model type for non-library transports (which are not supported for VSM). For more information, see *SMC Configuration and Administration Guide*.

If your system's RTDs are new transports, determine 4-digit hexadecimal MVS unit addresses for these transports. The addresses you choose must be the same for all hosts in the configuration. You will use these addresses to:

- Add a SLIDRIVS macro to define RTD device addresses during the HSC LIBGEN update as described in "Creating or Updating the HSC LIBGEN" on page 80.
- Run the HCD facility to assign MVS device numbers to these transports as described in "Assigning MVS Device Numbers to RTDs" on page 66.

Caution: Note the following:

- StorageTek **strongly recommends** that you define your RTDs to MVS (as normal 3490 tape drives), even if you do not intend to vary them online to MVS. This prevents the RTD addresses used in CONFIG and LIBGEN from accidentally being used for other devices. If you do not do this, and subsequently use the addresses for other MVS devices, you will cause problems with LOGREC processing, because VTCS will write records using the RTD addresses, and MVS will write records for other devices with those same addresses.
- StorageTek **requires** that RMM users define their RTDs to MVS. RMM causes problems if it sees the IEC501 mount message generated for RTDs by VTCS and if the device in the mount message is not defined to MVS.

Whether your system's RTDs are new or existing transports, you will use their MVS unit addresses to define the RTDs to VTCS on the CONFIG VTSS RTD DEVNO parameter.

You also specify the RTD identifier on the CONFIG VTSS RTD NAME parameter. To help identify the RTDs connected to each VTSS, StorageTek recommends that you choose RTD identifiers that reflect the VTSS identifier (specified on the VTSS NAME parameter) and the RTD's MVS device number (specified on the RTD DEVNO parameter).

In configurations where multiple VTSSs are connected to and dynamically share the same RTD, in each VTSS definition you can either assign unique RTD identifiers or use the same RTD identifier.

Note: You can specify that Nearline transports can only be used as RTDs. For more information, see "Creating or Updating the HSC LIBGEN" on page 80.

MVC Definitions

You define MVCs as described in the following sections:

- "VTCS Considerations to Correctly Specify MVC Media"
- "Define and Select Nearline Volumes" on page 28
- "Define Available MVCs with CONFIG" on page 28
- "Define the MVC Pool" on page 29
- "Protect MVCs and Nearline Volumes" on page 30

VTCS Considerations to Correctly Specify MVC Media

Table 11 describes the values required to specify the desired media and recording technique on the HSC VOLATTR statement and HSC STORCLAS statement to correctly specify the desired MVC media.

Table 11. RTD Model/MVC Media Values

TAPEREQ/ VOLATTR MEDIA	Transport MODEL	RECTECH	STORCLAS MEDIA	Density	Encrypted?
STK1R	9840	STK1RA	STK1RAB	single	N/A
	T9840B	STK1RB	STK1RAB	single	N/A
	T9840C	STK1RC	STK1RC	double	N/A
STK2P	T9940A	STK2PA	STK2PA	single	N/A
	T9940B	STK2PB	STK2PB	double	N/A
T10000T1	T1A	T1AN	T10000T1	single	no
	T1AE34	T1AE	T10000E1	single	yes
T10000TS	T1A	T1AN	T10000TS	single	no
	T1AE34	T1AE	T10000ES	single	yes

Note: Use Table 11 as a guideline to:

- Create VOLATTR statements that segregate single/double density media or encrypted/non-encrypted media.
- Specify the correct STORCLAS MEDIA values to assign the desired cartridge type and recording technique to MVCs.
- Determine which transport models can write to/read from which media. A higher capability transport (double density vs. single, or encryption vs. non-encryption) can read from media written by a lower capability transport, but can only write to that media from the beginning of the tape. A lower capability transport, however, cannot read from media written by a higher capability transport but can write to that media from the beginning of the tape.

Examples.

- If you are adding T10000 encrypting transports and new media to encrypt, create VOLATTRs for the new media and STORCLAS statements to allow VTCS to select this media. For example:

```
VOLATTR SERIAL(MVC900-MVC999) MEDIA(T10000T1) RECTECH(T1AE)
STORCLAS NAME(T1ENCRYPT) MEDIA(T10000E1)
```

- If you are adding T10000 encrypting transports and want to convert existing media to encryption media, change existing VOLATTRs to specify encryption and change existing STORCLAS statements to request encryption. For example:

```
VOLATTR SERIAL(MVC800-MVC899) MEDIA(T10000T1) RECTECH(T1AE)
STORCLAS NAME(T1) MEDIA(T10000E1)
```

  Here's how it works: If I have MVCs that already contain data, I cannot add "encrypted" VTVs to these MVCs. I can, however, encrypt data on initialized MVCs that do not contain data. To make this strategy work, therefore, ensure that you have sufficient free T10000 MVCs and also consider doing demand drains on MVCs that do contain data to free them up.

Define and Select Nearline Volumes

First, to define and select Nearline volumes for MVCs, use these guidelines:

- MVCs require `VOLATTR` statements to ensure that VTCS will select the correct RTD device type for each MVC. Select volumes for MVCs that are compatible with your system's RTD transport types.
- For mixed–media VSM systems, select volumes that include at least one media type compatible with each of your system's RTD transport types. See Table 9. on page 12 for information about the RTD transport types and media that VSM supports.

  Note that VSM selects media for migration processing and reclaim processing according to the media types of volumes in your system's MVC pool.
- If you define new Nearline volumes as MVCs, you must create MVS volsers for these volumes and initialize STANDARD, ECART, and ZCART volumes as 36-track format standard label volumes.
- As described in "Protect MVCs and Nearline Volumes" on page 30, if possible, create a new and separate volser range for MVCs. Ensure that if you define new volumes, you do not overlap existing TMS ranges.

Define Available MVCs with CONFIG

Second, use VTCS `CONFIG` to define all MVCs *available* to VTCS. `CONFIG` reserves space for these volumes in the HSC CDS. The `MVCPool` statements define the *MVC pool*, which contains the MVCs that VTCS actually *uses*.

For an initial `CONFIG` definition, consider defining only enough MVCs for reasonable growth of your MVC pool. This method allows you to expand your MVC pool without rerunning `CONFIG` (you only have to change your `MVCPool` statements) but does not reserve unnecessary space in the CDS, which can impact MVC processing performance. Note that if the CDS does not contain sufficient space to run VTCS `CONFIG`, you will also have to run HSC `RECONFIG`. For more information about sizing the CDS for VSM, see "HSC CDS DASD Space" on page 43.

For example, if you currently need 300 MVCs but will need to add 150 more MVCs within the next 6 months, define an MVC range of 450 volsers with `CONFIG`, but only apply `MVCPool` statements to the first 300 "in use" MVCs. As your MVC space requirements increase, update and reapply your `MVCPool` statements to add the second 150 MVCs.

If your MVC space requirements expand beyond the second 150 MVCs, then rerun `CONFIG` to define additional MVC ranges and update and reapply your `MVCPool` statements.

Note:

- You can only add new MVC ranges. A range can consist of a single volume. You cannot delete or modify existing ranges.
- A VSM audit of all MVCs will audit all MVCs defined with `CONFIG` including those that are *not* specified in the `MVCPool` statements.

Define the MVC Pool

Third, create MVCPool statements, which specify the pool of MVCs available for migration and consolidation requests, using the following guidelines:

- Because MVCPool statements specify the "in use" MVCs, MVCPool statements can (and typically do) define a subset of the available MVCs you defined via CONFIG. MVCPool statements, however, can only specify MVCs you already defined with CONFIG. For more information about defining an initial MVC pool, see "Creating an MVC Pool" on page 95.
- StorageTek recommends that you use identical MVCPool statements on all hosts. A host can automigrate any VTV on any VTSS to which the host is connected, including VTVs created by another host. If your VSM configuration consists of hosts cross-connected to multiple VTSSs, therefore, separate MVC pools do not guarantee that a host automigrates only VTVs it creates to only its MVC pool. To most effectively segregate VTVs on groups of MVCs, see "Using Storage and Management Classes to Segregate Individual Workloads on Separate Sets of MVCs" on page 99.
- Ensure that your MVC pool consists of volumes that physically reside in ACS that contains your system's RTDs.

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

- To redefine your MVC pool, change your MVCPool statements and reload them via the VT MVCDEF command.

Protect MVCs and Nearline Volumes

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

- If possible, create a new and separate volser range for MVCs to prevent HSC from writing to MVCs and to prevent VSM from writing to conventional Nearline volumes.
- VTCS, not MVS, controls access to MVCs. The tape management system does not control VSM access to an MVC volume and does not record its usage. If you choose to define MVCs to the tape management system, to ensure that the tape management system does not accidentally access MVCs, follow the guidelines in "Updating the Tape Management System" on page 120.
- Use your security system to restrict access to MVCs as described in "Defining MVC Pool Volser Authority" on page 112.
- HSC automatically marks newly entered MVC volumes as non-scratch. If you define existing Nearline volumes as MVCs, ensure that these volumes do not contain data you need, then run the HSC `UNSCratch` Utility to unscratch them. For more information, see *HSC System Programmer's Guide for MVS*.

MVS/CSC and Non-MVS/CSC Client Connection Values

"Virtual ACS IDs" describes the connection values you determine for MVS/CSC and non-MVS/CSC clients.

Virtual ACS IDs

To connect to VSM, clients use a decimal virtual ACS ID that maps to a VTSS identifier that you determined in "VTSS Identifiers" on page 15. You use the LibraryStation `VIRTACS` statement to do this mapping. To avoid conflicts with real ACS IDs, StorageTek recommends that you select virtual ACS IDs by starting with the highest possible value (126) and working backwards.

For example, to define virtual ACS 126 and map it to VTSS VTSS02, you would create the following `VIRTACS` statement:

```
VIRTACS ID(126) VTSSNAME(VTSS02)
```

Virtual ACS Location for VSM2s and VSM3s. To LibraryStation, a VSM2/3 virtual ACS consists of four virtual LSMs (0-3), each containing four drive panels (1-4) with four VTDs each (0-3) for a total of 64 VTDs. LibraryStation, therefore, references VTDs by their ACS location in *ascid,lsmid,panelnum,devicenum* format. For example, LibraryStation references the 27th VTD in virtual ACS 126, above, by ACS location 126,1,3,2.

Table 12 cross-references VTD numbers to their LibraryStation virtual ACS locations. See "Connecting MVS/CSC Clients to VSM" on page 106 for procedures for defining these library locations to MVS/CSC clients and mapping them to MVS device numbers.

Table 12. LibraryStation VSM2/3 Virtual ACS Locations for VTDs

VTD	Virtual ACS Location
1	*acsid*,0,1,0
2	*acsid*,0,1,1
3	*acsid*,0,1,2
4	*acsid*,0,1,3
5	*acsid*,0,2,0
6	*acsid*,0,2,1
7	*acsid*,0,2,2
8	*acsid*,0,2,3
9	*acsid*,0,3,0
10	*acsid*,0,3,1
11	*acsid*,0,3,2
12	*acsid*,0,3,3
13	*acsid*,0,4,0
14	*acsid*,0,4,1

15	*acsid*,0,4,2
16	*acsid*,0,4,3
17	*acsid*,1,1,0
18	*acsid*,1,1,1
19	*acsid*,1,1,2
20	*acsid*,1,1,3
21	*acsid*,1,2,0
22	*acsid*,1,2,1
23	*acsid*,1,2,2
24	*acsid*,1,2,3
25	*acsid*,1,3,0
26	*acsid*,1,3,1
27	*acsid*,1,3,2
28	*acsid*,1,3,3
29	*acsid*,1,4,0
30	*acsid*,1,4,1
31	*acsid*,1,4,2
32	*acsid*,1,4,3
33	*acsid*,2,1,0
34	*acsid*,2,1,1
35	*acsid*,2,1,2
36	*acsid*,2,1,3
37	*acsid*,2,2,0
38	*acsid*,2,2,1
39	*acsid*,2,2,2
40	*acsid*,2,2,3
41	*acsid*,2,3,0
42	*acsid*,2,3,1
43	*acsid*,2,3,2
44	*acsid*,2,3,3
45	*acsid*,2,4,0
46	*acsid*,2,4,1

47	*acsid*,2,4,2
48	*acsid*,2,4,3
49	*acsid*,3,1,0
50	*acsid*,3,1,1
51	*acsid*,3,1,2
52	*acsid*,3,1,3
53	*acsid*,3,2,0
54	*acsid*,3,2,1
55	*acsid*,3,2,2
56	*acsid*,3,2,3
57	*acsid*,3,3,0
58	*acsid*,3,3,1
59	*acsid*,3,3,2
60	*acsid*,3,3,3
61	*acsid*,3,4,0
62	*acsid*,3,4,1
63	*acsid*,3,4,2
64	*acsid*,3,4,3

Virtual ACS Location for VSM4s. To LibraryStation, a VSM4 virtual ACS consists of sixteen virtual LSMs (0-15), each containing four drive panels (1-4) with four VTDs each (0-3) for a total of 256 VTDs. LibraryStation, therefore, references VTDs by their ACS location in *ascid,lsmid,panelnum,devicenum* format. For example, LibraryStation references the 27th VTD in virtual ACS 126, above, by ACS location 126,1,3,2.

Table 13 cross-references VTD numbers to their LibraryStation virtual ACS locations. See "Connecting MVS/CSC Clients to VSM" on page 106 for procedures for defining these library locations to MVS/CSC clients and mapping them to MVS device numbers.

Table 13. ***LibraryStation VSM4 Virtual ACS Locations for VTDs***

VTD	Virtual ACS Location
1	*acsid*,0,1,0
2	*acsid*,0,1,1
3	*acsid*,0,1,2
4	*acsid*,0,1,3
5	*acsid*,0,2,0
6	*acsid*,0,2,1
7	*acsid*,0,2,2
8	*acsid*,0,2,3
9	*acsid*,0,3,0
10	*acsid*,0,3,1
11	*acsid*,0,3,2
12	*acsid*,0,3,3
13	*acsid*,0,4,0
14	*acsid*,0,4,1
15	*acsid*,0,4,2
16	*acsid*,0,4,3
17	*acsid*,1,1,0
18	*acsid*,1,1,1
19	*acsid*,1,1,2
20	*acsid*,1,1,3
21	*acsid*,1,2,0
22	*acsid*,1,2,1

23	*acsid*,1,2,2
24	*acsid*,1,2,3
25	*acsid*,1,3,0
26	*acsid*,1,3,1
27	*acsid*,1,3,2
28	*acsid*,1,3,3
29	*acsid*,1,4,0
30	*acsid*,1,4,1
31	*acsid*,1,4,2
32	*acsid*,1,4,3
33	*acsid*,2,1,0
34	*acsid*,2,1,1
35	*acsid*,2,1,2
36	*acsid*,2,1,3
37	*acsid*,2,2,0
38	*acsid*,2,2,1
39	*acsid*,2,2,2
40	*acsid*,2,2,3
41	*acsid*,2,3,0
42	*acsid*,2,3,1
43	*acsid*,2,3,2
44	*acsid*,2,3,3
45	*acsid*,2,4,0
46	*acsid*,2,4,1
47	*acsid*,2,4,2
48	*acsid*,2,4,3
49	*acsid*,3,1,0
50	*acsid*,3,1,1
51	*acsid*,3,1,2
52	*acsid*,3,1,3
53	*acsid*,3,2,0
54	*acsid*,3,2,1

55	*acsid*,3,2,2
56	*acsid*,3,2,3
57	*acsid*,3,3,0
58	*acsid*,3,3,1
59	*acsid*,3,3,2
60	*acsid*,3,3,3
61	*acsid*,3,4,0
62	*acsid*,3,4,1
63	*acsid*,3,4,2
64	*acsid*,3,4,3
65	*acsid*,4,1,0
66	*acsid*,4,1,1
67	*acsid*,4,1,2
68	*acsid*,4,1,3
69	*acsid*,4,2,0
70	*acsid*,4,2,1
71	*acsid*,4,2,2
72	*acsid*,4,2,3
73	*acsid*,4,3,0
74	*acsid*,4,3,1
75	*acsid*,4,3,2
76	*acsid*,4,3,3
77	*acsid*,4,4,0
78	*acsid*,4,4,1
79	*acsid*,4,4,2
80	*acsid*,4,4,3
81	*acsid*,5,1,0
82	*acsid*,5,1,1
83	*acsid*,5,1,2
84	*acsid*,5,1,3
85	*acsid*,5,2,0
86	*acsid*,5,2,1

87	*acsid*,5,2,2
88	*acsid*,5,2,3
89	*acsid*,5,3,0
90	*acsid*,5,3,1
91	*acsid*,5,3,2
92	*acsid*,5,3,3
93	*acsid*,5,4,0
94	*acsid*,5,4,1
95	*acsid*,5,4,2
96	*acsid*,5,4,3
97	*acsid*,6,1,0
98	*acsid*,6,1,1
99	*acsid*,6,1,2
100	*acsid*,6,1,3
101	*acsid*,6,2,0
102	*acsid*,6,2,1
103	*acsid*,6,2,2
104	*acsid*,6,2,3
105	*acsid*,6,3,0
106	*acsid*,6,3,1
107	*acsid*,6,3,2
108	*acsid*,6,3,3
109	*acsid*,6,4,0
110	*acsid*,6,4,1
111	*acsid*,6,4,2
112	*acsid*,6,4,3
113	*acsid*,7,1,0
114	*acsid*,7,1,1
115	*acsid*,7,1,2
116	*acsid*,7,1,3
117	*acsid*,7,2,0
118	*acsid*,7,2,1

119	*acsid*,7,2,2
120	*acsid*,7,2,3
121	*acsid*,7,3,0
122	*acsid*,7,3,1
123	*acsid*,7,3,2
124	*acsid*,7,3,3
125	*acsid*,7,4,0
126	*acsid*,7,4,1
127	*acsid*,7,4,2
128	*acsid*,7,4,3
129	*acsid*,8,1,0
130	*acsid*,8,1,1
131	*acsid*,8,1,2
132	*acsid*,8,1,3
133	*acsid*,8,2,0
134	*acsid*,8,2,1
135	*acsid*,8,2,2
136	*acsid*,8,2,3
137	*acsid*,8,3,0
138	*acsid*,8,3,1
139	*acsid*,8,3,2
140	*acsid*,8,3,3
141	*acsid*,8,4,0
142	*acsid*,8,4,1
143	*acsid*,8,4,2
144	*acsid*,8,4,3
145	*acsid*,9,1,0
146	*acsid*,9,1,1
147	*acsid*,9,1,2
148	*acsid*,9,1,3
149	*acsid*,9,2,0
150	*acsid*,9,2,1

151	*acsid*,9,2,2
152	*acsid*,9,2,3
153	*acsid*,9,3,0
154	*acsid*,9,3,1
155	*acsid*,9,3,2
156	*acsid*,9,3,3
157	*acsid*,9,4,0
158	*acsid*,9,4,1
159	*acsid*,9,4,2
160	*acsid*,9,4,3
161	*acsid*,10,1,0
162	*acsid*,10,1,1
163	*acsid*,10,1,2
164	*acsid*,10,1,3
165	*acsid*,10,2,0
166	*acsid*,10,2,1
167	*acsid*,10,2,2
168	*acsid*,10,2,3
169	*acsid*,10,3,0
170	*acsid*,10,3,1
171	*acsid*,10,3,2
172	*acsid*,10,3,3
173	*acsid*,10,4,0
174	*acsid*,10,4,1
175	*acsid*,10,4,2
176	*acsid*,10,4,3
177	*acsid*,11,1,0
178	*acsid*,11,1,1
179	*acsid*,11,1,2
180	*acsid*,11,1,3
181	*acsid*,11,2,0
182	*acsid*,11,2,1

183	*acsid*,11,2,2
184	*acsid*,11,2,3
185	*acsid*,11,3,0
186	*acsid*,11,3,1
187	*acsid*,11,3,2
188	*acsid*,11,3,3
189	*acsid*,11,4,0
190	*acsid*,11,4,1
191	*acsid*,11,4,2
192	*acsid*,11,4,3
193	*acsid,12*,1,0
194	*acsid*,12,1,1
195	*acsid*,12,1,2
196	*acsid*,12,1,3
197	*acsid*,12,2,0
198	*acsid*,12,2,1
199	*acsid*,12,2,2
200	*acsid*,12,2,3
201	*acsid*,12,3,0
202	*acsid*,12,3,1
203	*acsid*,12,3,2
204	*acsid*,12,3,3
205	*acsid*,12,4,0
206	*acsid*,12,4,1
207	*acsid*,12,4,2
208	*acsid*,12,4,3
209	*acsid*,13,1,0
210	*acsid*,13,1,1
211	*acsid*,13,1,2
212	*acsid*,13,1,3
213	*acsid*,13,2,0
214	*acsid*,13,2,1

215	*acsid*,13,2,2
216	*acsid*,13,2,3
217	*acsid*,13,3,0
218	*acsid*,13,3,1
219	*acsid*,13,3,2
220	*acsid*,13,3,3
221	*acsid*,13,4,0
222	*acsid*,13,4,1
223	*acsid*,13,4,2
224	*acsid*,13,4,3
225	*acsid*,14,1,0
226	*acsid*,14,1,1
227	*acsid*,14,1,2
228	*acsid*,14,1,3
229	*acsid*,14,2,0
230	*acsid*,14,2,1
231	*acsid*,14,2,2
232	*acsid*,14,2,3
233	*acsid*,14,3,0
234	*acsid*,14,3,1
235	*acsid*,14,3,2
236	*acsid*,14,3,3
237	*acsid*,14,4,0
238	*acsid*,14,4,1
239	*acsid*,14,4,2
240	*acsid*,14,4,3
241	*acsid*,15,1,0
242	*acsid*,15,1,1
243	*acsid*,15,1,2
244	*acsid*,15,1,3
245	*acsid*,15,2,0
246	*acsid*,15,2,1

247	*acsid*,15,2,2
248	*acsid*,15,2,3
249	*acsid*,15,3,0
250	*acsid*,15,3,1
251	*acsid*,15,3,2
252	*acsid*,15,3,3
253	*acsid*,15,4,0
254	*acsid*,15,4,1
255	*acsid*,15,4,2
256	*acsid*,15,4,3

HSC CDS DASD Space

Before installing VTCS, you must calculate the DASD space required for the HSC control data set (CDS). The DASD space for the CDS must be increased to accommodate your VSM system's resource definitions. The additional number of 4k blocks required in the CDS for VTCS can be expressed as:

- **For B format CDSs**:

  (number of VTVs /58) + (number of MVCs/71) + 17(number of VTSS) + number of configured MVC ranges + number of configured VTV ranges + 13

- **For C, D, and E format CDSs**:

  (number of VTVs /23) + (number of MVCs/37) + 17(number of VTSS) + number of configured MVC ranges + number of configured VTV ranges + 13

- **For F format CDSs**:

  (# VTV ranges) + (# VTV ranges)/862 + (# VTVs defined)/23 + (# VTVs defined)/19826 + (# MVC ranges) + (# MVCs defined)/37 + 18*(# of VTSSs) + 14

Tape Management System DASD Space

To accommodate your VSM system's VTVs, you may need to increase the DASD space for your tape management system. After you determine the number and range of VTVs your VSM system requires, see your tape management system documentation for specific information on calculating the DASD space requirements.

VSM Candidate Data Sets

Your StorageTek representative will run the VSM pre-sales planning tool to identify VSM candidate data sets. You choose a method to route these data sets to VSM as described in "Routing Data Sets to VSM" on page 121.

HSC COMMPath METHod Value

To optimize performance, StorageTek recommends that you set the HSC `COMMPath METHod` parameter to either `LMU` or `VTAM`, *not* to `CDS` to allow even sharing of resources in a multi-host configuration as shown in the example in Figure 22 on page 104.

Data Chaining a VTD Read Forward or Write Command

Note that when data chaining a Read Forward or Write command, the VTSS requires the minimum data chained update count.

VSM Policies

The following sections describe VSM operating policies:

- "The Advanced Management Feature"
- "VTSS Management Policies" on page 47
- "VTV Migration and Consolidation Policies" on page 52
- "VTV Recall Policy" on page 58
- "VTV Maximum Size Policy" on page 58
- "MVC Policies" on page 59

Caution: Many VSM policies are determined by `MGMTclas`, `STORclas`, and `TAPEREQ` statements, which should be the same on all hosts for the following reasons:

- Common `TAPEREQ` statements ensure that a VTV is handled identically regardless of which host accesses the VTV. For example, if a `TAPEREQ` statement assigns a Management Class to a VTV when it is created, you want the same Management Class assigned regardless of where (host-wise) the VTV is created.
- Similarly, common `MGMTclas` statements ensure that a VTV is managed identically regardless of which host accesses the VTV. For example, if a `MGMTclas` specifies duplexing, you want the same Management Class to apply to the VTV assigned regardless of where (host-wise) the VTV is accessed so the VTV is always duplexed.
- `STORclas` statements specify the ACS and MVC media for VTV migration. As above, common `STORclas` statements ensure that VTV migrate, recall, and reclaim operations access the desired MVCs.

The Advanced Management Feature

The Advanced Management Feature is an optional VTCS feature, and you enabled it as described in "Updating the HSC PARMLIB Member (SLSSYSxx)" on page 104. Table 14 describes the VTCS features enabled by the Advanced Management Feature.

Table 14. Features Enabled by the Advanced Management Feature

For this feature...	...use this interface...	...which is described in...
Management and Storage Classes	• `STORclas` statement • `MGMTclas` statement `MIGpol`, `RESTIME`, `CONSRC`, `CONTGT`, and `REPLICAT` parameters	• *VTCS Command and Utility Reference*, Chapter 3, "HSC Enhancements and Additions for VSM" • "Creating and Using VSM Management and Storage Classes" on page 96
	• `EXPORT` command/utility • `IMPORT` command/utility	• *VTCS Command and Utility Reference*, Chapter 1, "VTCS Utilities and Commands" • *VTCS Administrator's Guide*, Chapter 2, "Managing VSM"
Clustered VTSS configurations	• `MGMTclas` statement `REPLICAT` parameter • `CONFIG CLUSTER`, `CONFIG CLINK` statements • `MGMTclas` and `STORclas` statements (typical but not required)	• *VTCS Command and Utility Reference*, Chapter 1, "VTCS Utilities and Commands"
	• Implementation procedures	• *VTCS Administrator's Guide*, Chapter 2, "Managing VSM"
VTV Residency Interval before Automatic Migration Candidacy	`MGMTclas` statement `REPLICAT` parameter	• *VTCS Command and Utility Reference*, Chapter 3, "HSC Enhancements and Additions for VSM" • "VTV Residency Interval before Automatic Migration Candidacy" on page 53

Table 14. Features Enabled by the Advanced Management Feature

For this feature...	...use this interface...	...which is described in...
ACS and Media Type of MVCs for Migration and Reclamation	`MGMTclas` statement `MIGpol` parameter	• *VTCS Command and Utility Reference*, Chapter 3, "HSC Enhancements and Additions for VSM" • "ACS and Media Type of MVCs for Migration and Reclamation." on page 55
Source MVC ACS and Media for Consolidation of Migrated Duplexed VTVs	`MGMTclas` statement `CONSRC` and `CONTGT` parameters	• *VTCS Command and Utility Reference*, Chapter 3, "HSC Enhancements and Additions for VSM" • "Source MVC ACS and Media for Consolidation of Migrated Duplexed VTVs" on page 57

VTSS Management Policies

The following sections describe these VTSS management policies:

- "Maximum and Minimum Concurrent Automatic Migration, Immediate Migration, and Migrate-to-Threshold Tasks"
- "AMT Settings" on page 48
- "Deleting Scratched VTVs" on page 50
- "VTSS Preferencing" on page 51

Maximum and Minimum Concurrent Automatic Migration, Immediate Migration, and Migrate-to-Threshold Tasks. These configuration parameters specify the maximum and minimum number of concurrent automatic migration, immediate migration, and migrate-to-threshold tasks for each VTSS. Use these parameters to balance migration tasks with other tasks (such as recall and reclaim) for the RTDs you have defined for each VTSS. To set these parameters:

- Use the `CONFIG VTSS MAXMIG` parameter to specify the maximum migration tasks. You can also use `SET MIGOPT` to change this value.
- Use the `CONFIG VTSS MINMIG` parameter to specify the minimum migration tasks. You can also use `SET MIGOPT` to change this value.

Note: In some situations, VTCS may not be able to activate *all* the migration tasks specified by the `MAXMIG` parameter. For example:

- The VSM-wide RTD configuration consists of four 9840 and four 9490 transports.
- No Storage Classes with `STK1R` as the primary media have been defined.
- There is sufficient MVC media for the 9490 transports.

In this configuration, because only the 9490 media is used, only a maximum of four migration tasks are activated using the 9490 RTDs.

Similarly, there are circumstances when VTCS will start *less* than the number of migration tasks specified by the MINMIG parameter. For example:

- The configuration consists of a single VTSS with 4 RTDs in ACS 0 and 4 RTDs in ACS 1. All RTD device types are identical.
- `MINMIG` and `MAXMIG` are both set to 8.
- Two Storage Classes are defined, which point, respectively, to ACS 0 and ACS 1.

In this configuration, if there are migrates queued for both Storage Classes, then VTCS will start 8 requests. If however, there are only migrations queued for one Storage Class, then VTCS *will not* start 8 requests because the workload can only be serviced by one Storage Class and this class can only run on 4 RTDs.

Finally, also note that when you reset the `MINMIG` and/or `MAXMIG` values, the actual number of migration tasks may not be immediately affected because of the way that VTCS manages migration tasks.

You can use `Query` to display migration status.

AMT Settings. This policy controls the automatic space management/migration cycle. This cycle begins when DBU (Disk Buffer Utilization) exceeds the high AMT (HAMT) or the number of VTVs exceeds 97,000 (for VSM2s and VSM3s) or 291,000 (for VSM4s) and continues until DBU drops below the low AMT (LAMT).

To set the AMTs, use either of the following:

- The `CONFIG VTSS LOW` and `HIGH` parameters.
- `SET MIGOPT`.

Note:

- With `CONFIG`, AMT settings take effect when you start HSC and apply to the specified VTSS.

- With SET MIGOPT:
  - AMT settings take effect immediately and apply to the specified VTSS or if no VTSS is specified, to all VTSSs. If you try to set global values (no VTSS specified) and the values are not valid for one VTSS (for example, MAXMIG(5) and one VTSS only has 4 RTDs connected), VTCS will not set values for any VTSSs
  - You can set the LAMT, the HAMT, or both.

The default high and low AMTs are 80% and 70% respectively. Valid values are 5 to 95 for the LAMT and 6 to 95 for the HAMT. The LAMT must be at least one less than the HAMT. For example, for a HAMT of 90%, you cannot specify a LAMT greater than 89%.

The following are general guidelines for changing the defaults:

- The *difference* between the high and low AMTs affects the duration of the space management/migration cycle.
- *Lowering* the HAMT tends to trigger *more frequent* space management/migration cycles.
- *Raising* the HAMT tends to trigger *less frequent* space management/migration cycles.
- *Lowering* the LAMT tends to free more VTSS space *and* migrate *more* VTVs.
- *Raising* the LAMT tends to keep more VTVs resident in VTSS space *and* migrate *fewer* VTVs.

Hint: You can use Query to display the DBU, HAMT, and LAMT for each VTSS in your system. You can also use Query to display migration status.

Deleting Scratched VTVs. Use the `DELSCR` parameter of the `MGMTclas` statement to specify whether VSM deletes scratched VTVs.

Specifying `DELSCR YES` causes VSM to delete scratched VTVs, which frees VTSS buffer space.

Warning: 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 "Using ExLM with VTCS (All Versions)" in Chapter 2, "Using ExLM to Manage Nearline and VTCS Resources" of *ExLM System Administrator's Guide.*

VTSS Preferencing. Use the `VTSSSEL` statement to define a VTSS usage rule that applies to the VTSS list and its preferencing specified on a referenced `VTSSLST` control statement.

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`

You can use VTSS preferencing for the following functions:

- VTV scratch and specific mounts
- Demand recall
- Reclaim
- Drain
- Audit
- Export
- Consolidation

This VTSS preferencing also influence RTD selection for the following functions:

- Demand recall
- Reclaim
- Drain
- Audit
- Export
- Consolidation

VTV Migration and Consolidation Policies

The following sections describe these VTV migration policies:

- "Hosts Disabled from Migration, Consolidation and Export by VTV or Management Class" on page 52
- "VTV Residency Interval before Automatic Migration Candidacy" on page 53
- "Immediately Migrate VTVs On Dismount" on page 54
- "MVC Retain Interval" on page 55
- "Maximum VTVs per MVC" on page 55
- "ACS and Media Type of MVCs for Migration and Reclamation." on page 55
- "Number of Migration Copies" on page 56
- "Migrate Duplexed VTVs to Separate ACSs" on page 56
- "Output MVC ACS and Media for VTV Consolidation" on page 56
- "Source MVC ACS and Media for Consolidation of Migrated Duplexed VTVs" on page 57

☞ **Note:** You can use the `VTVMAINT` utility to change a VTV's Management and Storage Class.

Hosts Disabled from Migration, Consolidation and Export by VTV or Management Class. You can specify that a host cannot initiate automatic and demand migration and consolidation processing. Use the `CONFIG HOST NOMIGRAT` parameter to set this policy. Note that `NOMIGRAT` also disables a host from doing exports by VTV or Management Class.

☞ **Note:** Specifying `NOMIGRAT` also causes `NORECLAM` to be set; for more information, see "Hosts Disabled for Reclamation" on page 59.

`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.

VTV Residency Interval before Automatic Migration Candidacy. By default, VSM selects VTVs for migration. You can, however, specify how long (in hours) VTCS attempts to keep a VTV as VTSS-resident before it becomes an automatic migration candidate.

Use the `RESTIME` parameter of the `MGMTclas` statement to set this policy. The `RESTIME` value in a VTV's Management Class sets the recommended interval that the VTV remains VTSS-resident from 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.

☞ **Note:** Note the following:

- 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`. For more information, see page 23.
- 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 Migrate VTVs On Dismount" on page 54.

The following example shows how the `RESTIME` parameter works:

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.

6. 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.

Immediately Migrate VTVs On Dismount. With the `IMMEDmig` parameter of the `MGMTclas` statement, you can specify whether VSM 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.

The following are guidelines for setting `IMMEDmig`:

- Specify `NO` (the default) if you *do not* want immediately migration and you *do* want other migration policies to determine your migration strategy.

- Specify `KEEP` if you want immediate migration and want to keep copies of the migrated VTVs resident on the VTSS until they become eligible for deletion.

Caution: `IMMEDmig KEEP` ensures that VTVs are immediately migrated and kept VTSS-resident; however, it does not free up VTSS space, may increase I/O to the RTDs, uses up MVC space more quickly, and may also increase the need for MVC space reclamation.

- Specify `DELETE` if you want immediate migration and want to delete VTVs from the VTSS after migration.

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 "VTV Migration and Consolidation Policies" on page 52.

- `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.

For information on setting this policy with the `MGMTclas` statement, see "Creating and Using VSM Management and Storage Classes" on page 96.

MVC Retain Interval. This policy specifies how long VTCS will retain an MVC on an RTD in idle mode after a migration. Retaining the MVC can reduce MVC mounts.

Note: When VTCS shuts down, VTCS dismounts all MVCs regardless of the MVC retain interval.

Use the `CONFIG VTSS RETAIN` parameter to set this policy.

Maximum VTVs per MVC. This policy specifies the maximum number of VTVs that VSM will migrate to a migration MVC or consolidate to a consolidation MVC. This policy applies to all MVCs and, at the time you set the policy, applies only to future migrations. That is, it will not lower the number of VTVs already migrated to an MVC. If the policy is not specified, the default is 32000 VTVs per MVC unless the available MVC space is less than any remaining current VTSS resident VTV.

Generally, use the default setting to allow VSM to automatically manage VTV stacking. However, specifying a maximum value lower than the default may improve recall performance in some situations (for example, in a VSM system where all MVCs are type `STK2P`). Note, however, that a very low maximum value can reduce that percentage of usable MVC space. If the maximum VTVs per MVC is exceeded, then usable space is reported as 0%.

Note: Use the `CONFIG GLOBAL MAXVTV` parameter to set this policy.

ACS and Media Type of MVCs for Migration and Reclamation. You can specify the ACS and a preference list of media types for MVCs for migration and reclamation with the `MIGpol` parameter of the `MGMTclas` statement.

For procedures to set this policy, see “Creating and Using VSM Management and Storage Classes” on page 96.

Number of Migration Copies. This policy specifies how many VTV copies VSM to separate MVCs (up to a maximum of 4 copies). Multiple copies provides multiple copies of a VTV on separate physical cartridges, but it also increases the MVC space required for migrated VTVs. Note that a consolidated VTV will *not* be copied multiple times for migration, even if multiple migration copies is specified for migration; only one copy will be migrated. You specify 1 to 4 migration copies on the `MIGpol` parameter of the `STORCLAS` statement.

To enable greater than 2 VTV copies and 800 Mb VTVs:

1. **Ensure that the HSC CDS is VSM Extended Format.**

   For more information, see "Converting the Formatted CDS to VSM Extended Format" on page 83.

2. **Specify the following on the `CONFIG` statement:**

   `CDSLEVEL(V6ABOVE)`

Caution: Note that if you want to run VTCS 6.0, 5.1, and/or 5.0 concurrently against a shared CDS, you **cannot specify** `CDSLEVEL(V6ABOVE)` because this parameter does not apply to releases prior to VTCS 6.0!

Also note that if you have a 6.0 system in "4 copy" mode that you then want to run concurrently in shared CDS mode with 5.0 and/or 5.1, you must:

1. Rerun `CONFIG` **without** specifying `CDSLEVEL(V6ABOVE)`.
2. Run HSC `MERGEcds` to create a new CDS.

Note: You can also specify duplexing for one or more VTVs on the `DUPlex` parameters of the `MGMTclas` statement. If the Management Class you use for consolidation specifies the `DUPlex` parameter, duplexing is ignored for consolidation for this Management Class but duplexing *is* supported for migration for this Management Class.

For procedures to set this policy, see "Creating and Using VSM Management and Storage Classes" on page 96.

Migrate Duplexed VTVs to Separate ACSs. You can specify whether duplexed VTVs go to different MVCs in the same ACS or different MVCs in separate ACSs with the `ACSlist` parameter of the `MGMTclas` statement.

For procedures to set this policy, see "Creating and Using VSM Management and Storage Classes" on page 96.

Output MVC ACS and Media for VTV Consolidation.

This policy specifies the Storage Class that determines the output MVC ACS and media for VTV consolidation. You set this policy on the `CONTGT` parameter of the `MGMTclas` statement.

For procedures to set this policy, see "Creating and Using VSM Management and Storage Classes" on page 96 and "Using Storage and Management Classes to Specify the Source and Target MVC for VTV Consolidation" on page 100.

Source MVC ACS and Media for Consolidation of Migrated Duplexed VTVs.

This policy specifies the Storage Class that determines the source MVC ACS and media for consolidation of VTVs that are migrated and duplexed to two different MVC locations or media types. You set this policy on the `MIGpol` and `CONSRC` parameters of the `MGMTclas` statement.

For procedures to set this policy, see "Creating and Using VSM Management and Storage Classes" on page 96 and "Using Storage and Management Classes to Specify the Source and Target MVC for VTV Consolidation" on page 100.

VTV Recall Policy

By default, VTCS recalls VTVs with read data checks. You can, however, specify whether VTCS recalls VTVs with read data checks on:

- The `GLOBAL` statement of `CONFIG`.
- `RECALL`.
- `MVCDRAIN`.
- `CONSolid`.
- `EXPORT`.

During MVC reclaims, VTCS will never recall VTVs with read data checks, regardless of the `RECALWER` setting on the `CONFIG GLOBAL` statement.

VTV Maximum Size Policy

You can use the `MAXVtvsz` parameter of the `MGMTclas` statement to specify the maximum VTV size in megabytes (400 or 800).

Note:

- The `MAXVtvsz` parameter applies to **only** VSM3s and VSM4s with the following microcode levels:
  - For VSM3s: microcode level N01.00.69.04 **or** microcode level N01.00.71.00 and above
  - For VSM4s: microcode level D01.00.04.03 **or** microcode level D01.00.06.03 and above
- The size of a VTV will only change once it goes through a scratch cycle. Therefore, if you change the Management Class and `DISP=MOD`, then it will still retain the original size.

MVC Policies

The following sections describe these policies:

- “Maximum MVCs Concurrently Processed for Reclamation and Drain”
- “Hosts Disabled for Reclamation”
- “Free MVCs Threshold - Starts Automatic Space Reclamation” on page 60
- “MVC Fragmented Space Threshold- Determines MVC Eligibility for Reclamation” on page 60
- “Eligible/Total MVCs Threshold - Starts Automatic Space Reclamation” on page 61
- “Maximum MVCs Processed Per Reclaim” on page 62

Note: Reclamation turns fragmented MVC space (space that contains non-current VTVs) into usable space (writable MVC space). MVC reports and `Query` show the percentages of MVC space that is fragmented, used (space that contains current VTVs), available, and usable. Note that usable space may be zero even if there is still space physically available. For example, if the maximum VTVs per MVC is exceeded, then usable space is reported as 0%. You set maximum VTVs per MVC as described in “Maximum VTVs per MVC” on page 55. Similarly, if a data check error has been reported against an MVC, VTCS will not use this MVC for output and usable space is reported as 0%.

Maximum MVCs Concurrently Processed for Reclamation and Drain. This policy specifies the maximum number of MVCs concurrently processed for reclamation and drain. Use the `CONFIG RECLAIM CONMVC` parameter to set this policy. Valid values for the `CONMVC` parameter are 1 to 99. The default is 1. You can also use the following to override the value specified on the `CONFIG RECLAIM CONMVC` parameter:

- `MVCDRAIN`
- `RECLAIM`

Hosts Disabled for Reclamation. You can specify that a host cannot initiate automatic or demand reclaim processing (the host can still do demand MVC drains via `MVCDRain`). Use the `CONFIG HOST NORECLAM` parameter to set this policy.

Free MVCs Threshold - Starts Automatic Space Reclamation. This policy specifies the minimum number of free MVCs in the MVC pool. A free MVC has 100% usable space and does not contain any migrated VTVs.

Use the `CONFIG GLOBAL MVCFREE` parameter to set this policy. Valid values are 0 to 255. The default is 40.

VTCS checks this value for each ACS. VTCS issues message `SLS6616I` and starts an automatic space reclamation if *both* of the following occurs:

- Free MVCs is equal to or less than the value specified on `CONFIG MVCFREE.`
- There is at least one eligible MVC as defined by the `CONFIG RECLAIM THRESHLD` parameter; for more information, see "MVC Fragmented Space Threshold- Determines MVC Eligibility for Reclamation" on page 60.

Note: If you set `MVCFREE`=0, VTCS actually uses the default value (40).

Hint: StorageTek recommends that you ensure that your MVC pool always has *at least* one eligible MVC for each MVC media type.

Otherwise, you may need to change the `CONFIG GLOBAL MVCFREE` value, add more MVCs to the pool, or both. You can use `Query` to display the number of free MVCs in your MVC pool.

MVC Fragmented Space Threshold- Determines MVC Eligibility for Reclamation. This policy specifies the fragmented space threshold (as a percentage) that determines when an MVC is eligible for demand or automatic reclamation.

Use the `CONFIG RECLAIM THRESHLD` parameter to set this policy. If fragmented space on an MVC exceeds the value specified on `CONFIG THRESHLD`, VTCS makes the MVC eligible for reclamation. You can use `RECLAIM` to change the `CONFIG` settings for the `THRESHLD` parameter.

Regardless of the percentage of fragmented space on an MVC versus this value, however, VTCS also considers where fragmented space occurs. For example, if the first fragmented space is near the end of the MVC, VTCS may process the MVC before an MVC with more total fragmented space.

Hint: You can use `Query` to display the MVCs eligible for reclamation in your MVC pool. You can also use `Query` to display information about MVC status and space.

Eligible/Total MVCs Threshold - Starts Automatic Space Reclamation. This policy specifies a percentage value, which is equal to:

(*Reclaim Candidates*/*Reclaim Candidates + Free MVCs*) * 100

Where:

Reclaim Candidates
: is the number of Reclaim Candidates determined by the `CONFIG RECLAIM THRESHLD` parameter. For more information, see "MVC Fragmented Space Threshold- Determines MVC Eligibility for Reclamation" on page 60.

Reclaim Candidates + Free MVCs
: equals the number of Reclaim Candidates *plus* the number of free MVCs. A free MVC:

- Has 100% usable space and does not contain any migrated VTVs.
- Is defined as described in "Define Available MVCs with CONFIG" on page 28 and "Define the MVC Pool" on page 29.
- Is writeable.
- Is resident in the ACS.

Use the `CONFIG RECLAIM START` parameter to set this policy. Valid values for the `START` parameter are 1 to 98%. The default is 35%.

For each ACS (not globally for all ACSs), VTCS issues message `SLS6616I` and starts an automatic space reclamation if *both* of the following occurs:

- The actual value of (*Reclaim Candidates*/*Reclaim Candidates + Free MVCs*) * 100 exceeds the value specified on `CONFIG RECLAIM START` parameter.
- The number of eligible MVCs exceeds the value specified on the `MAXMVC` parameter; for more information, see "Maximum MVCs Processed Per Reclaim" on page 62.

Note: The only exception to the above two conditions occurs if an SLS6699 message indicates a critical shortage of free MVCs, in which case automatic reclamation will start anyway.

The following are general guidelines for specifying values for the `START` parameter:

- A *low* value (for example, 5%), starts automatic space reclamation when there are *few* eligible MVCs compared to free MVCs *unless* you set the `MAXMVC` value high compared to the number of eligible MVCs.
- A *high* value (for example, 95%), starts automatic space reclamation when there are *many* eligible MVCs compared to free MVCs unless you set the `MAXMVC` value *very* high and your MVC pool is *very* small.

Hint: You can use `Query MVCPOOL` to display eligible and free MVCs.

Maximum MVCs Processed Per Reclaim. Automatic and demand space reclamation processes one MVC at a time. You can, however, control the maximum number of MVCs that will be processed in a single space reclamation run with the `CONFIG RECLAIM MAXMVC` parameter. You can use `RECLAIM` to change the `CONFIG` setting for the `MAXMVC` parameter.

For automatic space reclamation to start via the `CONFIG RECLAIM START` parameter setting, the number of eligible MVCs (determined by the `CONFIG RECLAIM THRESHLD` parameter) must also exceed the `MAXMVC` value. For more information, see "Eligible/Total MVCs Threshold - Starts Automatic Space Reclamation" on page 61.

The following are general guidelines for specifying values for the `MAXMVC` parameter:

- A *low* value reclaims *fewer* MVCs in a single run, but may have *negligible* effect on migrations and recalls and may start automatic space reclamation *more* frequently; for more information, see "Eligible/Total MVCs Threshold - Starts Automatic Space Reclamation" on page 61.
- A *high* value reclaims *more* MVCs in a single run, but may have *considerable* effect on migrations and recalls and may start automatic space reclamation *less* frequently; for more information, see "Eligible/Total MVCs Threshold - Starts Automatic Space Reclamation" on page 61.

Hint: You can use `Query` to display eligible and free MVCs.

Storage Class Preferencing

Use the `STORSEL` statement to define a VTSS usage rule that applies to the Storage Class list and its preferencing specified on a referenced `STORLST` control statement.

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`

You can use Storage Class preferencing to influence RTD selection for specific mounts for automatic recalls. You can also use Storage Class preferencing to influence MVC selection for the following functions:

- Demand recall
- Export
- Consolidation

Chapter 2. Preparing for Installation

Before doing the tasks described in Chapter 3 “Installing HSC and VTCS”, complete the preparation tasks described in the following sections:

- “Defining A Security System User ID for HSC, SMC, and VTCS” on page 66
- “Configuring MVS Device Numbers and Esoterics” on page 66
- “Setting the MVS Missing Interrupt Handler (MIH) Value” on page 67
- “Specifying the Region Size” on page 67

Defining A Security System User ID for HSC, SMC, and VTCS

VSM software (HSC, SMC, and VTCS) uses the MVS System Authorization Facility (SAF) to control the usage of MVCs. Before mounting an MVC and before issuing writes to an MVC, VSM issues a SAF query to verify that the HSC user has UPDATE authority for the MVC.

You must define a security system user ID to be used by the SAF queries. All VSM SAF queries are issued on behalf of HSC, so the security system user ID must be associated with the HSC started task. Refer to your security system documentation for details on how to associate a security system user ID with the HSC started task.

You must also set up TAPEVOL profiles to ensure that VSM is authorized to mount MVCs, to ensure that VSM is authorized to update MVCs, and to guard against accidental overwrites of MVCs; for more information, see "Defining MVC Pool Volser Authority" on page 112.

Warning: Depending on the default settings of your security system, VSM may not be able to mount and to write to MVCs until you have defined a security system user ID for HSC and TAPEVOL profiles for the MVCs.

Configuring MVS Device Numbers and Esoterics

The following sections tell how to use the `HCD` facility to do the following:

- Assign MVS device numbers to VTDs and shared RTDs.
- Associate VTD MVS device numbers and esoterics.

You determined these values in "Determining VSM Configuration Values" on page 14 and recorded them in Table 25. on page 125. See your IBM documentation for more information on the `HCD` facility.

Assigning MVS Device Numbers to VTDs

Use the `HCD` facility to assign MVS 3490E device numbers to your VSM system's VTDs. You determined these device numbers in "VTD Unit Addresses" on page 16. For more information about assigning these device numbers, see *Virtual Storage Manager Planning, Implementation, and Usage Guide*.

Associating VTD MVS Device Numbers and Esoterics

If you use esoteric substitution to allocate VTDs as described in "VSM Esoterics and Esoteric Substitution" on page 17, use the `HCD` facility to associate each esoteric name with the MVS device numbers for the VTDs that you have chosen for that esoteric.

Assigning MVS Device Numbers to RTDs

Use the `HCD` facility to assign MVS device numbers to these RTDs.

Hint: You must use the same unit addresses you determined for these transports for LIBGEN updates as described in "RTD Unit Addresses and Identifiers" on page 24.

Setting the MVS Missing Interrupt Handler (MIH) Value

The VTSS's internal error recovery procedures requires the MVS missing-interrupt handler (MIH) value to be 20 minutes. You set this value by modifying the `MIH` parameter in SYS1.PARMLIB member IECIOS*xx*.

☞ Adjust applications running on your system that detect missing interrupts and that are independent of the system MIH setting to allow a five-minute MIH value.

Specifying the Region Size

StorageTek recommends that you run HSC/VTCS with a region size of **at least** 6 MB except if you are running utilities or commands that manipulate manifest files, in which case you need the maximum region size your system will allow.

Chapter 3. Installing HSC and VTCS

Before you install the software, complete the pre-installation tasks described in the following sections:

- "Reviewing Coexistence Requirements" on page 70
- "Verifying Installation Materials" on page 70

Next, install HSC, HSC maintenance, and VTCS 6.1.0 as described in the following sections:

- "Installing HSC and SMC" on page 72
- "SMC Installation Considerations" on page 72
- "Receiving the VTCS 6.1.0 FMID" on page 73
- "Receiving the VTCS 6.1.0 Service" on page 73
- "Creating the VTCS 6.1.0 LINKLIB Data Sets and Defining Libraries to the HSC Target Zone" on page 74
- "Applying the VTCS 6.1.0 FMID" on page 75
- "Accepting the VTCS 6.1.0 FMID" on page 76
- "Applying the VTCS 6.1.0 Service" on page 77
- "Adding SWSLINK to the Authorized Program List" on page 78
- "Modifying the HSC Startup Procedure to include the VTCS 6.1.0 LINKLIB" on page 78

Caution: In a VSM configuration with multiple hosts that share the same HSC CDS, StorageTek strongly recommends that you install VTCS on all MVS hosts by completing all tasks described in this chapter. Installing VTCS on all MVS hosts ensures that these hosts cannot scratch an MVC.

Reviewing Coexistence Requirements

For more information, see "VTCS System Software Requirements" on page 10.

Note: The VSM Extended Format CDS **is required for VTCS 6.1.0. Also note that** after you convert the CDS to VSM Extended Format, you **cannot** run VTCS 4.0.0 or lower against the converted CDS. For more information, see "Converting the Formatted CDS to VSM Extended Format" on page 83.

Verifying Installation Materials

The installation materials include the following

- VTCS 6.1 Installation Base Tape. This tape, which is a single standard label tape with a volume serial number of `SWS6100`, contains VTCS 6.1.
- VTCS Service Tape, which contains VTCS PTFs since the base tape was created.

Note:

- Contact StorageTek Software Support for information about additional PTFs that might be required before installing the NCS product components. See the *Requesting Help from Software Support* guide for information about contacting StorageTek for technical support and for requesting changes to software products.
- If you are using HSC or MVS/CSC, the SMC software **must** be installed.
- StorageTek recommends that you use the MVS Program Binder when installing NCS products and maintenance. Failure to do so may result in link-editing errors.

NCS/VTCS Installation Tape Contents

As described in “Verifying Installation Materials” on page 70, VTCS 6.1.0 is delivered on two tapes. The Installation Base Tape contains the VTCS 6.1.0 FMIDs and the Service Tape contains PTFs for the VTCS 6.1.0 FMID. You **must** install both tapes.

Table 15 lists the files included on the NCS/VTCS 6.1.0 product tape.

Table 15. NCS/VTCS 6.1.0 Installation Base- Tape Contents

File	Data Set Name	Description
1	`SMPMCS`	SMP/E control statements
2	`SWS6100.F1`	SWS6100 JCLIN
3	`SWS6100.F2`	SWS6100 SAMPLIB members (automatically installed in the HSC SAMPLIB)
4	`SWS6100.F3`	SWS6100 MACLIB members (automatically installed in the HSC MACLIB)
5	`SWS6100.F4`	SWS6100 object modules

Note: The VTCS 6.1.0 installation automatically installs the following VTCS members in the HSC SAMPLIB:

`SWSJCRDB`
: Sample `CONFIG` utility JCL

`SWSJMVCR`
: Sample `MVCRPT` utility JCL

`SWSJVTVR`
: Sample `VTVRPT` utility JCL

Table 16 lists the files included on the VTCS 6.1.0 service tape.

Table 16. NCS/VTCS 6.1.0 Service Tape Contents

File	Data Set Name	Description
1	`SYSMODS`	SYSMODS

VTCS FMIDs

The VTCS 6.1.0 software is packaged in standard SMP/E format. The NCS/VTCS 6.1.0 installation tape includes the following VTCS FMID:

`SWS6100`
: The `SWS6100` function contains the VTCS load modules for VTCS 6.1.0 running with HSC 6.1.0.

Note: The SWS6100 FMID is a subsidiary HSC 6.1.0 FMID (`SOS6100` for HSC 6.1.0) and you must apply the SWS6100 FMID to the same SMP/E zone as HSC.

The VTCS service tape contains SYSMODS (all maintenance since base).

Installing HSC and SMC

Install HSC, SMC and all HSC/SMC maintenance. For more information, see Chapter 5, “Installing the MVS/HSC Functions” of *NCS Installation Guide.*

StorageTek **strongly recommends** upgrading HSC and the LibraryStation and MVS/CSC components (if you intend to connect MVS/CSC clients to VSM) to the current maintenance level before installing VTCS. Also note that StorageTek **requires** you to install SMC, which is required to perform allocation influencing and message interception on MVS.

Note: If you are running at an SMP/E level lower than that supplied with OS/390 Version 2.5, you *must* accept the HSC FMID (`SOS6100` for HSC 6.1.0) and all HSC maintenance before installing VTCS 6.1.0.

SMC Installation Considerations

Caution: Ensure that you create and install the SMC JES3 IATUX09 User Exit as described in *NCS Installation Guide.* This exit modification enables deferred mount processing, without which VTV mounts may fail.

Receiving the VTCS 6.1.0 FMID

Use the example JCL in Figure 2 to receive the VTCS 6.1.0 FMID.

```
//S1       EXEC PGM=GIMSMP,
//         PARM='PROCESS=WAIT',
//         DYNAMNBR=120
//SMPCSI   DD DISP=SHR,DSN=hsc.CSI
//SMPPTFIN DD DISP=SHR,DSN=SMPMCS,VOL=SER=SWS6100,UNIT=3480,
//         LABEL=(1,SL)
//SMPCNTL  DD *
  SET    BOUNDARY (GLOBAL) .
  RECEIVE S(SWS6100).
```

Figure 2. ***JCL Example: Receiving the VTCS 6.1.0 FMID***

Receiving the VTCS 6.1.0 Service

Use the example JCL in Figure 3 to receive the VTCS 6.1.0 service.

```
//S1       EXEC PGM=GIMSMP,
//         PARM='PROCESS=WAIT',
//         DYNAMNBR=120
//SMPCSI   DD DISP=SHR,DSN=hsc.CSI
//SMPPTFIN DD DISP=SHR,DSN=SYSMODS,VOL=SER=anyvol,UNIT=3480,
//     LABEL=(1,NL),
//     DCB=(LRECL=80,BLKSIZE=7200,RECFM=FB)
//SMPHOLD DD DISP=SHR,DSN=HOLDDATA,UNIT=AFF=SMPPTFIN,
//     LABEL=(4,NL),VOL=SER=anyvol,
//     DCB=(LRECL=80,BLKSIZE=7200,RECFM=FB)
//SMPCNTL  DD *
  SET    BOUNDARY (GLOBAL) .
  RECEIVE SYSMODS HOLDDATA.
```

Figure 3. ***JCL Example: Receiving the VTCS 6.1.0 service***

Note: StorageTek **strongly recommends** that you check the StorageTek Customer Resource Center and download any additional maintenance for VTCS posted since the date of the Service Tape you received. You can execute the `RECEIVE` step again to receive this additional maintenance.

Creating the VTCS 6.1.0 LINKLIB Data Sets and Defining Libraries to the HSC Target Zone

Use the example JCL in Figure 4 to create the VTCS 6.1.0 LINKLIB (SWSLINK) data sets and to define these data sets and the HSC distribution library (ASLSLINK) to the HSC target zone.

☞ StorageTek has intentionally chosen to use the existing HSC distribution library (ASLSLINK).

```
//*==>>EXPECT RC=4 ON FIRST RUN
//DELETE   EXEC PGM=IEFBR14
//SWSLINK1 DD  DISP=(MOD,DELETE),
//             SPACE=(TRK,0),
//             DSN=hlq.SWSLINK,
//             UNIT=SYSDA
//ALLOC    EXEC PGM=IEFBR14,COND=(0,NE)
//SWSLINK1 DD  DCB=(BLKSIZE=23476,RECFM=U),
//             DISP=(NEW,CATLG,DELETE),
//             SPACE=(CYL,(1,1,5)),
//             DSN=hlq.SWSLINK,
//             UNIT=SYSDA
//ALTER1   EXEC PGM=GIMSMP
//SMPCSI   DD   DISP=SHR,DSN=hsc.CSI
//SMPCNTL  DD   *
SET BDY(tzone) .
  UCLIN .

  REP DDDEF(SWSLINK) DATASET(hlq.SWSLINK) SHR .
  REP DDDEF(ASLSLINK) DATASET(hlq.ASLSLINK) SHR .

  ENDUCL .
```

Figure 4. ***JCL Example: Creating the SWSLINK data sets and defining libraries to the HSC target zone***

Applying the VTCS 6.1.0 FMID

Use the example JCL in Figure 5 to apply the FMID for VTCS 6.1.0.

```
//APPTSOS  EXEC PGM=GIMSMP,
//         PARM='PROCESS=WAIT',
//         DYNAMNBR=120
//SMPCSI   DD DISP=SHR,DSN=hsc.CSI
//SMPCNTL  DD *
  SET    BOUNDARY (tzone).
  APPLY
  S(SWS6100)
       GROUPEXTEND.
```

Figure 5. ***JCL Example: Applying the VTCS 6.1.0 FMID***

Note: Because the VTCS FMID is a subsidiary HSC FMID, SMPE will also apply any VTCS PTFs called out by conditional COREQS in the HSC PTFs already applied to your system. If any of these PTFs have HOLDDATA, you will receive a GIM35965I message for each, and the APPLY will fail. This is not an error condition. If you encounter this condition, please review the individual PTF cover letters, note any additional action(s) to be taken, and repeat the APPLY step with the following parameters:

```
APPLY S(SWS6100) GROUPEXTEND
           BYPASS(HOLDSYSTEM) .
```

Accepting the VTCS 6.1.0 FMID

Use the example JCL in Figure 6 to accept the VTCS 6.1.0 FMID.

```
//APPTSOS  EXEC PGM=GIMSMP,
//         PARM='PROCESS=WAIT',
//         DYNAMNBR=120
//SMPCSI   DD DISP=SHR,DSN=hsc.CSI
//SMPCNTL  DD *
  SET    BOUNDARY (dzone).
  ACCEPT
  S(SWS6100)
       GROUPEXTEND.
```

Figure 6. ***JCL Example: Accepting the VTCS 6.1.0 FMID***

Note: If you are running at an SMP/E level lower than that supplied with OS/390 Version 2.5, you *must* accept the HSC FMID and all HSC maintenance before installing VTCS 6.1.0. For more information, see “Installing HSC and SMC” on page 72.

Also note that because the VTCS FMID is a subsidiary HSC FMID, SMPE will also accept any VTCS PTFs called out by conditional COREQS in the HSC PTFs already accepted on your system. If any of these PTFs have HOLDDATA, you will receive a GIM35965I message for each, and the ACCEPT will fail. This is not an error condition. If you encounter this condition, please review the individual PTF cover letters, note any additional action(s) to be taken, and repeat the ACCEPT step with the following parameters:

```
ACCEPT S(SWS6100) GROUPEXTEND
            BYPASS(HOLDSYSTEM) .
```

Applying the VTCS 6.1.0 Service

Use the example JCL in Figure 7 to apply the VTCS 6.1.0 service:

```
//APPTPTF  EXEC PGM=GIMSMP,
//         PARM='PROCESS=WAIT',
//         DYNAMNBR=120
//SMPCSI   DD DISP=SHR,DSN=hsc.CSI
//SMPCNTL  DD *
  SET    BOUNDARY (tzone).
  APPLY PTFs
  FORFMID (SWS6100)
GROUPEXTEND.
```

Figure 7. JCL Example: Applying the VTCS 6.1.0 service

Note: If any of the PTFs to be applied have HOLDDATA, you will receive a GIM35965I message for each, and the APPLY will fail. This is not an error condition. If you encounter this condition, please review the individual PTF cover letters, note any additional action(s) to be taken, and repeat the APPLY step with the following parameters:

```
APPLY PTFS FORFMID(SWS6100)
           GROUPEXTEND
           BYPASS(HOLDSYSTEM) .
```

Adding SWSLINK to the Authorized Program List

VTCS must run as an authorized program, which you do by adding the VTCS Link Library (SWSLINK) to the authorized program list on your system in one of two ways:

- Dynamically
- “Using IEAAPFxx to APF Authorize the SWSLINK”
- “Using PROGxx to APF Authorize SWSLINK”

Using IEAAPF*xx* to APF Authorize the SWSLINK

To use the IEAAPF*xx* member of SYS1.PARMLIB to authorize the SWSLINK, add the following entry to that list with your HLQ and volser:

```
your.SWSLINK volser
```

Note: If SWSLINK resides on an SMS-managed volume, you do not need to specify a volume in the IEAAPF*xx* member as follows:

```
your.SWSLINK
```

Using PROG*xx* to APF Authorize SWSLINK

To use the PROG*xx* member of SYS1.PARMLIB to authorize SWSLINK, add the following entries to that list with your HLQ and volser:

```
APF ADD
  DSNAME(your.SLSLINK)
  VOLUME(volser)
```

Note: If SWSLINK resides on an SMS-managed volume, you do not need to specify a volume in the PROG*xx* member as follows:

```
APF ADD
  DSNAME(your.SLSLINK)
  VOLUME SMS
```

Modifying the HSC Startup Procedure to include the VTCS 6.1.0 LINKLIB

Use the example JCL in Figure 8 as an example of how to modify the HSC startup procedure to start VTCS. Include the VTCS 6.1.0 LINKLIB (SWSLINK) in the STEPLIB **before** the HSC LINKLIB (SLSLINK).

```
//SLSO     PROC PROG=SLSBINIT
//IEFPROC  EXEC PGM=&PROG,TIME=1440,DPRTY=(7,5),
//         PARM='SSYS(SLSO) E(E086) F(23) M(00)'
//STEPLIB  DD DSN=hlq.SWSLINK,DISP=SHR
//         DD DSN=hlq.SLSLINK,DISP=SHR
```

Figure 8. ***JCL Example: Modifying the HSC started task to include the SWSLINK library***

Chapter 4. Reconfiguring NCS

Before you configure VSM, you must do some or all of the HSC reconfiguration tasks described in the following sections:

- "Creating or Updating the HSC LIBGEN" on page 80
- "Verifying the LIBGEN" on page 82
- "Formatting the New CDS" on page 82
- "Converting the Formatted CDS to VSM Extended Format" on page 83
- "Updating the HSC Definition Data Sets" on page 95
- "Creating and Using VSM Management and Storage Classes" on page 96
- "Creating and Using Named MVC Pools" on page 101
- "Updating the HSC PARMLIB Member (SLSSYSxx)" on page 104
- "Connecting MVS/CSC Clients to VSM" on page 106
- "Connecting Non-MVS/CSC Clients to VSM" on page 108
- "Restarting NCS" on page 109

Hint: Except for updating the HSC definition data sets, you can do all these tasks before you install VTCS. Most of the tasks in this Chapter require you to specify VSM system values that you determined on page 14 and recorded in Table 25. on page 125.

Also note that if you are upgrading from a previous release of VTCS, you may not need to do all the tasks in this chapter. For example, if you are not adding RTDs to your configuration, you do not need to update the HSC LIBGEN.

Creating or Updating the HSC LIBGEN

If your system's RTDs are new transports, you must update the HSC LIBGEN by adding a SLIDRIVS macro to define the device addresses you determined in "RTD Unit Addresses and Identifiers" on page 24. Similarly, if you have made other hardware changes (for example, adding or removing LSMs), you must update the related LIBGEN macros as described in Step 2, below. If you are converting the CDS to VSM Extended Format as described in "Converting the Formatted CDS to VSM Extended Format" on page 83, you must create a new CDS. For more information, see *HSC System Programmer's Guide for MVS*.

To update the HSC LIBGEN to define new transports that are RTDs:

1. **Run the HSC Database Decompile (LIBGEN) Utility to create LIBGEN macro statements from your existing CDS.**

   Do *not* edit the original LIBGEN, because if the SET Utility was used to change the library configuration stored in the CDS, the original LIBGEN no longer matches the CDS. For more information about the Database Decompile Utility, see *HSC System Programmer's Guide for MVS*.

2. **After you run the HSC Database Decompile Utility, add a SLIDRIVS macro to define the RTD device addresses.**

   You may also need to update related LIBGEN macros, such as the SLIACS, SLILSM and SLIDLIST macros. For more information about the LIBGEN macros, see Chapter 4, "Creating the Library Configuration File (LIBGEN)" of *HSC Configuration Guide for MVS.*

Note: You can specify that Nearline transports can only be used as RTDs on the SLIACS macro as shown below:

```
SLIACS VSMONLY=YES,ACSDRV=(esoteric0,...,esoteric15),LSM=(....)
```

As shown in this example, in the HSC SLIACS macro:

- The VSMONLY=YES parameter specifies that the RTDs in the ACS are attached only to a VSM system for one or more of the HOSTs connected to this ACS. VSMONLY=NO is the default.
- The ACSDRV parameter specifies the esoteric name of each host that refers to the transports attached to this ACS. A comma is a placeholder for any esoteric name not specified **except** when VSMONLY=YES is specified. In this case, the esoteric "placeholder" is ignored (set to blanks), which then "dedicates" the ACS to VSM use. If VSMONLY=NO, then the esoteric of the first host system is substituted when an esoteric is omitted for a host.

Also note that you can use the HSC `SET ACS` utility to you specify that Nearline transports can only be used as RTDs as follows:

- The `ACSDRV` parameter specifies the esoteric name of the host that refers to the transports attached to this ACS. A () specifies that the esoteric for the specified host and ACS is set to blank only when `VSMONLY(YES)` is specified on the `SET ACS` utility, which then "dedicates" the ACS to VSM use. If `VSMONLY(NO)`, then the esoteric of the first host system is substituted when an esoteric is omitted for a host.
- The `VSMONLY` parameter is added. `VSMONLY(YES)` specifies that the RTDs in the ACS are attached only to a VSM if the `ACSDRV` parameter specifies () for the esoteric for this host. `VSMONLY=NO` is the default.

Finally, also note that if you want to have MVS/CSC clients connected to an ACS whose RTDs are dedicated to VSM:

- In the `LIBDEV` startup parameter, for an ACS that is only attached to VSM, the position ACS esoteric name must be blank.
- Omit the `LIBUNIT` startup parameter for an ACS that is only attached to VSM.
- Omit the `UNITMAP` startup parameter for an ACS that is only attached to VSM if *both* of following are true:
  - Clients use the same MVS device numbers defined for this ACS's RTDs.
  - Your configuration has no cartridge tape UCB defined with an MVS device number that matches a device number defined for this ACS's RTDs.

  If either or both of these statements **is not** true, then you must create both a `LIBUNIT` and a `UNITMAP` statement to map the drives to the client addresses.

Hint: As an alternative to MVS/CSC and LibraryStation, you can use the SMC 6.1 client/server capabilities. For more information, see *SMC Configuration and Administration Guide*.

3. **After you update the `LIBGEN` macros, reassemble and link–edit the `LIBGEN` file.**

   For more information, see "LIBGEN Process Verification" in Chapter 4, "Creating the Library Configuration File (LIBGEN)" of *HSC Configuration Guide for MVS*.

Verifying the LIBGEN

After you assemble and link edit the LIBGEN file, run the SLIVERFY program to verify the LIBGEN. For more information, see “Verifying the Library Generation” in Chapter 4, “Creating the Library Configuration File (LIBGEN)” of *HSC Configuration Guide.*

Caution: Before you run SLIVERFY, if your system’s RTDs are new transports that you will share with MVS, you must install them and define their MVS unit addresses via the HCD facility as described in “Assigning MVS Device Numbers to RTDs” on page 66.

Formatting the New CDS

In “HSC CDS DASD Space” on page 43, you determined the size of the new CDS to support your VSM system. After you verify the LIBGEN, you must format a new CDS to this size by using the HSC SLICREAT macro. For more information on the HSC SLICREAT macro, see Chapter 5, “Initializing the Control Data Sets” of *HSC Configuration Guide.*

Note that:

- Before converting the CDS to VSM Extended Format as described in “Converting the Formatted CDS to VSM Extended Format” on page 83, you need to allocate a new data set for the VSM Extended Format CDS.
- If you change the CDS data set name, make sure to update the name in the HSC started task and in any other started tasks or batch jobs (such as ExPR and ExLM) that reference this data set.
- VSM **does not** support copies of the CDS at multiple sites (for example, Primary CDS at one site and Secondary at another). A link failure would allow the two sites to run independently, and VSM cannot enforce separation of all resources. This prevents reconciliation of the two divergent CDSs as can be accomplished in a pure NCS environment.

Converting the Formatted CDS to VSM Extended Format

This section tells how to convert the CDS to VSM Extended Format.

First, A Word About CDS Formats...

...and how to find out which format you have. The HSC CDS manages HSC and VTCS. There is only one CDS, but internally it has two elements – the HSC portion and the VTCS portion. The VTCS portion has been extended to support VTCS enhancements, and there are now five formats for the VTCS portion: B, C, D, E, and F. As described in Table 17. on page 84, each VTCS version supports **only** a subset of these levels. If you are, therefore, running with a mixed set of VTCS versions against a CDS it is important to ensure that the CDS is set at a level that is supported by **all** the versions being run. **Also note that** certain VTCS functions are **only** available by running with the CDS at a certain level.

How do you know which format you have? Starting with HSC 6.0, you can use the HSC D CDS command to find out. For example, D CDS displays the information shown in Figure 9.

```
.SLS0000I D CDS
.SLS2716I DATABASE INFORMATION 063

SYS00001 = SLS.HSCVJ.NCS60.DBASEPRM
  PRIVOL = CIM003     FLAGS(40) ACTIVE
SYS00003 = SLS.HSCVJ.NCS60.DBASESEC
  SECVOL = CIM003     FLAGS(40) ACTIVE
SYS00002 = SLS.HSCVJ.NCS60.DBASESBY
  SBYVOL = CIM003     FLAGS(00) INACTIVE

JOURNALING NOT ACTIVE FOR THIS SUB-SYSTEM

CDS LEVEL = 020100           DATE = 20030826
CREATE    = I791693          TIME = 15:46:47
VSM CDS LEVEL = E

ENQNAME   = STKALSQN       - SMFTYPE = 255
CLEAN PREFIX = CLN         - LABTYPE = (00) SL
RECOVERY = (40) STANDBY    - DELETE DISP = (80,40) SCRTCH
.
.
.
etc
```

HSC Format = 2.1.0

VTCS Format = E

Figure 9. ***Example Output from the HSC D CDS Command***

In Figure 9, note that there are actually two separate formats, one each for the HSC and VTCS portions of the CDS. In this example, the HSC portion is at the 2.1.0 level and the VTCS portion is at Format E.

Browsing the CDS

What if you are at a lower level than HSC 6.0? The answer is to browse the CDS directly. Browse the CDS in non-hex mode and search for `VSML`. You should find an eye-catcher like this … `VSMLxVSMHDR`. The "*x*" in the fifth position indicates the format of the VTCS portion of the CDS. For example, if the CDS contains `VSMLCVSMHDR`, it is a C format CDS.

What Are the Differences Between the Formats of the VTCS Portion of the CDS?

In "First, A Word About CDS Formats..." on page 83, we said that the VTCS portion of the CDS has been extended to support VTCS enhancements. Basically, therefore, as you go from the B format upwards, you gain more hardware support and more function as described in Table 17.

Table 17. HSC CDS - Formats of the VTCS Portion

VTCS Format	Valid VTCS/NCS Releases	VTSS Hardware	VTCS Enhancements
B	4.0, 5.0, 5.1	• VSM2 and VSM3 • VSM4 (**only** if you configure it as a VSM2/3. That is, you cannot configure greater than 64 VTDs per VTSS and/or greater than 8 RTDs per VTSS.) • RTD sharing between VTSSs.	• VSM4 in VSM2/3 compatibility mode only.
C	5.0, 5.1	• VSM2 and VSM3 • VSM4 with up to 256 VTDs per VTSS and/or up to 16 RTDs per VTSS. Note that this is **only** available with VTCS 5.1, **not** with 5.0. • RTD sharing **except** for paired RTDs (a paired RTD shares a CIP with another Nearlink connection, either an RTD or a CLINK).	• Full VSM4 Support
D	5.0, 5.1, 6.0, 6.1	• VSM2 and VSM3 • VSM4 with up to 256 VTDs per VTSS and/or up to 16 RTDs per VTSS. Note that this is **only** available with VTCS 5.1 and above, **not** with 5.0. • RTD sharing **except** for paired RTDs (a paired RTD shares a CIP with another Nearlink connection, either an RTD or a CLINK).	• Full VSM4 Support

Table 17. HSC CDS - Formats of the VTCS Portion

VTCS Format	Valid VTCS/NCS Releases	VTSS Hardware	VTCS Enhancements
E	6.0, 6.1	• VSM2 and VSM3 • VSM4 with up to 256 VTDs per VTSS and/or up to 16 RTDs per VTSS. Note that this is **only** available with VTCS 5.1 and above, **not** with 5.0. • RTD sharing **except** for paired RTDs (a paired RTD shares a CIP with another Nearlink connection, either an RTD or a CLINK).	• Full VSM4 Support • 4 MVC copies • 800 Mb VTVs (see for additional requirements) **Note**: 4 MVC copies and/or 800 Mb VTVs also **require** that you specify CDSLEVEL(V6ABOVE) on the 6.0 CONFIG statement.
F	6.1	• VSM2 and VSM3 • VSM4 with up to 256 VTDs per VTSS and/or up to 16 RTDs per VTSS. Note that this is **only** available with VTCS 5.1 and above, **not** with 5.0. • RTD sharing **except** for paired RTDs (a paired RTD shares a CIP with another Nearlink connection, either an RTD or a CLINK).	• Full VSM4 Support • 4 MVC copies • 800 Mb VTVs (see for additional requirements) • Near Continuous Operations (NCO) • Bi-directional clustering • **Note**: 4 MVC copies and/or 800 Mb VTVs and/or NCO and/or bi-directional clustering also **require** that you specify CDSLEVEL(V61ABOVE) on the 6.1 CONFIG statement.

CDS Conversion Guidelines

Note the following:

- C, D, E, and F formats are considered "VSM Extended Format CDSs." The VSM Extended Format CDS **is required for VTCS 6.1. Also note that** after you convert the CDS to VSM Extended Format, you **cannot** run VTCS 4.0 or lower against the converted CDS. VTCS 4.0 and below **is incompatible with** and **will not initialize with** the VSM Extended Format CDS. If you are a **new** VTCS 6.1 customer, VSM Extended Format is the default, so no conversion is required.
- **Note that** converting from a B or C level requires a reorganization of the internal CDS structure for the VTCS area. This reorganization requires that:

  1. A new CDS set of data sets is created
  2. The new CDS is configured for HSC and VTCS at the required target level.
  3. The new CDS is populated with data from the old CDS by the `MERGECDS` utility.

  For all other conversions, creating a new CDS is **not** required.
- Also note that you **cannot use** `CD EXPAND` to create the new CDS. For more information about creating a new CDS, see *HSC System Programmer's Guide for MVS*.
- You use the HSC `SLICREAT` macro to format the new CDS. For more information, see "Formatting the New CDS" on page 82.
- "CDS Conversion Procedures" on page 87 describes your upgrade/downgrade options. **Note that** VTCS **will not start** with an unsupported CDS format. **Also note that** you **must** use the downgrade procedures in this section. Just restoring back to or activating previous copies of the CDS can result in data loss because the locations of VTVs may have changed!

Warning: Regressing from an 'E' or 'F' level CDS can cause unpredictable results if the 4 VTV copy feature has been used. Any copies above the maximum of two allowed on a 'B' to 'D' level CDS will be dropped!

You may want to install 'E' or 'F' in a test system only, or run it in production without using an 'E' or 'F' feature until you are sure that the V6+ environment is stable.

- The compatibility PTFs allow V5/V5.1 systems to share a CDS with V6.0 systems. Note that this sharing is only possible with V6.0 systems that **do not** have `CONFIG CDSLEVEL(V6ABOVE)` applied! See Table 18 for more information.

Table 18. VTCS 6.0 Compatibility PTFs

Release	HSC	VTCS
5.0	L1H11PW for SOS500	L1H11PX for SWS500
5.1	L1H11PT for SOS510	L1H11PU for SWS510

StorageTek recommends that you apply the coexistence PTFs before converting the 6.0 CDS.

CDS Conversion Procedures

CDS format conversion consists of six general steps. Depending upon the 'from' and 'to' levels of the CDS, individual steps may be **optional** as described in the following sections. The following are the six general steps.

To convert CDS formats:

1. **Complete all tasks that apply in the following sections:**
   - "Creating or Updating the HSC LIBGEN" on page 80
   - "Verifying the LIBGEN" on page 82
   - "Formatting the New CDS" on page 82

   For more information on using `SLICREAT` to format the new CDS, see *HSC System Programmer's Guide*. Typically, you run `SLICREAT` with the level of highest level of HSC that you will run after converting the CDS format.

2. **Apply compatibility PTFs as required.**

   This step typically enables an older version of VTCS to tolerate a new CDS version in some kind of compatibility mode.

3. **Produce a CDS of an intermediate level as described in Step 4 and Step 5.**

   The CDS formats for the latest versions of VTCS are not directly compatible with all of the old levels of CDS format and vice versa, which is why you may have to go to an intermediate CDS level before reaching your final destination.

4. **Run the VTCS `DECOM` utility to list the current configuration information.**

   You can update the `DECOM` output with the desired changes for the new configuration, then use it as input in Step 5. Typically, you run `DECOM` with the highest level of the VTCS that supports the CDS 'from' level. This step is only required if you do not have an up-to-date version of the VTCS configuration deck.

5. **Run the VTCS `CONFIG` utility against the updated `DECOM` output from Step 4.**

Warning: You **must** specify `RESET` on the `CONFIG` JCL and in most cases, another parameter to specify the CDS level to be created in the ‘to’ CDS. For example, for the “F” level CDS, you must specify `CDSLEVEL(V61ABOVE)` on the `CONFIG` JCL (which means you must remove any previous `CDSLEVEL` specifications, such as `CDSLEVEL(NEW)`).

6. **Run the HSC `MERGEcds` utility to copy the real and virtual records from the ‘from’ CDS to the ‘to’ CDS.**

   For more information on using `MERGEcds`, see *VTCS Command and Utility Reference* and *HSC System Programmer’s Guide*.

Note: Some CDS conversions require the creation of a new CDS and running a `MERGECDS`. The version of the `MERGECDS` utility should always be the at the same version as the version of HSC being run. Typically the version used will be the **highest** level of HSC/VTCS that supports **BOTH** the source and target CDS (as described in Table 17. on page 84). **Note that** this may or may **not** be the version that you wish to run after the completion of the conversion!

Currently Running Version 4.0, 5, or 5.1, Upgrading to Higher Function Version 5.0 or 5.1 (Or As Intermediate Step to Upgrade to Version 6.0 or Above)

Use Table 19 if you are currently running Version 4.0, 5, or 5.1 and are upgrading to higher functionality of Version 5.0 or 5.1 (or as an intermediate step towards upgrading to Version 6.0 or above).

Table 19. Converting from Version 4.0, 5, or 5.1 up to Higher Function Version 5.0 or 5.1

From 'B'	To 'C'	To 'D'	To 'E'	To 'F'
Conversion	Not required.	Not required.	Convert to 'D' first and then follow 'D' to 'E' procedure.	Convert to 'D' first and then follow 'D' to 'F' procedure.
PTFs	Not required.	Apply V6.0 compatibility PTFs to V5/V5.1		
HSC SLICREAT utility	V5/V5.1	V5/V5.1/V6		
VTCS DECOM utility	V5/V5.1	V5/V5.1		
VTCS CONFIG utility	V5/V5.1 with RESET VERSION(NEW)	V6.0 with RESET		
HSC MERGEcds utility	V5/V5.1	V5/V5.1		

Currently Running Version 5.0 or 5.1, Upgrading to Version 6.0 (Or Downgrading to Version 4.0)

Use Table 20 if you are upgrading from Version 5.0 or 5.1 to Version 6.0 or above (or if you are downgrading to Version 4.0).

Table 20. Converting from Version 5.0 or 5.1 up to Version 6.0 or Above (Or Downgrading to Version 4.0)

From ‘C’	To ‘B’	To ‘D’	To ‘E’	To ‘F’
Conversion	Not required.	Not required.	Not required.	Not required.
PTFs	Not required.	Apply V6.0 compatibility PTFs to V5/V5.1	Not required.	Not required.
HSC SLICREAT utility	V5/V5.1	V5/V5.1/V6	Not required.	Not required.
VTCS DECOM utility	V5/V5.1	V5/V5.1	V5/V5.1	V5/V5.1
VTCS CONFIG utility	V5/V5.1 with RESET	V6.0 with RESET	V6.0 with RESET CDSLEVEL(V6ABOVE)	V6.1 with RESET CDSLEVEL(V61ABOVE)
HSC MERGEcds utility	V5/V5.1	V5/V5.1	Not required.	Not required.

Enable New Version 6.0 Features or Remove This Support to Run at Version 5.0 or 5.1 without Compatibility PTFs

Use Table 21 if you want to enable the new Version 6.0 features or remove this support to run at Version 5.0 or 5.1 without compatibility PTFs.

Table 21. Enable New Version 6.0 Features or Remove This Support to Run at Version 5.0 or 5.1 without Compatibility PTFs

From 'D'	To 'B'	To 'C'	To 'E'	To 'F'
Conversion	Not required.	Not required.	Not required.	Not required.
PTFs	Not required.	compatibility PTFs	Not required.	Not required.
HSC SLICREAT utility	V5/V5.1	V5/V5.1	Not required.	Not required.
VTCS DECOM utility	V5/V5.1/V6	V5/V5.1/V6	V5/V5.1/V6; remove VER(56) parameter from output	V5/V5.1/V6; remove VER(56) parameter from output
VTCS CONFIG utility	V5/V5.1 with RESET	V5/V5.1 with RESET VERSION(NEW)	V6/V6.1 with RESET CDSLEVEL(V6ABOVE)	V6.1 with RESET CDSLEVEL(V61ABOVE)
HSC MERGEcds utility	V5/V5.1	V5/V5.1 with compatibility PTFs applied.	Not required.	Not required.

Enable New Version 6.1 Features or Remove E or F Level Support to Run at Version 5.0 or 5.1

Use Table 22 if you want to enable the new Version 6.1 features (or remove E Level Support) or Table 23 to remove F Level support to run at Version 5.0 or 5.1.

Table 22. Enable New Version 6.1 Features or Remove E Level Support to Run at Version 5.0 or 5.1

From 'E'	To 'B'	To 'C'	To 'D'	To 'F'
Conversion	Convert to 'D' first and then follow 'D' to 'B' procedure.	Convert to 'D' first and then follow 'D' to 'C' procedure.	Not required.	Not required.
PTFs			Apply V6.0 compatibility PTFs to V5/V5.1	Not required.
HSC SLICREAT utility			V5/V5.1/V6.0	Not required.
VTCS DECOM utility			V6	V5/V5.1/V6
VTCS CONFIG utility			V6.0 with RESET	V6.1 with RESET CDSLEVEL(V61ABOVE)
HSC MERGEcds utility			V6	Not required.

Table 23. Remove F Level Support down to Run at Version 5.0 or 5.1

From 'F'	To 'B'	To 'C'	To 'D'	To 'E'
Conversion	Convert to 'D' first and then follow 'D' to 'B' procedure.	Convert to 'D' first and then follow 'D' to 'C' procedure.	Not required.	Not required.
PTFs			Apply V6.0 compatibility PTFs to V5/V5.1	Not required.
HSC SLICREAT utility			V5/V5.1	Not required.
VTCS DECOM utility			V6.1	V6.1
VTCS CONFIG utility			V6.0 with RESET	V6/V6.1 with RESET CDSLEVEL(V6ABOVE)
HSC MERGEcds utility			V6.1	Not required.

STOP

Regressing from an 'E' or 'F' level CDS can cause unpredictable results if the 4 VTV copy feature has been used. Any copies above the maximum of two allowed on a 'B' to 'D' level CDS will be dropped!

JCL Examples

Figure 10 shows a CONFIG JCL example to the CDS for conversion to Format F.

```
//CREATECFG   EXEC PGM=SWSADMIN,PARM='MIXED'
//STEPLIB     DD DSN=hlq.SLSLINK,DISP=SHR
//SLSCNTL     DD DSN=FEDB.VSMLMUL5.DBASEPRM,DISP=SHR
//SLSPRINT    DD   SYSOUT=*
//SLSIN       DD   *
 CONFIG RESET CDSLEVEL(V61ABOVE)
 .
 .
 .
 (modified CONFIG statements from DECOM
 .
 .
 .
```

Figure 10. CONFIG ***example: VER(V61ABOVE) specified to prepare to convert the CDS to Format F***

Figure 11 shows a MERGEcds JCL example where the ALL parameter specifies to copy and convert the all CDS records to VSM Extended Format.

```
//MERGECDS    EXEC PGM=SLUADMIN,PARM='MIXED'
//SLSFCNTL    DD DSN=FEDB.FROMVM4.DBASEPRM,DISP=SHR
//SLSFCTL2    DD DSN=FEDB.FROMVM4.DBASESEC,DISP=SHR
//SLSFSTBY    DD DSN=FEDB.FROMVM4.DBASESBY,DISP=SHR
//SLSPRINT    DD   SYSOUT=*
//SLSIN       DD   *
 MERGECDS ALL
 /*
```

Figure 11. MERGEcds ***example: converting the CDS to VSM Extended Format***

Note: As shown in Figure 11, in the MERGECDS JCL you specify the **old** CDS. MERGEcds senses an old format CDS and automatically converts it to VSM Extended Format. You **must use** HSC MERGEcds 5.0.0 and above for this task.

Also note that, for the MERGEcds VSM Extended Format conversion to succeed, HSC must be at Full Service level with no tape activity unless PTF L1H119X (5.0) or L1H11BE (5.1) is applied. With the PTF applied, HSC must be at Base Service Level for the MERGEcds conversion to succeed.

Updating the HSC Definition Data Sets

After you format the new CDS, update the HSC definition data for VSM as described in the following sections.

Note: For NCS 6.1, the `UNITATTR` statement has been moved from HSC to SMC and is **no longer required** for VTDs. The `UNITATTR` statement is required **only** to set the real transport model type for non-library transports (which are not supported for VSM). For more information, see *SMC Configuration and Administration Guide*.

Creating MVC VOLATTR Statements

MVCs require `VOLATTR` statements to ensure that VTCS will select the correct RTD device type for each MVC. Create `VOLATTR` statements for any new volumes you will use as MVCs. For more information, see Chapter 3, "Control Statements and Start Procedure" of *HSC System Programmers Guide*.

Creating an MVC Pool

This procedure tells how to create a pool of volumes that VSM uses as MVCs.

To create an MVC pool:

1. **Edit the data set that will contain your system's** `MVCPool` **statements.**
2. **Add** `MVCPool` **statements for the MVC ranges and save the data set.**
3. **Run the** `VT MVCDEF` **command to activate the updated data set.**

Hint: MVCs require additional definitions besides the `MVCPool` statements. For more information, see "MVC Definitions" on page 25.

Creating and Using VSM Management and Storage Classes

VTCS 4.0 and above provides VSM policy management through VSM Management and Storage Classes. See the following sections for more information:

- "Basic Procedure for Creating and Using VSM Management and Storage Classes"
- "Using Storage and Management Classes to Group Multiple Workloads on Shared MVCs"
- "Using Storage and Management Classes to Segregate Individual Workloads on Separate Sets of MVCs"
- "Using Storage and Management Classes to Specify the Source and Target MVC for VTV Consolidation"

Caution: Note the following:

- 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!

Basic Procedure for Creating and Using VSM Management and Storage Classes

The following steps show the basic procedure for creating and using VSM Management and Storage Classes.

To create and use VSM Management and Storage Classes:

1. **Determine the definition data set that contains the `STORclas` and `MGMTclas` statements.**

   As described in "HSC and SMC Definition Data Set Names" on page 14, `MGMTclas` and `STORclas` statements must reside in the same data set for cross-validation.

2. **Enable the VSM Advanced Management Feature via the HSC `FEATures` control statement.**
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 `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; for more information, see "The StorageTek DFSMS Interface" on page 122.

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.

Using Storage and Management Classes to Group 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 12 define Storage Classes `LOC`, `LOCAC`, `REM`, and `REMAC`.

```
STORCLAS NAME(LOC) ACS(00) MEDIA(ECART,ZCART,STK1R)
STORCLAS NAME(LOCAC) ACS(00) MEDIA(ECART,ZCART,STK1R)
STORCLAS NAME(REM) ACS(01) MEDIA(STK1R)
STORCLAS NAME(REMAC) ACS(01) MEDIA(STK1R)
```

Figure 12. Storage Classes for Workload Grouping

Figure 13 defines the following Management Classes:

- Management Classes `PAY` and `ACCOUNT` both specify the `LOCAC` and `REMAC` Storage Classes on the `MIGPOL` parameter. VTCS duplexes VTVs in both Management Classes to identical Storage Classes; the VTVs in `PAY` and `ACCOUNT`, therefore, are duplexed and grouped on the MVCs defined by Storage Classes `LOCAC` and `REMAC`.
- Management Class `PROD` specifies the `LOC` and `REM` Storage Classes on the `MIGPOL` parameter. The VTVs in `PROD`, therefore, are duplexed and grouped on the MVCs defined by Storage Classes `LOC` and `REM`.

```
MGMT NAME(PAY) MIGPOL(LOCAC,REMAC)
MGMT NAME(ACCOUNT) MIGPOL(LOCAC,REMAC)
MGMT NAME(PROD) MIGPOL(LOC,REM)
```

Figure 13. Management Classes for Workload Grouping

Figure 14 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 13, 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 `LOCAC` and `REMAC`.
- 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 `LOC` and `REM`.

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

Figure 14. 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.

Using Storage and Management Classes to Segregate 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 15 define Storage Classes LOC, CUSTA, and CUSTB.

```
STORCLAS NAME(LOC) ACS(00) MEDIA(ECART,ZCART,STK1R)
STORCLAS NAME(CUSTA) ACS(00) MEDIA(ECART,ZCART,STK1R)
STORCLAS NAME(CUSTB) ACS(00) MEDIA(ECART,ZCART,STK1R)
```

Figure 15. Storage Classes for Workload Segregation

Figure 16 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.
- Similarly, VTCS simplexes VTVs in Management Class CUSTB to only the CUSTB Storage Class, and VTVs in Management Class PROD to only the LOC Storage Class.

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

Figure 16. Management Classes for Workload Segregation

Figure 17 consists of TAPEREQ statements that do the following:

- As shown in Figure 17, 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.** data sets is effectively segregated on a separate set of MVCs from the workload from the CUSTB.** data sets.

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

Figure 17. 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 Storage and Management Classes to Specify the Source and Target MVC for VTV Consolidation

For duplexed VTVs, you can use Storage and Management Classes to specify the source MVC for VTV consolidation. You can also use Storage Classes to specify the target MVC for consolidation. For example, the STORclas statements in Figure 18 define Storage Classes LOCFAST, LOC9490, and REMOTE.

```
STORCLAS NAME(LOCFAST) ACS(00) MEDIA(ECART,ZCART,STK1R)
STORCLAS NAME(LOC9490) ACS(00) MEDIA(ECART)
STORCLAS NAME(REMOTE) ACS(01) MEDIA(STK1R)
```

Figure 18. Storage Classes for Source and Target MVC for Consolidation

Figure 19 defines the Management Class PROD that specifies:

- The LOCFAST and REMOTE Storage Classes on the MIGPOL parameter. VTCS duplexes VTVs in this Management Class to both Storage Classes.
- The LOCFAST Storage Class on the CONSRC parameter. For consolidations that specify Management Class PROD, VTCS consolidates VTV copies from MVCs in the LOCFAST Storage Class.
- The LOC9490 Storage Class on the CONTGT parameter. For consolidations that specify Management Class PROD, VTCS consolidates VTVs to MVCs in the LOC9490 Storage Class.

```
MGMT NAME(PROD) MIG(LOCFAST,REMOTE) CONSRC(LOCFAST) CONTGT (LOC9490)
```

Figure 19. Management Class for Source and Target MVC for Consolidation

Figure 20 shows a TAPEREQ statement that specifies Management Class PROD for all data sets with a qualifier of PROD.** to implement the migration and consolidation policies shown in Figure 19.

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

Figure 20. TAPEREQ Statement for Source and Target MVC for Consolidation

Figure 21 shows JCL for the CONSOLID utility that specifies Management Class PROD. For all data sets with a qualifier of PROD.**, VTCS selects VTV copies from MVCs in Storage Class LOCFAST for the consolidation and consolidates the selected VTVs to MVCs in Storage Class LOC9490.

```
CONS MGMT(PROD)
```

Figure 21. CONSOLID Utility JCL Specifying Source and Target MVC for Consolidation by Management Class

Creating and Using Named MVC Pools

Caution: You can use Named MVC Pools to give an application ownership of the MVCs in the named pool. For example, a service bureau might elect to use Named MVC Pools where its customers have legal requirements to buy and own a group of MVCs. 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:

- "Basic Procedure for Creating and Using VSM Management and Storage Classes" on page 97
- "Using Storage and Management Classes to Group Multiple Workloads on Shared MVCs" on page 98
- "Using Storage and Management Classes to Segregate Individual Workloads on Separate Sets of MVCs" on page 99
- "Using Storage and Management Classes to Specify the Source and Target MVC for VTV Consolidation" on page 100

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, which includes setting the policies described in "MVC Policies" on page 59 for the entire 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, use the procedure on page 102 to implement them.

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

1. **Edit the data set that contains your system's `MVCPool` statements.**
2. **Modify existing `MVCPool` statements and/or add additional statements to define the Named MVC Pools and save the data set.**

   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.

3. **Run the `VT MVCDEF` command to activate the updated data set.**
4. **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 5.

5. **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)
   ```

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

   For more information, see "Basic Procedure for Creating and Using VSM Management and Storage Classes" on page 97.

7. **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 "The StorageTek DFSMS Interface" on page 122.

☞ **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.

Updating the HSC PARMLIB Member (SLSSYSxx)

You can specify the `VT MVCDEF` command as a statement in the HSC PARMLIB. Figure 22 shows an example of `TREQDEF`, `VT MVCDEF`, and `MGMTDEF` commands specified as statements in the HSC PARMLIB member.

```
TREQDEF DSN(SMC.TAPEREQ)
VT MVCDEF DSN(VSM.MVCPOOL)
MGMTDEF DSN(HSC.PARMS)
COMMP METH LMU
FEAT VSM(ADVMGMT)
```

Figure 22. ***Example: Updating the HSC PARMLIB Member for VSM***

In Figure 22:

`SMC.TAPEREQ`
: is the data set that contains your system's `TAPEREQ` statements (including `TAPEREQ` statements for VTVs).

Note: For NCS 6.1 the `TAPEREQ` statement (and the accompanying `TREQDEF` command) has been moved from HSC and MVS/CSC to SMC. For more information, see *SMC Configuration and Administration Guide*.

`VSM.MVCPOOL`
: is the data set that contains your system's `MVCPool` statements.

`HSC.PARMS`
: is the data set that contains your system's `MGMTclas` and `STORclas` statements.

`COMMP METH LMU`
: specifies that `LMU` is the communications method. StorageTek recommends that you specify either `LMU` or `VTAM`, not `CDS` to allow even sharing of resources in a multi-host environment.

`FEAT VSM(ADVMGMT)`
: enables the Advanced Management Feature.

Adding SMF Parameters for VTCS to SYS1.PARMLIB

HSC can produce SMF record subtypes for VTCS events. To produce these record subtypes, you must add two statements to your SMF parameters in `SYS1.PARMLIB` member `SMFPRMxx` to specify the following:

- HSC subsystem for which records are produced
- Recording interval in seconds
- SMF record subtypes. The record subtypes must be specified as a list (*subtype1*, *subtype2*,...*subtypen*), as a range (*subtype1*-*subtypen*), or as a combination (*subtype1*, *subtype2*-*subtypen*). A range must be specified using a dash; a colon is invalid for a range.

Hint: If you use ExPR for VSM reporting, StorageTek recommends that you specify that your system produces the HSC SMF record subtypes 1 through 8 and 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 25, 26, 27, 28, and 29 as shown in Figure 23.

Figure 23 shows example statements that produce record subtypes 1 through 8 and 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 25, 26, 27, 28, and 29 at 1500 second intervals for HSC subsystem `SLS0`.

```
SUBSYS(SLS0,INTERVAL(001500),TYPE(255))
SUBPARM(SLS0(SUBTYPE,
(1-8, 10, 11, 13, 14, 15, 16, 17, 18, 19, 20, 21, 25, 26, 27, 28, 29)))
```

Figure 23. SYS1.PARMLIB ***member*** SMFPRMxx ***example for VTCS SMF records***

Connecting MVS/CSC Clients to VSM

The following procedure tells how to update LibraryStation and MVS/CSC to connect MVS/CSC clients to VSM.

Note: As an alternative to this configuration, you can simply install SMC 6.0 in your client MVS system, and SMC will route virtual allocation and mount requests to HSC running in a remote server HSC system. For more information, see the *SMC 6.0 Configuration and Administration Guide*.

To connect MVS/CSC clients to VSM:

1. **Define a virtual ACS using the LibraryStation `VIRTACS` statement.**

   For example, to define virtual ACS 126 and map it to VTSS VTSS02, create the following `VIRTACS` statement:

   ```
   VIRTACS ID(126) VTSSNAME(VTSS02)
   ```

   This concludes this procedure if *both* of following are true:

   - Clients use the same MVS device numbers defined for VSM as described in "VTD Unit Addresses" on page 16.
   - Your configuration has no cartridge tape UCB defined with an MVS device number that matches a device number defined for VSM.

   Otherwise, continue with Step 2 to map the MVS device numbers defined for VSM to client device numbers.

2. **For each VTSS to which MVS/CSC clients connect, define VTD device addresses using the MVS/CSC `LIBUNIT` statement.**

   Caution: Each device in the `LIBUNIT` statement must be represented by a UCB and may not be used to access any device other than the associated VTD. These device addresses do not, however, have to be online.

   For more information about the `LIBUNIT` statement, see Chapter 3, "Defining MVS/CSC Startup Parameters" of *MVS/CSC Configuration Guide*.

   For example, for a single VTSS (VSM2 or VSM3), create a `LIBUNIT` statement such as the following:

   ```
   LIBUNIT (B00,B01,B02,B03,B04,B05,B06,B07,  -
            B08,B09,B0A,B0B,B0C,B0D,B0E,B0F,  -
                         .
                         .
            B38,B39,B3A,B3B,B3C,B3D,B3E,B3F)
   ```

3. **For each VTSS to which MVS/CSC clients connect, map the VTD device addresses from Step 2 on page 106 to the VTD virtual ACS locations using the MVS/CSC `UNITMAP` statement.**

   For more information about the `UNITMAP` statement, see Chapter 3, "Defining MVS/CSC Startup Parameters" of *MVS/CSC Configuration Guide*.

   See Table 12. on page 31 and Table 13. on page 34 for information on VTD locations in virtual ACSs.

   For example, for the device addresses you defined in Step 2, create a `UNITMAP` statement such as the following:

```
UNITMAP (B00,7E:00:1:0,B01,7E:00:1:1,  -
        B02,7E:00:1:2,B03,7E:00:1:3, -
                     .
         B3E,7E:03:4:2,B3F,7E:03:4:3)
```

   Note: The `VIRTACS` statement specifies virtual ACS IDs in decimal, but the `UNITMAP` statement specifies these IDS in hexadecimal.

   This concludes this procedure if you use the HSC common subpool. Otherwise, continue with Step 4.

4. **Define an HSC subpool that contains VTVs.**

   For more information, see *HSC System Programmer's Guide for MVS.*

5. **Define a LibraryStation subpool that corresponds to the HSC subpool in Step 4 using the LibraryStation `SPNUM` statement.**

   For example, create the following `SPNUM` statement to define VTV subpool 7 that corresponds to HSC subpool `LSVIRT1`.

```
SPNUM NUM(07) SPNAME(LSVIRT1) VIRT(YES)
```

Connecting Non-MVS/CSC Clients to VSM

The following procedure tells how to connect non-MVS/CSC 4.0 and above clients to VSM and define LibraryStation subpools that contain VTVs. Contact StorageTek Software Support for information on the supported clients.

To connect non-MVS/CSC clients to VSM:

1. **Define a virtual ACS using the LibraryStation `VIRTACS` statement.**

   For example, to define virtual ACSs 126 and 125 and map them to VTSSs VTSS01 and VTSS02, create the following `VIRTACS` statements:

   ```
   VIRTACS ID(126) VTSSNAME(VTSS01)
   VIRTACS ID(125) VTSSNAME(VTSS02)
   ```

2. **Define an HSC subpool that contains VTVs.**

   Note: If you define this subpool as ANSI label, VTCS will mount scratch VTV as ANSI labeled tapes from this subpool.

   For more information, see *HSC System Programmer's Guide for MVS.*

3. **Define a LibraryStation subpool that corresponds to the HSC subpool in Step 2 using the LibraryStation `SPNUM` statement.**

   For example, create the following `SPNUM` statement to:

   - Define VTV subpool 7 that corresponds to HSC subpool `LSVIRT1`
   - Pass Management Class `MGMTCLS7` to VSM when a VTV is mounted
   - Specify VTSSs `VTSS01` and `VTSS02` are used to satisfy VTV mounts
   - Restrict VTV mount requests to the client at IP address 129.80.57.16

   ```
   SPNUM NUM(07) SPNAME(LSVIRT1) VIRT(YES) MGMT(MGMTCLS7)
   VTSSL(VTSS01,VTSS02) IPADDR(129.80.57.16)
   ```

Restarting NCS

To complete the NCS reconfiguration, start one or more of the following on all hosts that are using the new CDS data sets:

- HSC; for more information, see Chapter 7, “Initializing the HSC” of *HSC Configuration Guide.* If you modified the HSC startup procedure as described in “Modifying the HSC Startup Procedure to include the VTCS 6.1.0 LINKLIB” on page 78, HSC initialization automatically starts VTCS.
- LibraryStation; for more information, see Chapter 4, “Starting and Stopping LibraryStation” of *LibraryStation Operator and System Programmer’s Guide*.
- MVS/CSC; for more information, see Chapter 2, “Operating the MVS/CSC” of *MVS/CSC Operator’s Guide*.

Chapter 5. Configuring VSM

After you reconfigure HSC, you must configure VSM, which includes the tasks described in the following sections:

- “Defining VSM Security”
- “Configuring VTCS” on page 114
- “Running the HSC MERGECDS Utility” on page 120
- “Updating the Tape Management System” on page 120
- “Updating HSM” on page 121
- “Routing Data Sets to VSM” on page 121
- “Starting VTCS” on page 124

Hint: Several tasks in this Chapter require you to specify VSM system values that you determined on page 14 and recorded in Table 25. on page 125.

Defining VSM Security

The following sections tell how to define security for VSM:

- "Defining MVC Pool Volser Authority"
- "Defining VTCS Command Authority" on page 113

Defining MVC Pool Volser Authority

When VSM needs to mount an MVC and to write to an MVC, a SAF query is issued to verify that the HSC user (see "Defining A Security System User ID for HSC, SMC, and VTCS" on page 66) has UPDATE authority for the MVC. The SAF query is issued on behalf of HSC and passed to the system security product (such as RACF, CA-ACF2, or CA-Top Secret).

VSM requires UPDATE authority for the volsers in the MVC pool. All other users should have an access of NONE for these volsers. Similarly, VSM should not have UPDATE authority for any volsers that are not in the MVC pool. See the documentation for your security product for procedures to add the appropriate TAPEVOL security for VSM. Table 24 summarizes these definitions.

Table 24. Security Class, Resource Class, and Access Values for MVC Pool Volser Authority

Class	Resource Name	Recommended User Access Levels
TAPEVOL	MVC Pool Volume Serials	UPDATE - allows VSM to write on MVC

Figure 24 shows an example of a RACF profile and permissions commands to give the user ID `VSM8HSC` update access to MVC volser CVC024.

```
*******************************************************************************************
* Define a profile in the TAPEVOL class for MVC CVC024                                    *
*******************************************************************************************
RDEFINE TAPEVOL CVC024 UACC(NONE)
*******************************************************************************************
** Allow user ID VSM8HSC update access to MVC CVC024                                      *
*******************************************************************************************
PERMIT CVC024 CLASS(TAPEVOL) ACCESS(UPDATE) ID(VSM8HSC)
*******************************************************************************************
```

Figure 24. Example RACF MVC volser access file

Caution: Note the following:

- To ensure that MVCs are not accidentally overwritten, for each MVC volser, you must update your TAPEVOL security as described above and your tape management system. For more information, see “Updating the Tape Management System” on page 120.
- You must also run the HSC `UNSCratch` Utility to unscratch any current scratch cartridges in the MVC range. For more information, see *HSC System Programmer’s Guide for MVS*.
- Depending on the default settings of your security system, VSM may not be able to mount and to write to MVCs until you have defined a security system user ID for HSC and TAPEVOL profiles for the MVCs.
- If you add new ranges of MVCs to your VSM system, remember to update the TAPEVOL profiles to include the new ranges.

Defining VTCS Command Authority

If HSC user exit SLSUX15 sets a return code of UX15CHKA, the exit issues a command authorization request to the system security product. Table 25 shows an example of RACF profile and permissions commands to give user SAM15 access to all VTCS commands (those with a VT command prefix). Note that you can only give a user access to *all* VTCS commands; you cannot give access to individual VTCS commands. For more information, see *HSC System Programmer’s Guide for MVS*.

```
*******************************************************************************************
* Define a profile in the OPERCMDS class for all VTCS commands                            *
*******************************************************************************************
RDEFINE OPERCMDS subsysname.VT UACC (NONE)
*******************************************************************************************
** Allow user SAM15 update access to all VTCS commands                                     *
*******************************************************************************************
PERMIT subsysname.VT CLASS(OPERCMDS) ID(SAM15) ACCESS (UPDATE)
*******************************************************************************************
```

Figure 25. ***Example RACF VTCS command authorization file***

Configuring VTCS

You run VTCS `CONFIG` to define the VSM resources and operational parameters to VTCS. For more information about `CONFIG`, see *VTCS Command and Utility Reference*.

Caution: You **must** run the VTCS `CONFIG` utility **before** you run the `MERGEcds` utility to ensure that the VTCS information is added to the CDS.

Storing VTCS Locks in a Coupling Facility (Optional)

Before VTCS updates a CDS record (for example, a VTV record) it locks the record to avoid contention from concurrent updates from multiple hosts. VTCS releases the CDS lock record once the CDS record has been updated.

Customers with large VTCS configurations experience high CDS I/O rates, part of which is due to the need to access CDS lock records. As configurations grow in size, (for example, by adding more hosts), CDS performance becomes a bottleneck.

An MVS Coupling Facility is a suitable alternative medium for VTCS Lock data because:

- A Coupling Facility provides very fast data transfer speeds, so the new I/O to the Coupling Facility with Lock records is less than the corresponding I/O to a CDS with Lock records.
- MVS provides a technique that allows data stored in a failing Coupling Facility to be re-built in another Coupling Facility (if one exists) without terminating the application.

Note: If VTCS locks are held in a coupling facility structure, VTCS uses the Structure (rather than the HSC mechanism) for sending/receiving Host-to-Host messages.

When to implement VTCS locks in a Coupling Facility Structure

Storing VTCS lock data in a Coupling Facility Structure is a solution to the **specific** problem of VTCS causing high I/O demand to the CDS in some configurations.

Note: Using a Coupling Facility Structure is **not** a solution to all CDS performance issues. For this reason, StorageTek recommends that if you believe you have a CDS performance problem, contact StorageTek Software support to have the problem analyzed **before** considering implementing VTCS lock data in a Coupling Facility.

Requirements

To store VTCS Locks in a Coupling Facility:

- All hosts must have access to the same Coupling Facility. Similarly, if you have an alternate Coupling Facility to rebuild the VTCS Lock Structure, all hosts must have access to that alternate Coupling Facility. All hosts must also be in a Sysplex.
- The Coupling Facility Structure must be predefined to MVS before VTCS can use it to store CDS Lock Records. VTCS uses the list form of a Coupling Facility Structure. `Display LOCKS` shows one of the following VTCS Coupling Facility lock types:

  Host Footprint
  used to serialize access to the host footprint list.

  Host to Host
  used to serialize access to a given host to host list.

  Lock data
  used to serialize access to the VTCS lock data.

  Formatting
  used to serialize the initial formatting of the structure; also used when rebuilding data.

  System
  lock is held, but is not a lock used by VTCS; assume it is used by MVS.

Sizing the Coupling Facility Structure

A Structure size of 768K should be sufficient for configurations up to 100 VTSSs.

If the Structure is sized too small, HSC/VTCS will be unable to connect to the Structure or will be able to connect but will be unable to format all of its data. In both cases, VTCS will terminate.

Defining the Coupling Facility Structure to MVS. Figure 26 on page 118 shows an example of an `IXCMIAPU` job to define a VTCS Lock Structure within a Coupling Facility Resource Manager (CRFM). In this example, note that:

- There are two Coupling Facilities, `FACIL01` and `FACIL02`.
- There is a 768K Structure called `STK_VTCS_LOCKS` to store VTCS Lock Records.
- Structure `STK_VTCS_LOCKS` can exist in either Coupling Facility, but `FACIL01` is preferred over `FACIL02`. If VTCS starts to store lock data in `FACIL01` and `FACIL01` then becomes unavailable, VTCS attempts to build the `STK_VTCS_LOCKS` in `FACIL02` to ensure continuous operations.

Note: If you define only one Coupling Facility, and it becomes unavailable, VTCS terminates on all hosts but HSC will still be running. If this occurs, do one of the following:

- Fix the Coupling Facility error, then recycle HSC/VTCS on all hosts; you can resume without changing the configuration.
- Stop HSC on all hosts, run `CONFIG RESET` *without* the `LOCKSTR` parameter so VTCS can store VTCS Locks in the CDS, then restart HSC/VTCS on all hosts. For more information, see *VTCS Command and Utility Reference*.

Managing Failures/Unavailability of the VTCS lock structure

VTCS supports Structure Rebuild to allow for failures/unavailability of the Structure or the Coupling Facility containing the Structure.

Structure rebuild can be initiated by:

- Operator command (`SETXCF START,REBUILD,`*`xxx`*) for a planned outage of the Structure or the Coupling Facility, and/or
- MVS or VTCS detecting an error in, or failure of, the Structure or the Coupling Facility.

Note that VTCS does **not** support System Managed Duplexing.

If the Structure used by VTCS can only be allocated in one Coupling Facility, VTCS will terminate on all Hosts if the Structure (or the Coupling Facility containing the Structure) fails or becomes unavailable.

If the Structure can be allocated in more than one Coupling Facility, VTCS's Structure Rebuild code will attempt to rebuild the data in an alternate Coupling Facility Structure. VTCS will only terminate if the rebuild fails.

```
//SYSPRINT DD SYSOUT=*
//SYSIN        DD *
  DATA TYPE(CFRM) REPORT(YES)
  DEFINE POLICY NAME(POLICY1) REPLACE(YES)
             CF   NAME(FACIL01)
                    TYPE(123456)
                   MFG(IBM)
                   PLANT(02)
                   SEQUENCE(123456789012)
                   PARTITION(1)
                   CPCID(00)
                   SIDE(0)
                    DUMPSPACE(2000)
           CF   NAME(FACIL02)
                   TYPE(123456)
                   MFG(IBM)
                   PLANT(02)
                   SEQUENCE(123456789012)
                   PARTITION(2)
                   CPCID(00)
                   SIDE(1)
                   DUMPSPACE(2000)
         STRUCTURE   NAME(STK_VTCS_LOCKS)
                   SIZE(768)
                    PREFLIST(FACIL01,FACIL02)
```

Figure 26. ***Example IXCMIAPU Job to Define a Coupling Facility Structure***

Defining the Coupling Facility Structure to MVS

Figure 27 shows an example of a CONFIG job to define a VTCS Lock Structure within a Coupling Facility to VTCS. In this example, note that:

- The Structure is predefined as shown in Figure 26 on page 118.
- RESET is specified. You must specify RESET (all hosts must be down) to implement or remove a VTCS Lock Structure within a Coupling Facility.

```
//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
//SLSPRINT     DD   SYSOUT=*
//SLSIN        DD   *
 CONFIG RESET
 GLOBAL   MAXVTV=32000   MVCFREE=40 LOCKSTR=STK_VTCS_LOCKS
.
.
.
```

Figure 27. CONFIG ***example: defining a Coupling Facility Structure to VTCS***

Hint: If a Lock Structure is already defined to VTCS, you can use DECOM, Display CONFIG, and Display LOCKS to display information about the Lock Structure.

Running the HSC MERGECDS Utility

If you are converting to VSM Extended Format, after you run the VTCS `CONFIG` utility, run the HSC `MERGEcds` Utility to transfer volume information from the old CDS to the new CDS.

Updating the Tape Management System

To update your tape management system (such as CA-1, CA-Dynam/TLMS, and DFSMSrmm), do the following:

- Add volser ranges for VTVs to your tape management system. Ensure that you do *not* assign vault codes to VTVs.
- Access to the MVCs via an RTD bypasses the MVS intercepts put in place by the tape management system so that it does *not* record within its database any access to the MVCs by VSM and does *not* automatically provide protection against inadvertent overwrites of non-expired data on MVCs. Therefore, if you choose to define MVCs to the tape management system, StorageTek **strongly recommends** that you define them as non-scratch, non-expiring volumes.
- The tape management system requires an entry in the MVS Subsystem Name Table; this entry must precede the entry for HSC. For more information, see Chapter 3 of the *HSC Configuration Guide.*

Note: If you are using AutoMedia for MVS, ensure that VTVs are defined as virtual volumes to direct AutoMedia to bypass DSN checking, which allows AutoMedia to recall, mount, and reuse non-resident scratch VTVs.

Caution: Note the following:

- VTCS has an automatic interface to notify RMM when a VTV becomes scratch, but RMM does not notify VTCS when you unscratch a VTV by changing the CV status to non-scratch. If this is done, you must also run the HSC SLUADMIN utility to unscratch the VTV for VTCS. Otherwise, you may encounter a mount failure when VTCS attempts to select the VTV to service a scratch mount request.
- RMM (DFSMS/RMM) has additional integrity checks at mount time to ensure that the correct volume has been mounted. Because VTCS has features and optimizations that sometimes present a new initialized version of a VTV rather than the current copy of a VTV, it is necessary for VTCS to override these RMM integrity checks. VTCS does these overrides via the LISTVOLUME and CHANGEVOLUME API calls to update the RMM database. You must therefore ensure that HSC has been given the appropriate security access to the RMM API. For more information, see your RMM documentation.

Updating HSM

HSM users that have mixed devices that were "logically" defined as the same type of device, such as 3490E, but are "physically" different, such as T9940, virtual (VTD), or 9490 must set the following parameter in HSM:

```
SETSYS RECYCLEINPUTDEALLOCFREQUENCY(MIGRATION(1))
```

By setting this parameter, when HSM is "recycling", it will deallocate the input drive after it processes each input tape. This is required where the tapes being recycled are "physically" mixed as described above.

If you do not set this parameter, it is possible that you could allocate a 9490 transport for the first tape, then if the second tape was virtual (VTV) or STK2P, the job would fail due to media incompatibility. That is, you could not physically mount the second tape (virtual or STK2P media) on the 9490 drive that had been allocated for the first tape.

Routing Data Sets to VSM

You recorded your VSM candidate data sets in Table 25. on page 125. To route these data sets to VSM, use any of the techniques described in the following sections:

- The StorageTek DFSMS interface; see page 122.
- SMC TAPEREQ statements; see page 123.

Note: For NCS 6.1 the TAPEREQ statement (and the accompanying TREQDEF command) has been moved from HSC and MVS/CSC to SMC. For more information, see *SMC Configuration and Administration Guide*.

Note: In addition, you can also change your JCL to direct data sets to VSM although StorageTek does not recommend this method.

Caution: StorageTek strongly recommends that you create VTVs as Standard Label (SL) tapes, otherwise unpredictable results can occur.
Also note that VSM does not provide readonly protection for VTVs. That is, even if MVS requests a mount READONLY of a VTV, VSM mounts the VTV as READ/WRITE.

The StorageTek DFSMS Interface

You can use the StorageTek DFSMS interface to route data sets to VSM via Unit Name substitution. These interfaces use names that you code in SMS routines to drive SMC allocation processing. For more information about the StorageTek DFSMS interface, see *SMC Configuration and Administration Guide*.

Use this interface for VSM as follows:

1. Define a Storage Group and Storage Class for data sets to be routed to VSM. The Storage Group must be identical in name to an esoteric that represents VTDs. Use an esoteric name and the data set selections that you recorded in Table 25. on page 125.

   For more information on defining and using VSM esoterics for the StorageTek DFSMS interface, see "VSM Esoterics and Esoteric Substitution for the StorageTek DFSMS Interface" on page 18.

2. Code Storage Class and Storage Group ACS routines to assign the correct Storage Class and Storage Groups to virtual tape data when &ACSENVIR = "STKTAP1". For more information, see *HSC System Programmer's Guide for MVS*.

3. If you want to pass one or more Management Classes to VTCS, create a Management Class that is identical in name to a Management Class you defined on a `MGMTclas` statement. Then add code to the Management Class ACS routine to assign a VTCS Management Class to selected data sets when &ACSENVIR = "STKTAP1".

   Note: The Management Class you define on the `MGMTclas` statement can specify a VSM Storage Class, which is **not** the same as the ACS Storage Class you define in Step 2.

4. Set the SMC `ALLOCDEF` or `ALLOCJOB` command `SMS` option to `ON` so the StorageTek DFSMS interface drives SMS ACS routines.

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

For more information about the StorageTek DFSMS interface, see *SMC Configuration and Administration Guide*.

SMC TAPEREQ Statements

To route data sets to VSM, you can create an SMC TAPEREQ statement. To route data sets to VSM with TAPEREQ statements, do one of the following:

- Specify Virtual on the MEDia, MODel, or RECtech parameter. If you specify Virtual, VSM selects an available VTD in your system and routes the job to that VTD.

  In a multi-VTSS environment, therefore, specifying Virtual does *not* direct the VTD allocation to a specific VTSS, but lets the allocation occur in any VTSS in the configuration.

- Specify an esoteric that represents VTDs on the ESOTeric parameter. You recorded your VSM esoterics in Table 25. on page 125.

  For VSM, esoteric definition and substitution is different in JES2 and JES3. For more information on defining and using VSM esoterics for TAPEREQ statements, see “VSM Esoterics and Esoteric Substitution for SMC TAPEREQ Statements and NCS User Exits” on page 19.

- Specify a scratch subpool that contains virtual volumes.

Caution: Multiple TAPEREQ statements that specify the same or overlapping selection criteria (such as jobname, stepname, or data set) can cause undesirable results (such as assignment of MEDia Virtual *and* an esoteric).

HSC User Exits

To route data sets to VSM with HSC User Exits, do one of the following:

- Use return code UX02VIRT (32) in register 15 in HSC User Exit SLSUX02 (JES2) or for SLSUX04 (JES3) use UX04VIRT (24), which you use to control transport allocation for scratch mounts. To satisfy a scratch mount request, return code UX0xVIRT causes VSM to select an available VTD in your system and routes the job to a VTV mounted on that VTD.

- Use esoteric substitution in any of the User Exits that support esoteric substitution. For example, to direct scratch allocation requests to a VTD, specify an esoteric that represents VTDs in the UX02ESO field of SLSUX02 or the UX04ESOT field of SLSUX04.

  For VSM, esoteric definition and substitution is different in JES2 and JES3. For more information on defining and using VSM esoterics for HSC User Exits, see “VSM Esoterics and Esoteric Substitution for SMC TAPEREQ Statements and NCS User Exits” on page 19.

For more information about HSC User Exits, see *HSC System Programmer’s Guide for MVS*.

MVS/CSC User Exits

To route data sets to VSM with MVS/CSC User Exits, do one of the following:

- MVS/CSC User Exit SCSUX02 (JES2 and JES3 without TAPE setup environments), which you use to control transport allocation for scratch mounts, now supports return code UX02VIRT in register 15. SCSUX04 (JES3 with TAPE setup environment) also supports return code UX04VIRT in register 15. To satisfy a scratch mount request, these return codes cause

VSM to select an available VTD in your system and route the data set to a VTV mounted on that VTD.

In a multi-VTSS environment, therefore, these return codes do *not* direct the VTD allocation to a specific VTSS, but let the allocation occur in any VTSS in the configuration.

- Use esoteric substitution in any of the User Exits that support esoteric substitution. For example, to direct scratch allocation requests to a VTD, specify an esoteric that represents VTDs in the UX02ESO field of SCSUX02 or the UX04ESOT field of SCSUX04.

  For VSM, esoteric definition and substitution is different in JES2 and JES3. For more information on defining and using VSM esoterics for MVS/CSC User Exits, see "VSM Esoterics and Esoteric Substitution for SMC TAPEREQ Statements and NCS User Exits" on page 19.

For more information about MVS/CSC User Exits, see Chapter 8 or Chapter 9 in *MVS/CSC System Programmer's Guide.*

Starting VTCS

Ensure that you modified the HSC startup procedure as described in "Modifying the HSC Startup Procedure to include the VTCS 6.1.0 LINKLIB" on page 78. HSC initialization automatically starts VTCS, and HSC termination automatically terminates VTCS.

Appendix A. VSM Configuration Record

Table 25 lists the installation and configuration values you determined on page 3. It also provides a record of your site's VSM configuration, which can help you and StorageTek service troubleshoot problems with your VSM system.

Table 25. VSM Configuration Record

Configuration Value	Planning Information	Your Site's Selection
`MVCPool` data set name	"HSC and SMC Definition Data Set Names" on page 14	
`MGMTclas/STORclas data set name`	"HSC and SMC Definition Data Set Names" on page 14	
`VOLATTR` data set name	"HSC and SMC Definition Data Set Names" on page 14	
`TAPEREQ` data set name	"HSC and SMC Definition Data Set Names" on page 14	
VTSS identifiers	"VTSS Identifiers" on page 15	
VTD unit addresses	"VTD Unit Addresses" on page 16	
VSM esoteric names	"VSM Esoterics and Esoteric Substitution" on page 17	
VTV volsers (all)	"VTV Definitions" on page 22	
VTV volsers (scratch pool ranges)	"VTV Definitions" on page 22	
RTD unit addresses	"RTD Unit Addresses and Identifiers" on page 24	
MVC volsers - `VOLATTR` statements and `CONFIG`	"MVC Definitions" on page 25	
MVC volsers - `MVCPool` statements	"MVC Definitions" on page 25	
HSC CDS DASD size	"HSC CDS DASD Space" on page 43	
Tape management system DASD size	"Tape Management System DASD Space" on page 43	
VSM candidate data sets		
	VTSS Policies	

Maximum and minimum concurrent automatic migration, immediate migration, and migrate-to-threshold tasks (`CONFIG MAXMIG/MINMIG`)	"Maximum and Minimum Concurrent Automatic Migration, Immediate Migration, and Migrate-to-Threshold Tasks" on page 47	
AMT settings	"AMT Settings" on page 48	
Deleting scratched VTVs setting	"Deleting Scratched VTVs" on page 50	
VTSS Preferencing	"VTSS Preferencing" on page 51	
	VTV Migration and Consolidation Policies	
Hosts disabled from migration, consolidation, and export by VTV or Management Class (`CONFIG NOMIGRAT`)	"Hosts Disabled from Migration, Consolidation and Export by VTV or Management Class" on page 52	
VTV residency interval	"VTV Residency Interval before Automatic Migration Candidacy" on page 53	
Migrate immediate setting (`MGMTclas IMMEDMig`)	"Immediately Migrate VTVs On Dismount" on page 54	
MVC retain interval	"MVC Retain Interval" on page 55	
VTVs per MVC (`CONFIG MAXMVC`)	"Maximum VTVs per MVC" on page 55	
ACS and media for migration MVCs	"ACS and Media Type of MVCs for Migration and Reclamation." on page 55	
Number of Migrated VTV copies	"Number of Migration Copies" on page 56	
Migrate duplexed VTVs to separate ACSs (`MGMTclas ACSlist`)	"Migrate Duplexed VTVs to Separate ACSs" on page 56	
ACS and media for output MVCs for consolidation	"Output MVC ACS and Media for VTV Consolidation" on page 56	
ACS and media for source MVCs for consolidation	"Source MVC ACS and Media for Consolidation of Migrated Duplexed VTVs" on page 57	

	MVC Space Reclamation Policies	
Hosts disabled from reclamation (`CONFIG NORECLAM`)	"Hosts Disabled for Reclamation" on page 59	
Free MVCs threshold (`CONFIG MVCFREE`)	"Free MVCs Threshold - Starts Automatic Space Reclamation" on page 60	
MVC fragmented space threshold (`CONFIG THRESHld`)	"MVC Fragmented Space Threshold- Determines MVC Eligibility for Reclamation" on page 60	
Eligible/Total MVCs threshold (`CONFIG START`)	"Eligible/Total MVCs Threshold - Starts Automatic Space Reclamation" on page 61	
MVCs processed per reclaim (`CONFIG MAXMVC`)	"Maximum MVCs Processed Per Reclaim" on page 62	
	Storage Class Policies	
Storage Class Policies	"Storage Class Preferencing" on page 63	

Appendix B. VSM Connectivity Requirements

StorageTek **strongly recommends** full VSM connectivity as explained in this appendix. This appendix also provides examples of the types of processing problems encountered with partial connectivity. "Full VSM connectivity" basically constitutes an all-to-all logical path connection of LPARs to VTSSs that allows VTCS to manage all possible combinations of VTV access and transfer, integrity checking, and so forth. Figure 28 shows a Full Connectivity configuration.

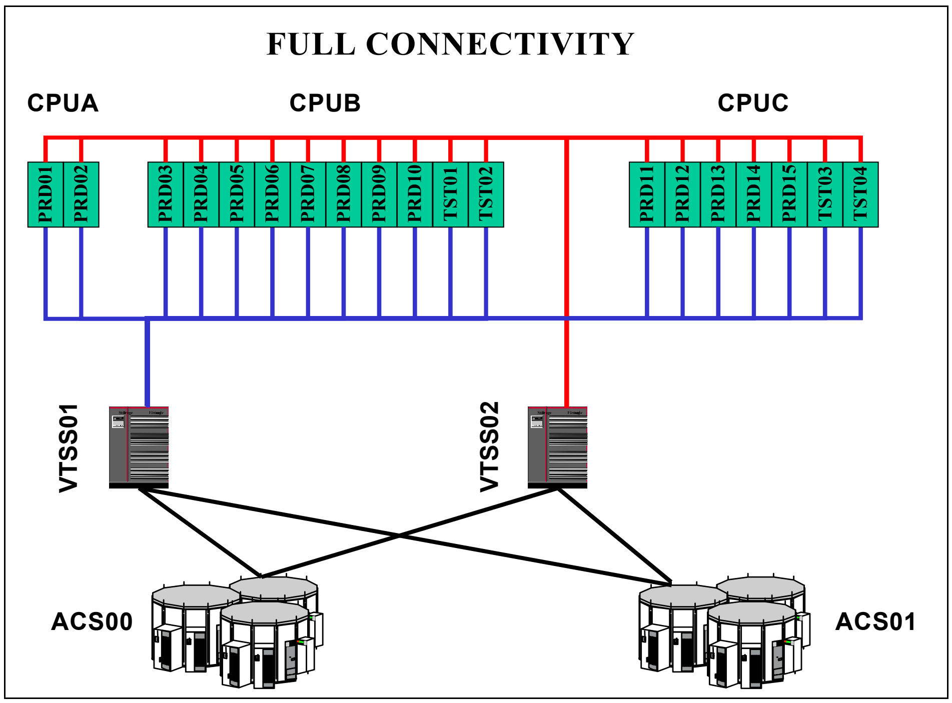


Figure 28. ***Full VSM Connectivity***

Partial Connectivity consists of connections to only those LPARs where throughput or attachment to VTDs are issues, as shown in Figure 29.

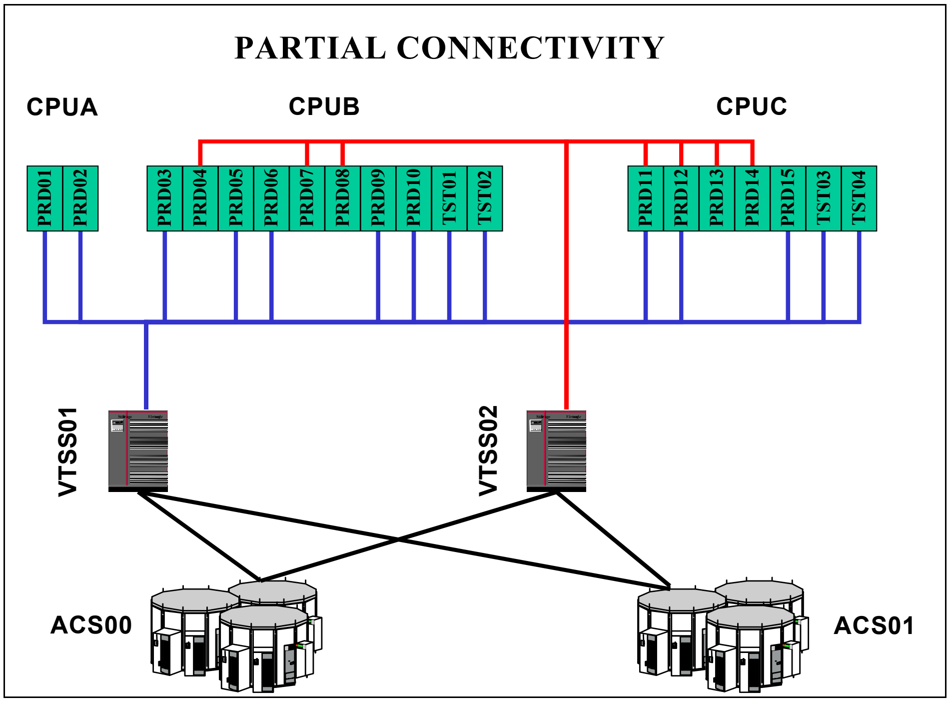


Figure 29. Partial VSM Connectivity

If you have elected to go with Partial Connectivity for throughput, consider the following:

1. If a connection from one LPAR to a VTSS is initially configured, **don't take it away later** because you will generate ECAM activity that cannot complete.
2. Before you initially create a partial configuration, consider the following scenario. For example, in Figure 29:
   - First, PRD04 creates VTV001 on VTSS02.
   - PRD03 (which is **not connected** to VTSS02) then tries to mount VTV001...which will not work because VTCS in PRD03 can't check VTSS02 to ensure that there are no copies of VTV001 resident in that VTSS, which will most likely cause a 522 timeout and job failure.

   Therefore, the best way to ensure that you do **not** experience these, and other, problem VSM situations is to configure logical paths for Full Connectivity.

Appendix C. VSM2 and VSM3 Logical Pathing

This appendix consists of the following sections:

VSM2 and VSM3 Logical Pathing Overview

VSM2 and VSM3 logical pathing isn't complex, but you have to understand the following basic concepts:

- You have a **theoretical** maximum of 128 logical paths on the VTSS. However, you must have *some* RTD connections so you cannot allocate *all* 128 logical paths for host-to-VTSS connections. In addition, host-to-VTSS connections are balanced across each port. Therefore, with 16 ports (the most common), each RTD uses a port and a total of 8 logical paths. If you have 8 RTDs, they use 8 ports and 64 logical paths, leaving 8 ports and 64 logical paths for front-end connectivity. 8 ports means a maximum of 8 hosts can connect (if they each use a logical path on every available port), using multiple logical paths if required up to the total of 64. If a ninth host tries to connect...you'll get an error showing no logical paths available.
- Each VTSS has either 4 or 8 ICE cards. Each ICE card has two ports, which are used **either** for host-to-VTSS ESCON channel connections or for VTSS-to-RTD Nearlink connections.
- A logical **host** path is the communication path between a host and **all 64** VTDs within the VTSS. These logical host path(s) also provide host-to-VTSS communications.
- Each VTSS must be connected to a **minimum** of two RTDs in **each** ACS to which the VTSS is connected. VTCS cannot check device type, but StorageTek s**trongly recommends** at least two RTDS of each device type in each ACS to which the VTSS is attached....otherwise, you can seriously compromise error recovery and also impact the efficiency of space reclamation. In an 8 ICE card configuration with a single media type, this leaves 14 available ports for host-to-VTSS ESCON channel connections, which equals a **maximum** of 112 logical paths.
- Logical paths to the host(s) are **evenly distributed** across the ICE Cards (and ports) used for host-to VTSS connections, so that:
  - **In an 8-ICE card VTSS**, each ICE card port used for a **physical** host-to-VTSS ESCON channel connection provides **8 logical** paths to the host.
  - **In a 4-ICE card VTSS**, each ICE card port used for a **physical** host-to-VTSS ESCON channel connection provides **16 logical** paths to the host.
- The **actual** host-to-VTSS pathing configuration, therefore, depends on:
  - The number of VTSS ICE Cards installed.
  - The number of RTDs attached to the VTSS.
  - Performance considerations. You should consider assigning more logical paths to hosts requiring performance and redundancy.

In the following sections, let's look at some typical examples of ICE card port configurations and the host logical paths available with each:

- "Host Paths for VTSS with 8 ICE Cards, 4 RTD Nearlink Connections"
- "Host Paths for VTSS with 8 ICE Cards, 8 RTD Nearlink Connections" on page 134
- "Host Paths for VTSS with 4 ICE Cards, 4 RTD Nearlink Connections" on page 134

Host Paths for VTSS with 8 ICE Cards, 4 RTD Nearlink Connections

As Table 26 shows, a VTSS with 8 ICE cards and 4 RTD Nearlink connections supports a maximum of 96 logical host paths.

Table 26. Host Paths for VTSS with 8 ICE Cards, 4 RTD Nearlink Connections

ICE Card Number	ICE00	ICE01	ICE02	ICE03	ICE13	ICE12	ICE11	ICE10	Total Host Paths
1st Port Connection	RTD	RTD	8 host paths	8 host paths	RTD	RTD	8 host paths	8 host paths	32
2nd Port Connection	8 host paths	8 host paths	8 host paths	8 host paths	8 host paths	8 host paths	8 host paths	8 host paths	64
Total Host Paths (Both Cards)									96

Host Paths for VTSS with 8 ICE Cards, 8 RTD Nearlink Connections

As Table 27 shows, a VTSS with 8 ICE cards and 8 RTD Nearlink connections supports a maximum of 64 logical host paths.

Table 27. Host Paths for VTSS with 8 ICE Cards, 8 RTD Nearlink Connections

ICE Card Number	ICE00	ICE01	ICE02	ICE03	ICE13	ICE12	ICE11	ICE10	Total Host Paths
1st Port Connection	RTD	8 host paths	RTD	8 host paths	RTD	8 host paths	RTD	8 host paths	32
2nd Port Connection	8 host paths	RTD	8 host paths	RTD	8 host paths	RTD	8 host paths	RTD	32
Total Host Paths (Both Cards)									64

Host Paths for VTSS with 4 ICE Cards, 4 RTD Nearlink Connections

As Table 28 shows, a VTSS with 4 ICE cards and 4 RTD Nearlink connections supports a maximum of 64 logical host paths.

Table 28. Host Paths for VTSS with 4 ICE Cards, 4 RTD Nearlink Connections

ICE Card Number	ICE00	ICE02	ICE12	ICE10	Total Host Paths
1st Port Connection	RTD	16 host paths	16 host paths	RTD	32
2nd Port Connection	16 host paths	RTD	RTD	16 host paths	32
Total Host Paths (Both Cards)					64

VSM2 and VSM3 Logical Path Planning and Configuration Example

In this example, we connect 3 CPUs, each with multiple LPARS, to two VTSSs. This example builds on the concepts described in "VSM2 and VSM3 Logical Pathing Overview" on page 132.

The planning and configuration of this example consists of four steps:

- "Step 1: Determine Logical Pathing Requirements" on page 136. For each LPAR, we analyze the number of logical paths for each of the following:
  - **Throughput**. In Table 29 on page 138 and Table 30 on page 139, the bandwidth requirements for each LPAR correspond to the number of logical paths we allocate for throughput.
  - **Redundancy**. We allocate redundant logical paths to LPARs that rank high on the "require continuous operations" scale.
  - **Connectivity**. A logical path for connectivity is basically a "yes" answer to the question "Does this LPAR need a connection to a VTSS?" For more information, see Appendix B "VSM Connectivity Requirements".

  Hint: Note the following about Table 29 on page 138 and Table 30 on page 139. Basically, if we allocate any logical paths in the **Throughput** column, we do **not** have to explicitly allocate a logical path for connectivity in the **Add 1 for Connectivity?** column. If we do **not** allocate any logical paths in the **Throughput** column, we **must** explicitly allocate one logical path for connectivity in the **Add 1 for Connectivity?** column.
- "Step 2: Determine Channel Requirements and Allocate Channels" on page 140. The logical path requirements we sized in Step 1 are the input to determining the channel requirements and allocation.
- "Step 3: Allocate Logical Paths" on page 141. Here we overlay the logical paths requirements from Step1 on top of the channel allocations in Step 2.
- "Step 4: Code The IOCP" on page 150. This is the easy part, thanks to the careful planning we did in Steps 1 through 3. We simply code the IOCP to match the final results, shown in Table 36 on page 144 and Table 41 on page 148.

Step 1: Determine Logical Pathing Requirements

"Logical Pathing Requirements for VTSS01" on page 138 and "Logical Pathing Requirements for VTSS02" on page 139 show examples of allocating logical paths for both VTSSs to satisfy the throughput, connectivity, and redundancy requirements for each LPAR. Now...we said earlier that Logical Pathing really isn't complex, but it requires the following information to decipher Table 29 on page 138 and Table 30 on page 139. For each **HOST/LPAR** for a specific **CPU**, we have the following information:

- **Bandwidth (MB/Sec)** is the estimated maximum bandwidth required for this **HOST/LPAR**.
- **Number of Front End Paths Required** is derived from its four subcolumns:
  - **Throughput** is the number of paths required to provide the estimated maximum bandwidth (**Bandwidth (MB/Sec)**). **Note that** for these examples, we're assuming a 9 MB/sec bandwidth per path and we're rounding up.

    For example, in Table 29 on page 138, the HOST/LPAR PRD10 on CPUB has a 15 MB/sec throughput requirement, so we round up to 2 paths times 9MB/sec for a total of 18 MB/sec.
  - **Redundancy** is the **additional** number of paths required for redundancy. This is, again, an estimate based on the criticality of the application(s) on any particular HOST/LPAR.

    Following along with our example, in Table 29 on page 138, the HOST/LPAR PRD10 on CPUB has a mission-critical payroll application, so we give it an one additional path for redundancy.
  - **Add 1 for Connectivity?** As we said back in Appendix B "VSM Connectivity Requirements", configuring for full VSM connectivity is the best way to avoid the kinds of problems that occur with partial connectivity. If you did not allocate any paths for throughput or redundancy, **make sure** that you allocate sufficient paths for connectivity. If, on the other hand, you allocate any paths for throughput/redundancy that **also** ensure full connectivity, you do not need an explicit allocation for redundancy......so this column is either a 0 or 1.

    In Table 29 on page 138, the HOST/LPAR PRD10 on CPUB has plenty of paths allocated for throughput/redundancy, so we don't need to allocate one for connectivity. PRD04 on CPUA, on the other hand, didn't get any for throughput or redundancy, so we allocate one for connectivity.

- **Total** is simply the total of the previous three columns. Notice that in the **Total** column you'll see some numbers in **bold**. That means within a CPU, the HOST/LPAR with the highest total paths wins, so that's what we allocate for that CPU. For example, In Table 29 on page 138, for CPUA, HOST/LPAR PRD01 needs a total of 3 paths, and PRD02 needs a total of 4 paths, so we allocate 4 paths for CPUA...maybe.

  We say "maybe" because this scheme does not take into account concurrent requirements across the HOST/LPARs within each CPU, so you may want to allow more paths for these requirements. Our calculations do, however, give an indication of the ratio of ports that might be allotted to each CPU. For example, in CPUB, PRD04, PRD07 and PRD08 might each be passing data concurrently, so you might want to allocate something like an additional 5 paths for this activity. As you'll see in "Step 3: Allocate Logical Paths" on page 141, our calculations are rigorous but do not use all available logical paths.

  These **bold** numbers in the **Total** column are key to "Step 2: Determine Channel Requirements and Allocate Channels" on page 140, because we plug these numbers into Table 31 on page 140.

Logical Pathing Requirements for VTSS01

Table 29 shows logical pathing requirements for VTSS01.

Table 29. Logical Pathing Requirements for VTSS01

CPU	HOST/ LPAR	Bandwidth (MB/Sec)	Number of Front End Paths Required			
			Throughput	Redundancy	Add 1 for Connectivity?	Total
CPUA	PRD01	12	2	1	0	3
	PRD02	34	4	0	0	**4**
CPUB	PRD03	8	1	1	0	2
	PRD04	0	0	0	1	1
	PRD05	15	2	0	0	2
	PRD06	12	2	0	0	2
	PRD07	0	0	0	1	1
	PRD08	0	0	0	1	1
	PRD09	12	2	0	0	2
	PRD10	15	2	1	0	**3**
	TST01	5	1	1	0	2
	TST02	5	1	1	0	2
CPUC	PRD11	15	2	0	0	**2**
	PRD12	12	2	0	0	**2**
	PRD13	0	0	0	1	1
	PRD14	0	0	0	1	1
	PRD15	12	2	0	0	**2**
	TST03	5	1	0	0	1
	TST04	5	1	0	0	1

Logical Pathing Requirements for VTSS02

Table 30 shows logical pathing requirements for VTSS02.

Table 30. Logical Pathing Requirements for VTSS02

CPU	HOST/ LPAR	Bandwidth (MB/Sec)	Number of Front End Paths Required			
			Throughput	Redundancy	Add 1 for Connectivity?	Total
CPUA	PRD01	0	0	0	1	**1**
	PRD02	0	0	0	1	**1**
CPUB	PRD03	0	0	0	1	1
	PRD04	12	2	1	0	**3**
	PRD05	0	0	0	1	1
	PRD06	0	0	0	1	1
	PRD07	12	2	1	0	**3**
	PRD08	15	2	1	0	**3**
	PRD09	0	0	0	1	1
	PRD10	0	0	0	1	1
	TST01	0	0	0	1	1
	TST02	0	0	0	1	1
CPUC	PRD11	12	2	1	0	**3**
	PRD12	12	2	1	0	**3**
	PRD13	18	2	1	0	**3**
	PRD14	18	2	1	0	**3**
	PRD15	6	1	1	0	2
	TST03	0	0	0	1	1
	TST04	0	0	0	1	1

Step 2: Determine Channel Requirements and Allocate Channels

Using the logical path requirements determined in “Step 1: Determine Logical Pathing Requirements” on page 136, we next determine the channel requirements for each CPU in Table 31. Table 32 then summarizes the actual channel allocations (CHPIDs and connection to VTSS01 or VTSS02).

Table 31. Channel Requirements for Each CPU

CPU	Front-End Channels Required		
	VTSS01	**VTSS02**	**Total**
RTDs	6	6	12
CPUA	4	1	5
CPUB	3	3	6
CPUC	2	3	5
Total	15	13	

Note: The total number of front-end channels can be a **maximum** of 16 per VTSS. If it exceeds 16, go back to Table 30. on page 139 and Table 29. on page 138 and rework to reduce the total logical paths to 16 or less

Table 32. Channel Allocation for Each CPU

CPU	CHPIDs	VTSS	
		VTSS01	**VTSS02**
CPUA	06	*	
	2E		*
	4F	*	
	D4	*	
	E3	*	
CPUB	2F	*	
	32	*	
	85	*	
	8C		*
	A4		*
	C4		*
CPUC	0F	*	
	30		*
	58	*	
	88		*
	8D		*

Step 3: Allocate Logical Paths

In this section, the objective is to allocate logical paths to satisfy the throughput/connectivity/redundancy considerations we determined **within the restriction** that we have a maximum of eight paths per port on an 8 ICE card VTSS. We use this allocation to build the Access List and place additional connections in the Candidate List, which we describe in “Step 4: Code The IOCP” on page 150. Note that these are **minimum configurations** for the requirements...in fact, there are logical paths left over that can be allocated if needed.

We’ll start with VTSS01, and we’ll do this in the following stages:

- “VTSS01 Logical Paths for Throughput”
- “VTSS01 Logical Paths for Throughput and Connectivity” on page 142
- “VTSS01 Logical Paths for Throughput, Connectivity, and Redundancy” on page 143
- “VTSS01 Logical Paths for Throughput, Connectivity, Redundancy, and RTD Connections” on page 144
- “VTSS01 Unallocated Logical Paths” on page 145

VTSS01 Logical Paths for Throughput

Table 33 shows **only** the VTSS01 logical paths required for throughput.

Table 33. VTSS01 Host Logical Paths for Throughput

VTSS Port	CPU	CHPID	Host Logical Paths Allocated							
001	CPUA	06	PRD01	PRD02						
011	CPUC	0F	PRD11	PRD12	PRD15	TST03				
021	CPUB	2F	PRD03	PRD06	PRD09	TST01				
030	CPUA	4F	PRD01	PRD02						
031	CPUB	32	PRD05	PRD06	PRD10	TST02				
101	CPUA	D4	PRD02							
111	CPUC	58	PRD11	PRD12	PRD15	TST04				
130	CPUA	E3	PRD02							
131	CPUB	85	PRD05	PRD09	PRD10					

VTSS01 Logical Paths for Throughput and Connectivity

Table 34 shows the VTSS01 logical paths required for throughput and connectivity (shaded in the table).

Table 34. VTSS01 Host Logical Paths for Throughput and Connectivity

VTSS Port	CPU	CHPID	Host Logical Paths Allocated							
001	CPUA	06	PRD01	PRD02						
011	CPUC	0F	PRD11	PRD12	PRD15	TST03	PRD13			
021	CPUB	2F	PRD03	PRD06	PRD09	TST01	PRD07			
030	CPUA	4F	PRD01	PRD02						
031	CPUB	32	PRD05	PRD06	PRD10	TST02	PRD08			
101	CPUA	D4	PRD02							
111	CPUC	58	PRD11	PRD12	PRD15	TST04	PRD14			
130	CPUA	E3	PRD02							
131	CPUB	85	PRD05	PRD09	PRD10	PRD04				

VTSS01 Logical Paths for Throughput, Connectivity, and Redundancy

Table 34 shows the VTSS01 logical paths required for throughput, connectivity, and redundancy (shaded in the table).

Table 35. VTSS01 Host Logical Paths for Throughput, Connectivity, and Redundancy

VTSS Port	CPU	CHPID	Host Logical Paths Allocated							
001	CPUA	06	PRD01	PRD02						
011	CPUC	0F	PRD11	PRD12	PRD15	TST03	PRD13			
021	CPUB	2F	PRD03	PRD06	PRD09	TST01	PRD07	TST02		
030	CPUA	4F	PRD01	PRD02						
031	CPUB	32	PRD05	PRD06	PRD10	TST02	PRD08	TST01		
101	CPUA	D4	PRD02							
111	CPUC	58	PRD11	PRD12	PRD15	TST04	PRD14			
130	CPUA	E3	PRD02	PRD01						
131	CPUB	85	PRD05	PRD09	PRD10	PRD04	PRD03			

VTSS01 Logical Paths for Throughput, Connectivity, Redundancy, and RTD Connections

Table 36 shows the VTSS01 logical paths required for throughput, connectivity, redundancy, and RTD connections (shaded in the table).

Table 36. VTSS01 Host Logical Paths for Throughput, Connectivity, and Redundancy, and RTD Connections

VTSS Port	CPU	CHPID	Host Logical Paths Allocated							
000	RTD									
001	CPUA	06	PRD01	PRD02						
010	RTD									
011	CPUC	0F	PRD11	PRD12	PRD15	TST03	PRD13			
020	RTD									
021	CPUB	2F	PRD03	PRD06	PRD09	TST01	PRD07	TST02		
030	CPUA	4F	PRD01	PRD02						
031	CPUB	32	PRD05	PRD06	PRD10	TST02	PRD08	TST01		
100	RTD									
101	CPUA	D4	PRD02	PRD01						
110	RTD									
111	CPUC	58	PRD11	PRD12	PRD15	TST04	PRD14			
120	RTD									
121	SPARE									
130	CPUA	E3	PRD02							
131	CPUB	85	PRD05	PRD09	PRD10	PRD04	PRD03			

VTSS01 Unallocated Logical Paths

Table 37 shows the VTSS01 unallocated logical paths (shaded in the table).

Table 37. VTSS01 Unallocated Host Logical Paths

VTSS Port	CPU	CHPID	Host Logical Paths Allocated							
000	RTD									
001	CPUA	06	PRD01	PRD02						
010	RTD									
011	CPUC	0F	PRD11	PRD12	PRD15	TST03	PRD13			
020	RTD									
021	CPUB	2F	PRD03	PRD06	PRD09	TST01	PRD07	TST02		
030	CPUA	4F	PRD01	PRD02						
031	CPUB	32	PRD05	PRD06	PRD10	TST02	PRD08	TST01		
100	RTD									
101	CPUA	D4	PRD02	PRD01						
110	RTD									
111	CPUC	58	PRD11	PRD12	PRD15	TST04	PRD14			
120	RTD									
121	SPARE									
130	CPUA	E3	PRD02							
131	CPUB	85	PRD05	PRD09	PRD10	PRD04	PRD03			

Next, we'll allocate logical paths for VTSS02 as follows:

- "VTSS02 Logical Paths for Throughput"
- "VTSS02 Logical Paths for Throughput and Connectivity"
- "VTSS02 Logical Paths for Throughput, Connectivity, and Redundancy" on page 147
- "VTSS02 Logical Paths for Throughput, Connectivity, Redundancy, and RTD Connections" on page 148
- "VTSS02 Unallocated Logical Paths" on page 149

VTSS02 Logical Paths for Throughput

Table 38 shows **only** the VTSS02 logical paths required for throughput.

Table 38. VTSS02 Host Logical Paths for Throughput

VTSS Port	CPU	CHPID	Host Logical Paths Allocated							
011	CPUB	8C	PRD04	PRD07						
021	CPUC	30	PRD11	PRD12	PRD14					
030	CPUB	A4	PRD04	PRD08						
101	CPUC	88	PRD11	PRD13	PRD14					
111	CPUC	8D	PRD12	PRD13	PRD15					
130	CPUB	C4	PRD07	PRD08						

VTSS02 Logical Paths for Throughput and Connectivity

Table 39 shows the VTSS02 logical paths required for throughput and connectivity (shaded in the table).

Table 39. VTSS02 Host Logical Paths for Throughput and Connectivity

VTSS Port	CPU	CHPID	Host Logical Paths Allocated							
001	CPUA	2E	PRD01	PRD02						
011	CPUB	8C	PRD04	PRD07	PRD03	PRD09	TST02			
021	CPUC	30	PRD11	PRD12	PRD14	TST03				
030	CPUB	A4	PRD04	PRD08	PRD05	PRD10				
101	CPUC	88	PRD11	PRD13	PRD14	TST04				
111	CPUC	8D	PRD12	PRD13	PRD15					
130	CPUB	C4	PRD07	PRD08	PRD06	TST01				

VTSS02 Logical Paths for Throughput, Connectivity, and Redundancy

Table 40 shows the VTSS02 logical paths required for throughput, connectivity, and redundancy (shaded in the table).

Table 40. VTSS02 Host Logical Paths for Throughput, Connectivity, and Redundancy

VTSS Port	CPU	CHPID	Host Logical Paths Allocated							
001	CPUA	2E	PRD01	PRD02						
011	CPUB	8C	PRD04	PRD07	PRD03	PRD09	TST02	PRD08		
021	CPUC	30	PRD11	PRD12	PRD14	TST03	PRD13	PRD15		
030	CPUB	A4	PRD04	PRD08	PRD05	PRD10	PRD07			
101	CPUC	88	PRD11	PRD13	PRD14	TST04	PRD12			
111	CPUC	8D	PRD12	PRD13	PRD15	PRD11	PRD14			
130	CPUB	C4	PRD07	PRD08	PRD06	TST01	PRD04			

VTSS02 Logical Paths for Throughput, Connectivity, Redundancy, and RTD Connections

Table 41 shows the VTSS02 logical paths required for throughput, connectivity, redundancy, and RTD connections (shaded in the table).

Table 41. VTSS02 Host Logical Paths for Throughput, Connectivity, and Redundancy, and RTD Connections

VTSS Port	CPU	CHPID	Host Logical Paths Allocated							
000	RTD									
001	CPUA	2E	PRD01	PRD02						
010	RTD									
011	CPUB	8C	PRD04	PRD07	PRD03	PRD09	TST02	PRD08		
020	RTD									
021	CPUC	30	PRD11	PRD12	PRD14	TST03	PRD13	PRD15		
030	CPUB	A4	PRD04	PRD08	PRD05	PRD10	PRD07			
100	RTD									
101	CPUC	88	PRD11	PRD13	PRD14	TST04	PRD12			
110	RTD									
111	CPUC	8D	PRD12	PRD13	PRD15	PRD11	PRD14			
120	RTD									
130	CPUB	C4	PRD07	PRD08	PRD06	TST01	PRD04			

VTSS02 Unallocated Logical Paths

Table 42 shows the VTSS02 unallocated logical paths (shaded in the table).

Table 42. VTSS02 Unallocated Host Logical Paths

VTSS Port	CPU	CHPID	Host Logical Paths Allocated							
000	RTD									
001	CPUA	2E	PRD01	PRD02						
010	RTD									
011	CPUB	8C	PRD04	PRD07	PRD03	PRD09	TST02	PRD08		
020	RTD									
021	CPUC	30	PRD11	PRD12	PRD14	TST03	PRD13	PRD15		
030	CPUB	A4	PRD04	PRD08	PRD05	PRD10	PRD07			
031	SPARE									
100	RTD									
101	CPUC	88	PRD11	PRD13	PRD14	TST04	PRD12			
110	RTD									
111	CPUC	8D	PRD12	PRD13	PRD15	PRD11	PRD14			
120	RTD									
121	SPARE									
130	CPUB	C4	PRD07	PRD08	PRD06	TST01	PRD04			
131	SPARE									

Step 4: Code The IOCP

Caution: Ensure that you complete Steps 1 through 3 **before** you code the IOCP, otherwise you may incur unpredictable and undesirable results!

In the IOCP example for VTSS01 in Figure 32 on page 152, we use the input from Table 32 on page 140 to define the following in the `PARTITION` statement:

- First, the **Access List**, which consists the LPARs that need access to a VTSS.

  Note: If more than 8 LPARs are coded in the Access List, the first 8 LPARs that are started will obtain one logical path through this CHPID. The 9th LPAR that is started will get "logical path not available."

- Next, the **Candidate List**, which consists of any LPARS that might need access to a VTSS some time in the future.

A visual representation of the above text is what's best at this point, so please see the following graphic that shows the position of the Access List (Figure 30 on page 151) and Candidate List (Figure 31 on page 151) in the `PARTITION` statement.

Note: Defining Access and Candidate List **is different** between HCD and IOCP. IBM's *MVS/ESA HCD and Dynamic I/O Reconfiguration Primer* (SG24-4037-01) says, "The HCD candidate list does not contain any partition that is already in the access list. It is viewed as an additional list of partitions that might get access to the channel path at a later time. Thus, a partition defined in the access list of a CHPID does not appear on the Define Candidate List panel. In IOCP, the candidate list includes the access list."

Completing the assignment of logical paths via the IOCP ensures that **only** those paths that need to be online are brought online and, in addition, all paths that need to be online (for VTCS connectivity purposes) are also online. This completes this example...and we hope it was worth the effort.

ACCESS LIST

```
CHPID PATH=(06),SHARED,                                              *
      PARTITION=(PRD01,PRD02),TYPE=CNC
```

Access List (**in bold**) with no Candidate List Specified

```
CHPID PATH=(0F),SHARED,                                              *
      PARTITION=((PRD11,PRD12,PRD13,PRD15,TST03),(PRD14,TST04)*
      ),TYPE=CNC
```

Access List (**in bold**) with Candidate List Specified as well

Figure 30. ***PARTITION Parameter on the CHPID Statement-Access List***

CANDIDATE LIST

```
CHPID PATH=(06),SHARED,                                              *
      PARTITION=(PRD01,PRD02),TYPE=CNC
```

No Candidate List Specified in this example

```
CHPID PATH=(0F),SHARED,                                              *
      PARTITION=((PRD11,PRD12,PRD13,PRD15,TST03),(PRD14,TST04)*
      ),TYPE=CNC
```

Candidate List Specified in this example (**in bold**).

Figure 31. ***PARTITION Parameter on the CHPID Statement-Candidate List***

```
CHPID PATH=(06),SHARED,                                              *
      PARTITION=(PRD01,PRD02),TYPE=CNC

CHPID PATH=(0F),SHARED,                                              *
      PARTITION=((PRD11,PRD12,PRD13,PRD15,TST03),(PRD14,TST04)*
      ),TYPE=CNC

CHPID PATH=(2F),SHARED,                                              *
      PARTITION=((PRD03,PRD06,PRD07,PRD09,TST01,TST02),(PRD04,*
      PRD05,PRD08,PRD10)),TYPE=CNC

CHPID PATH=(32),SHARED,                                              *
      PARTITION=((PRD05,PRD06,PRD08,PRD10,TST01,TST02),(PRD03,*
      PRD04,PRD07,PRD09)),TYPE=CNC

CHPID PATH=(4F),SHARED,                                              *
      PARTITION=(PRD01,PRD02),TYPE=CNC

CHPID PATH=(58),SHARED,                                              *
      PARTITION=((PRD11,PRD12,PRD14,PRD15,TST04),(PRD13,TST03)*
      ),TYPE=CNC

CHPID PATH=(85),SHARED,                                              *
      PARTITION=((PRD03,PRD04,PRD05,PRD09,PRD10),(PRD06,PRD07,*
      PRD08,TST01,TST02)),TYPE=CNC

CHPID PATH=(D4),SHARED,                                              *
      PARTITION=(PRD01,PRD02),TYPE=CNC

CHPID PATH=(E3),SHARED,                                              *
      PARTITION=(PRD02,PRD01),TYPE=CNC
```

Figure 32. ***IOCP Example for VTSS01***

Appendix D. VSM4 ESCON Configuration

The newest generation VTSS is the VSM4, which provides the following advantages over its predecessors:

- Enhanced connectivity options.
- Greater throughput.
- Greater VTSS capacity.
- 4x the number of VTDs and 3x the maximum number of VTVs per VTSS.
- Improved reliability and serviceability.

Table 43 summarizes the VSM3 to VSM4 ESCON enhancements that you see from a software and system configuration perspective.

Table 43. VSM3 to VSM4 Comparison: Software and System Configuration ESCON Enhancements

Product Feature	VSM3	VSM4
ESCON Interfaces	16 **total** where: • 2 to 14 can be host channels • 2 to 8 can be Nearlink/CLINK connections	32 **total** where: • 2 to 28 can be host channels • 2 to 16 can be Nearlink/CLINK connections Note: VSM4s are shipped with 16 ports enabled.With the 16 ports enabled option, **only** the top port on each CIP is enabled (Port 0 or Port 2). 32 ports enabled is an **optional, separately priced feature** that is activated via microcode diskette. On a VSM4 with 32 ports enabled, each ICE3 ESCON interface card contains two pairs of ESCON ports. Each pair is controlled by its own Channel Interface Processor (CIP). Each CIP switches between the two ports, so that only one port can transfer data at a time.

Table 43. VSM3 to VSM4 Comparison: Software and System Configuration ESCON Enhancements

Product Feature	VSM3	VSM4
Maximum Logical Paths	128	16 per port for the 16 port standard configuration = 256 logical paths 16 per port for the 32 port optional configuration= 512 logical paths **Note**: VSM4 provides a theoretical maximum of 512 logical paths per VTSS, but you cannot allocate all 512 logical paths for host-to-VTSS connections.
VTDs per VTSS	64	256
Maximum resident VTVs per VTSS	100,000	300,000

VSM4 with 32 Ports

For the 32 port option, the 8 ICE3 cards have four ESCON ports per card as shown in Figure 33.

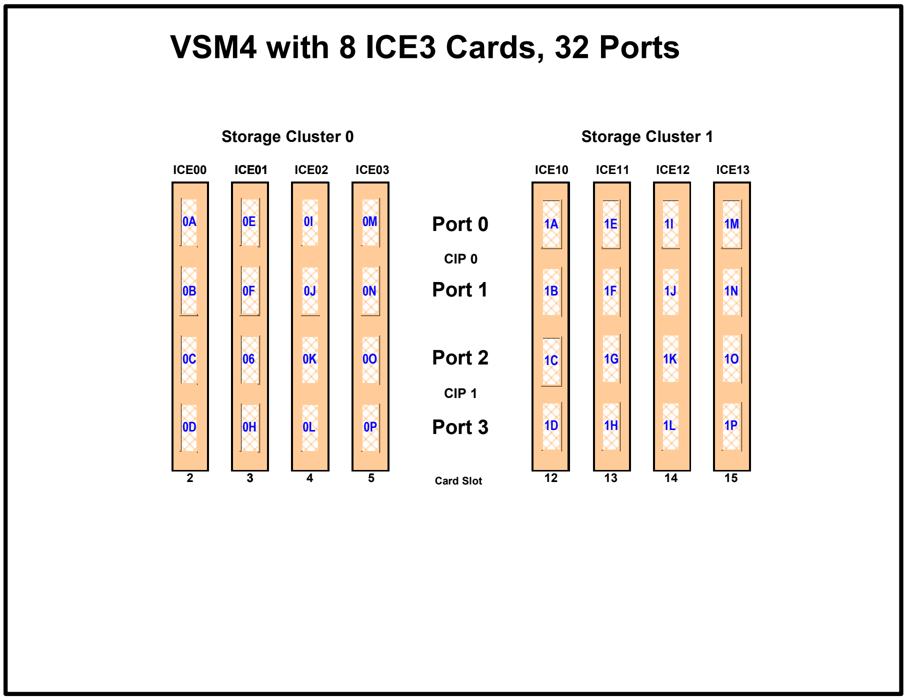


Figure 33. ***VSM4 with 32 Ports***

Note: In Figure 33 and all the other figures in this appendix, the ports are shown with their channel interface identifiers for **enabled** ports (32 in Figure 33). These channel interface identifiers are the values that are required for the CHANIF values that you code for the CONFIG utility. Each value is two characters in length and has a value from 0A to 1P. The first digit is the VTSS cluster ID (valid values are 0 or 1). The second digit is the group or adapter ID (valid values are A to P).

In Figure 33 on page 155, note the following:

- Each ICE3 card contains two pairs of ESCON ports. Each pair is controlled by its own Channel Interface Processor (CIP). Each CIP switches between the two ports, so that **only one port** can transfer data at a time.
- For a VSM4, each CIP can operate with only *one* of two “personalities”, which is set at the VTSS LOP:
  - *Host Mode*. In Host Mode, either or both ESCON ports can connect to host CPU channels, including via ESCON Director(s) or channel extenders. Ports of a CIP in Host mode **cannot** connect to RTDs or to Secondary VTSSs via CLINKS. Note, however, that Secondary VTSSs must have an ESCON port in Host Mode to connect via a CLINK *from* an ESCON port in Nearlink Mode in a Primary VTSS.

    Also note that you can have two physical paths from the same LPAR to the same CIP, as long as the two physical paths address different (not overlapping) logical control units. For example, a single host LPAR can address logical control units 0-7 on one CIP port, and 8-F on the other CIP port of the same CIP.
  - *Nearlink Mode*. In Nearlink Mode, either or both ESCON ports can connect to an RTD or via a CLINK to a Secondary VTSS. Ports of a CIP in Nearlink mode **cannot** connect to host CPU channels.You can set a **maximum** of 8 CIPs to Nearlink Mode, and here's the important fine print: only **one** Nearlink port per CIP is active at one time. What are Best Practices for optimizing port operations? See Table 44...

Table 44. Optimizing VSM4 Port Operations

Configuration - Two Ports on a CIP	Best Practices
Two CLINKs	**Don’t use**....because only one port can be active at a time. If you're doing Clustered VTSS, you want all CLINK connections to be active all the time.
CLINK and RTD	**An advantage** in Degraded Cluster Mode. You normally have fewer RTDs on the Primary VTSS because the Secondary is doing most of the migrations. If you have an offline RTD on the same CIP as an active CLINK, if the Secondary fails you can vary the CLINK offline and bring the RTD online to handle more workload on the Primary.

Table 44. ***Optimizing VSM4 Port Operations***

Configuration - Two Ports on a CIP	Best Practices
Two RTDs	**An advantage** for the following: • **Optimize use of local and remote RTDs**. During busy shifts, use only the local RTD on the CIP. During quiet periods, switch to the remote RTD for deep archive and DR work. • **Optimize use of different drive technologies**. As described in the previous bullet, use a T9840 as a local RTD, then switch to a T9940 for deep archive. You can also use this feature to migrate from older drive technology (such as 9490) to newer technology (such as 9840). Use Management and Storage Classes to read in data from older media, then switch to the newer technology drive to place data on new media. This technique effectively gives you greater physical connectivity to different drive technologies without incurring the overhead of full time, real time ESCON connections to each drive type. **Note that** Because of the “only one active” rule, if an RTD on one port is migrating or recalling a VTV, the RTD on the second port cannot be accessed until the operation on the first port completes (the RTD on the second port is in “suspend” mode, as shown by the D RTD command/utility). Best Practices suggests, therefore, that RTDs that **must** be active simultaneously should connect to different CIPs. One more piece of fine print: If you have two RTDs on a CIP, you can't share them between VTSSs.

- On a VSM4 with 32 ports enabled, you have a **theoretical** maximum of 512 logical paths on the VSM4. However, you must have *some* RTD connections so you cannot allocate *all* 512 logical paths for host-to-VTSS connections. What's the **minimum** number of RTDs? Well, it's like this: (1) `CONFIG` **will not allow** fewer than 2 RTDS per VTSS. (2) `CONFIG` cannot check device type, but StorageTek **strongly recommends** at least two RTDS of each device type in each ACS to which the VTSS is attached....otherwise, you can seriously compromise error recovery, and also impact the efficiency of space reclamation. If you had only two RTDs, Best Practices would suggest that you connect them to different ICE3 cards...and once you've done that, you've effectively used up 4 Nearlink ports due to the "CIP personality" nature of the ICE3 card. Therefore, in an 8 ICE card configuration, this leaves 28 available ports for host-to-VTSS ESCON channel connections, which equals a **maximum** of 16 x 28 or 448 logical paths. For more information, see "Logical Paths for VSM 4 with 32 Ports" on page 174.
- A **host** logical path is the communication path between a host and all of the 256 VTDs within the VSM4. Table 45 summarizes the configuration options and maximum host logical paths for a VSM4 with 32 enabled ports.

Table 45. VSM4 Configuration Options - 32 Ports

Host CIPs	Maximum Host Connections	Nearlink CIPs	Max Nearlink Connections	Maximum Host Logical Paths
8	16	8	16	256
9	18	7	14	288
10	20	6	12	320
11	22	5	10	352
12	24	4	8	384
14	28	2	4	448

- In HCD:
  - From a single MVS host, you can only define 8 channels (CHPIDs) running to a single control unit (single VSM4). **Also note that** ICE3 cards **cannot** have 2 paths from the same LPAR connected to two ports with a common CIP.
  - You use the CNTLUNIT statement to define each VSM4 as 16 3490 images.
  - You use the IODEVICE statement to define the 16 VTDs that are associated with each 3490 image.

VSM4 Configuration Examples - 32 Ports

For VSM4s with 32 ports, let's look at two examples of port configurations:

- "VSM4 Configuration Example: 16 Host Ports, 16 RTD Ports" on page 160
- "VSM 4 Configuration Example: 20 Host Ports, 12 RTD Ports" on page 162

For a VSM4 host gen example, see "IOCP Example for Single MVS Host Connected to a VSM4 Via ESCON Directors" on page 172.

VSM4 Configuration Example: 16 Host Ports, 16 RTD Ports

Figure 34 shows CONFIG channel interface identifiers of 16 for hosts, 16 for RTDs for a VSM4.

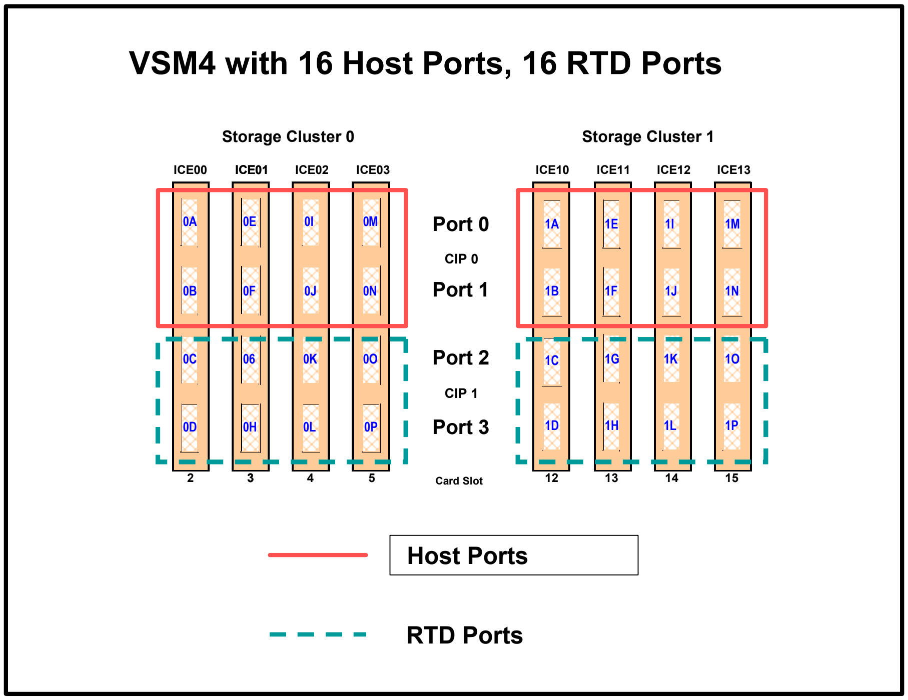


Figure 34. ***VSM4 with 16 Host Ports, 16 RTD Ports***

CONFIG Example for VSM4 with 16 Host Ports, 16 RTD Ports

Figure 35 shows example CONFIG JCL to define the VSM4 configuration shown in Figure 34 on page 160.

```
//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
//SLSPRINT     DD   SYSOUT=*
//SLSIN        DD   *
 CONFIG
 GLOBAL   MAXVTV=32000   MVCFREE=40
 RECLAIM  THRESHLD=70    MAXMVC=40  START=35
 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=VSM401 LOW=70 HIGH=80 MAXMIG=8 RETAIN=5
 RTD  NAME=VSM42A00 DEVNO=2A00 CHANIF=0C
 RTD  NAME=VSM42A01 DEVNO=2A01 CHANIF=0D
 RTD  NAME=VSM42A02 DEVNO=2A02 CHANIF=0G
 RTD  NAME=VSM42A03 DEVNO=2A03 CHANIF=0H
 RTD  NAME=VSM42A04 DEVNO=2A04 CHANIF=0K
 RTD  NAME=VSM42A05 DEVNO=2A05 CHANIF=0L
 RTD  NAME=VSM42A06 DEVNO=2A06 CHANIF=0O
 RTD  NAME=VSM42A07 DEVNO=2A07 CHANIF=0P
 RTD  NAME=VSM42A08 DEVNO=2A08 CHANIF=1C
 RTD  NAME=VSM42A09 DEVNO=2A09 CHANIF=1D
 RTD  NAME=VSM42A0A DEVNO=2A0A CHANIF=1G
 RTD  NAME=VSM42A0B DEVNO=2A0B CHANIF=1H
 RTD  NAME=VSM42A0C DEVNO=2A0C CHANIF=1K
 RTD  NAME=VSM42A0D DEVNO=2A0D CHANIF=1L
 RTD  NAME=VSM42A0E DEVNO=2A0E CHANIF=1O
 RTD  NAME=VSM42A0F DEVNO=2A0F CHANIF=1P
 VTD LOW=9900 HIGH=99FF
```

Figure 35. `CONFIG` ***example: VSM4 with 16 Host Ports, 16 RTD Ports***

VSM 4 Configuration Example: 20 Host Ports, 12 RTD Ports

Figure 36 shows port assignments of 20 for hosts, 12 for RTDs for a VSM4.

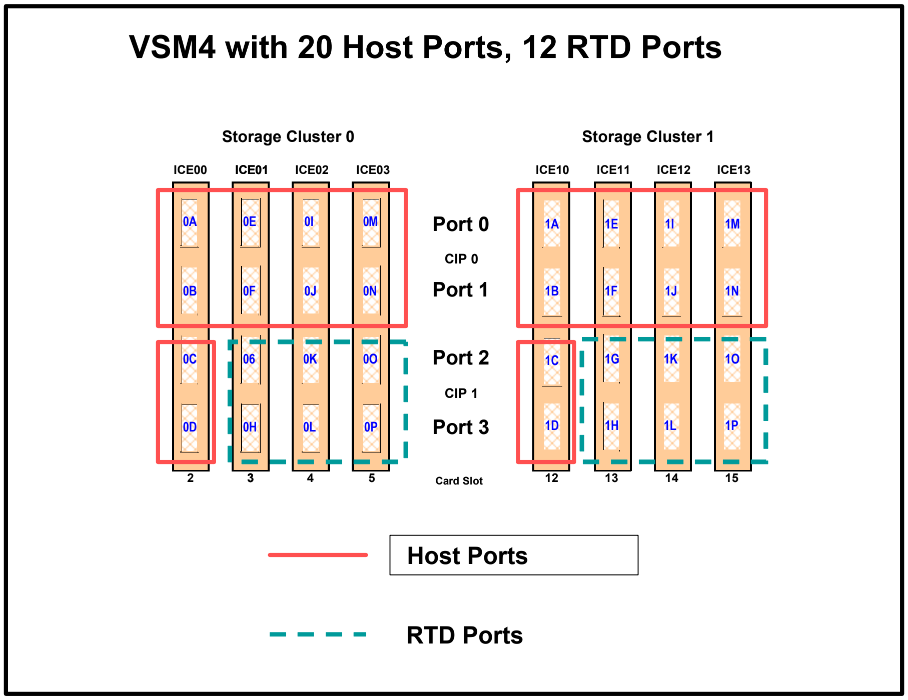


Figure 36. ***VSM4 with 20 Host Ports, 12 RTD Ports***

CONFIG Example for VSM4 with 20 Host Ports, 12 RTD Ports

Figure 37 shows example CONFIG JCL to define the VSM4 configuration shown in Figure 36 on page 162.

```
//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
//SLSPRINT     DD   SYSOUT=*
//SLSIN        DD   *
 CONFIG
 GLOBAL        MAXVTV=32000 MVCFREE=40
 RECLAIM       THRESHLD=70 MAXMVC=40  START=35
 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=VSM401 LOW=70 HIGH=80 MAXMIG=6 RETAIN=5
 RTD  NAME=VSM42A00 DEVNO=2A00 CHANIF=0G
 RTD  NAME=VSM42A01 DEVNO=2A01 CHANIF=0H
 RTD  NAME=VSM42A02 DEVNO=2A02 CHANIF=0K
 RTD  NAME=VSM42A03 DEVNO=2A03 CHANIF=0L
 RTD  NAME=VSM42A04 DEVNO=2A04 CHANIF=0O
 RTD  NAME=VSM42A05 DEVNO=2A05 CHANIF=0P
 RTD  NAME=VSM42A06 DEVNO=2A06 CHANIF=1G
 RTD  NAME=VSM42A07 DEVNO=2A07 CHANIF=1H
 RTD  NAME=VSM42A08 DEVNO=2A08 CHANIF=1K
 RTD  NAME=VSM42A09 DEVNO=2A09 CHANIF=1L
 RTD  NAME=VSM42A0A DEVNO=2A0A CHANIF=1O
 RTD  NAME=VSM42A0B DEVNO=2A0B CHANIF=1P
 VTD LOW=9900 HIGH=99FF
```

Figure 37. `CONFIG` ***example: VSM4 with 20 Host Ports, 12 RTD Ports***

VSM4 with 16 Ports

For the 16 port option, the 8 ICE3 cards have two ESCON ports per card as shown in Figure 38.

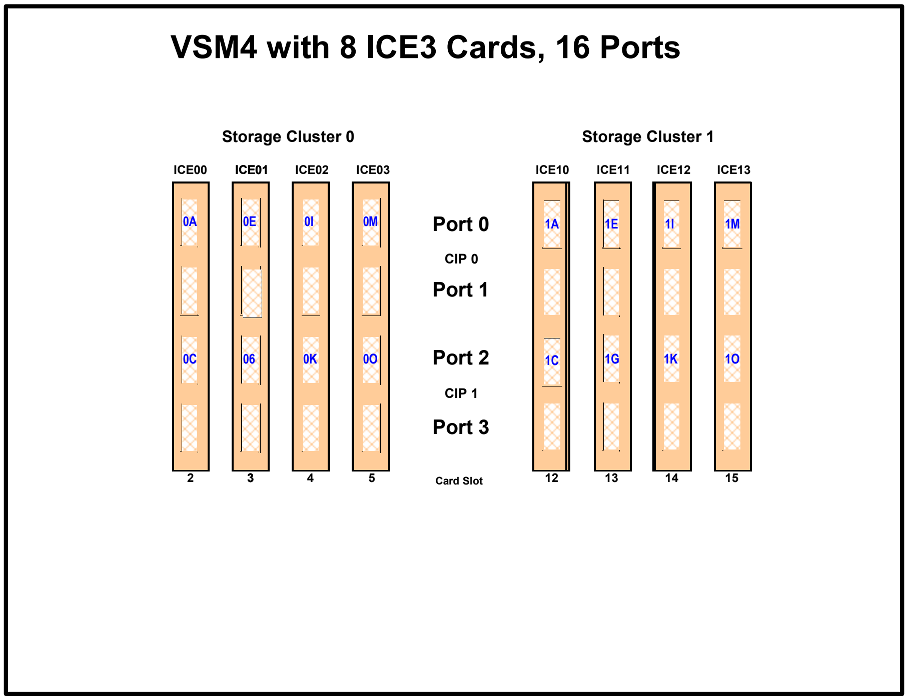


Figure 38. ***VSM4 with 16 Ports***

Note: In Figure 38 and all the other figures in this appendix, the ports are shown with their channel interface identifiers for **enabled** ports (16 in Figure 38). These channel interface identifiers are the values that are required for the CHANIF values that you code for the CONFIG utility. Each value is two characters in length and has a value from 0A to 1P. The first digit is the VTSS cluster ID (valid values are 0 or 1). The second digit is the group or adapter ID (valid values are A to P).

In Figure 38 on page 164, note the following:

- Each ICE3 card has two CIPs with **a single port enabled** on each CIP. As with the 32 port option, each CIP can operate with only *one* of two “personalities”, which is set at the VTSS LOP:
  - *Host Mode*. In Host Mode, the single ESCON port can connect to host CPU channels, including via ESCON Director(s) or channel extenders. Ports of a CIP in Host mode **cannot** connect to RTDs or to Secondary VTSSs via CLINKS. Note, however, that Secondary VTSSs must have an ESCON port in Host Mode to connect via a CLINK *from* an ESCON port in Nearlink Mode in a Primary VTSS.
  - *Nearlink Mode*. In Nearlink Mode, the single ESCON port can connect to an RTD or via a CLINK to a Secondary VTSS. Ports of a CIP in Nearlink mode **cannot** connect to host CPU channels.

    You can set a **maximum** of 8 CIPs to Nearlink Mode. Therefore, in a 16 port configuration, the single port on a CIP can be either a CLINK or an RTD connection.
- On a VSM4 with 16 ports enabled, you have a **theoretical** maximum of 256 logical paths on the VSM4. However, you must have *some* RTD connections so you cannot allocate *all* 256 logical paths for host-to-VTSS connections. What’s the **minimum** number of RTDs? Well, it's like this: (1) `CONFIG` will not allow fewer than 2 RTDS per VTSS. (2) `CONFIG` cannot check device type, but StorageTek s**trongly recommends** at least two RTDS of each device type in each ACS to which the VTSS is attached....otherwise, you can seriously compromise error recovery and also impact the efficiency of space reclamation. If you had only two RTDs, Best Practices would suggest that you connect them to different ICE3 cards...and once you’ve done that, you’ve effectively used up 4 Nearlink ports. Therefore, in an 8 ICE card 16 port configuration, this leaves 12 available ports for host-to-VTSS ESCON channel connections, which equals a **maximum** of 16 x 12 or 192 logical paths. For more information, see “Logical Paths for VSM 4 with 32 Ports” on page 174.
- A **host** logical path is the communication path between a host and all of the 256 VTDs within the VSM4. Table 46 summarizes the configuration options and maximum host logical paths for a VSM4 with 16 enabled ports.

Table 46. VSM4 Configuration Options - 16 Ports

Host CIPs	Maximum Host Connections	Nearlink CIPs	Max Nearlink Connections	Maximum Host Logical Paths
8	8	8	8	128
9	9	7	7	144
10	10	6	6	160
11	11	5	5	176

Table 46. VSM4 Configuration Options - 16 Ports

Host CIPs	Maximum Host Connections	Nearlink CIPs	Max Nearlink Connections	Maximum Host Logical Paths
12	12	4	4	192
14	14	2	2	224

- In HCD:
  - From a single MVS host, you can only define 8 channels (CHPIDs) running to a single control unit (single VSM4). **Also note that** ICE3 cards **cannot** have 2 paths from the same LPAR connected to two ports with a common CIP.
  - You use the CNTLUNIT statement to define each VSM4 as 16 3490 images.
  - You use the IODEVICE statement to define the 16 VTDs that are associated with each 3490 image.

VSM4 Configuration Examples - 16 Ports

For VSM4s with 16 ports, let’s look at two examples of port configurations:

- “VSM4 Configuration Example: 8 Host Ports, 8 RTD Ports” on page 168
- “VSM 4 Configuration Example: 10 Host Ports, 6 RTD Ports” on page 170

For a VSM4 host gen example, see “IOCP Example for Single MVS Host Connected to a VSM4 Via ESCON Directors” on page 172.

VSM4 Configuration Example: 8 Host Ports, 8 RTD Ports

Figure 39 shows CONFIG channel interface identifiers of 8 for hosts, 8 for RTDs for an 8 ICE3 card VSM4 with 16 ports.

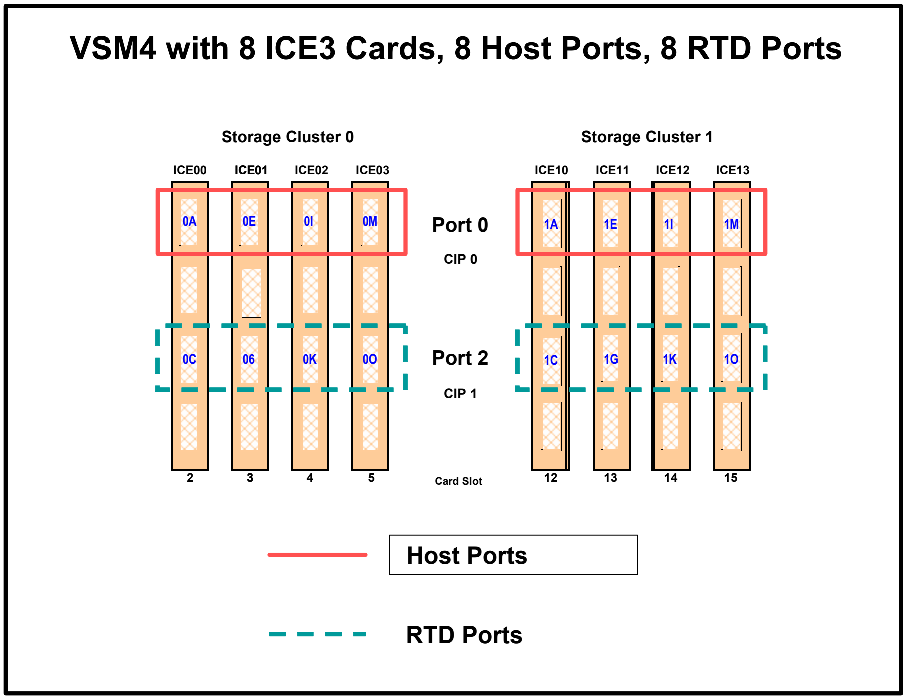


Figure 39. ***VSM4 with 8 Host Ports, 8 RTD Ports***

CONFIG Example for VSM4 with 8 Host Ports, 8 RTD Ports

Figure 40 shows example CONFIG JCL to define the VSM4 configuration shown in Figure 39 on page 168.

```
//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
//SLSPRINT    DD   SYSOUT=*
//SLSIN       DD   *
 CONFIG
 GLOBAL   MAXVTV=32000   MVCFREE=40
 RECLAIM  THRESHLD=70    MAXMVC=40  START=35
 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=VSM401 LOW=70 HIGH=80 MAXMIG=8 RETAIN=5
 RTD  NAME=VSM42A00 DEVNO=2A00 CHANIF=0C
 RTD  NAME=VSM42A02 DEVNO=2A02 CHANIF=0G
 RTD  NAME=VSM42A04 DEVNO=2A04 CHANIF=0K
 RTD  NAME=VSM42A06 DEVNO=2A06 CHANIF=0O
 RTD  NAME=VSM42A08 DEVNO=2A08 CHANIF=1C
 RTD  NAME=VSM42A0A DEVNO=2A0A CHANIF=1G
 RTD  NAME=VSM42A0C DEVNO=2A0C CHANIF=1K
 RTD  NAME=VSM42A0E DEVNO=2A0E CHANIF=1O
 VTD LOW=9900 HIGH=99FF
```

Figure 40. CONFIG ***example: VSM4 with 8 Host Ports, 8 RTD Ports***

VSM 4 Configuration Example: 10 Host Ports, 6 RTD Ports

Figure 41 shows port assignments of 10 for hosts, 6 for RTDs for a VSM4 with 16 ports.

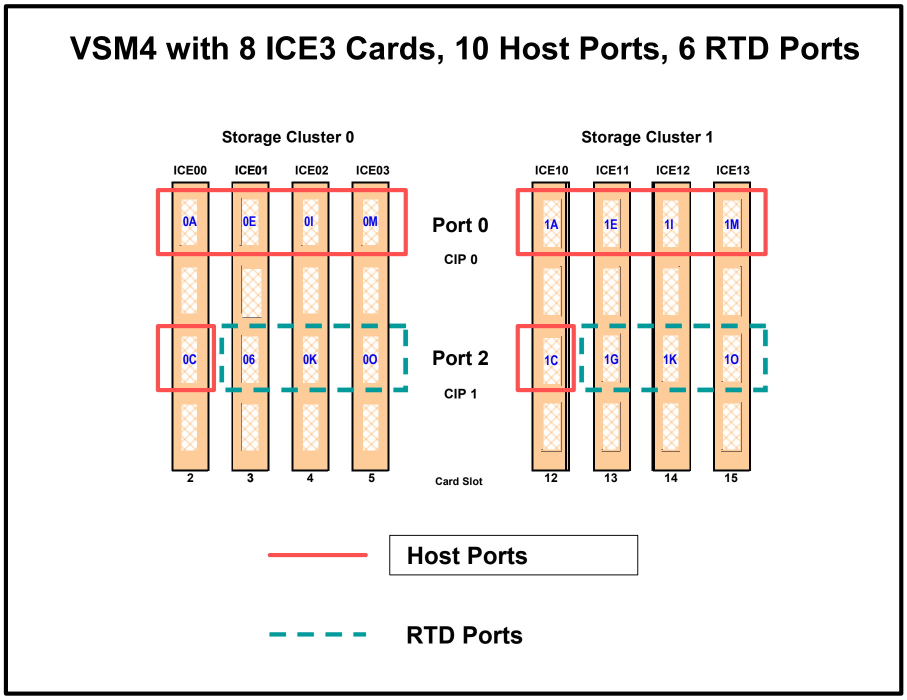


Figure 41. ***VSM4 with 10 Host Ports, 6 RTD Ports***

CONFIG Example for VSM4 with 10 Host Ports, 6 RTD Ports

Figure 42 shows example CONFIG JCL to define the VSM4 configuration shown in Figure 41 on page 170.

```
//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
//SLSPRINT    DD   SYSOUT=*
//SLSIN       DD   *
 CONFIG
 GLOBAL   MAXVTV=32000   MVCFREE=40
 RECLAIM  THRESHLD=70    MAXMVC=40  START=35
 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=VSM401 LOW=70 HIGH=80 MAXMIG=8 RETAIN=5
 RTD  NAME=VSM42A02 DEVNO=2A02 CHANIF=0G
 RTD  NAME=VSM42A04 DEVNO=2A04 CHANIF=0K
 RTD  NAME=VSM42A06 DEVNO=2A06 CHANIF=0O
 RTD  NAME=VSM42A0A DEVNO=2A0A CHANIF=1G
 RTD  NAME=VSM42A0C DEVNO=2A0C CHANIF=1K
 RTD  NAME=VSM42A0E DEVNO=2A0E CHANIF=1O
 VTD LOW=9900 HIGH=99FF
```

Figure 42. CONFIG ***example: VSM4 with 10 Host Ports, 6 RTD Ports***

IOCP Example for Single MVS Host Connected to a VSM4 Via ESCON Directors

Figure 43 shows a configuration diagram for a single MVS host connected to a VSM4 via ESCON Directors, and Figure 44 on page 173 shows example IOCP statements for this configuration. **Note that**:

- From MVSA, you define 8 CHPIDs, with each path switched in the ESCON Director, for a total of 8 channels running to the VSM4.
- You code 16 CNTLUNIT statements to define the VSM4 as 16 3490 images.
- You code IODEVICE statement to define the 16 VTDs that are associated with each 3490 image.

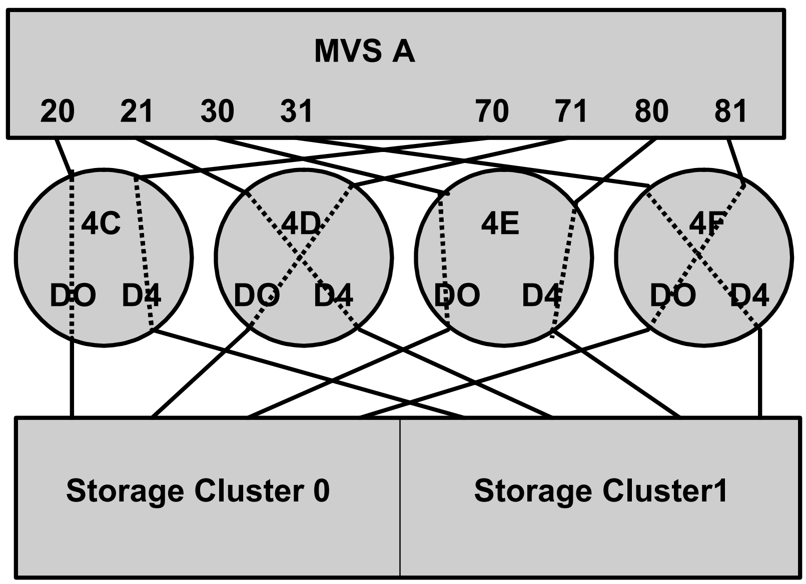


Figure 43. ***Configuration Diagram: Single MVS Host Connected to a VSM4 via ESCON Directors***

```
ESCD4C   CHPID PATH=(20,70),TYPE=CNC,SWITCH=4C
ESCD4D   CHPID PATH=(21,71),TYPE=CNC,SWITCH=4D
ESCD4E   CHPID PATH=(30,80),TYPE=CNC,SWITCH=4E
ESCD4F   CHPID PATH=(31,81),TYPE=CNC,SWITCH=4F

CU1      CNTLUNIT CUNUMBR=001,
                  PATH=(20,21,30,31,70,71,80,81),
                  LINK=(D0,D4,D0,D4,D4,D0,D4,D0),
                  UNIT=3490,CUADD=0,
                  UNITADD=((00,16))

STRING1  IODEVICE ADDRESS=(0500,16),
                  CUNUMBER=(001),
                  UNIT=3490,
                  UNITADD=00,STADET=Y

CU2      CNTLUNIT CUNUMBR=002,
                  PATH=(20,21,30,31,70,71,80,81),
                  LINK=(D0,D4,D0,D4,D4,D0,D4,D0),
                  UNIT=3490,CUADD=1,
                  UNITADD=((00,16))

STRING2  IODEVICE ADDRESS=(0510,16),
                  CUNUMBER=(002),
                  UNIT=3490,
                  UNITADD=00,STADET=Y
                  .
                  .
                  .
CU15     CNTLUNIT CUNUMBR=015,
                  PATH=(20,21,30,31,70,71,80,81),
                  LINK=(D0,D4,D0,D4,D4,D0,D4,D0),
                  UNIT=3490,CUADD=E,
                  UNITADD=((00,16))

STRING15  IODEVICE ADDRESS=(05E0,16),
                  CUNUMBER=(015),
                  UNIT=3490,
                  UNITADD=00,STADET=Y

CU16     CNTLUNIT CUNUMBR=016,
                  PATH=(20,21,30,31,70,71,80,81),
                  LINK=(D0,D4,D0,D4,D4,D0,D4,D0),
                  UNIT=3490,CUADD=F,
                  UNITADD=((00,16))

STRING16  IODEVICE ADDRESS=(05F0,16),
                  CUNUMBER=(016),
                  UNIT=3490,
                  UNITADD=00,STADET=Y
```

Figure 44. ***IOCP Example: Single MVS Host Connected to a VSM4 via ESCON Directors***

Logical Paths for VSM 4 with 32 Ports

A VSM4 with 32 ports has 4x the number of logical paths available to VSM2s and VSM3s. Does this mean that a VSM4 has enough logical paths for connectivity, redundancy, and throughput for *all* attached hosts? Even with 16 RTDs and 31 hosts attached, the answer is "yes" as shown in Figure 45.

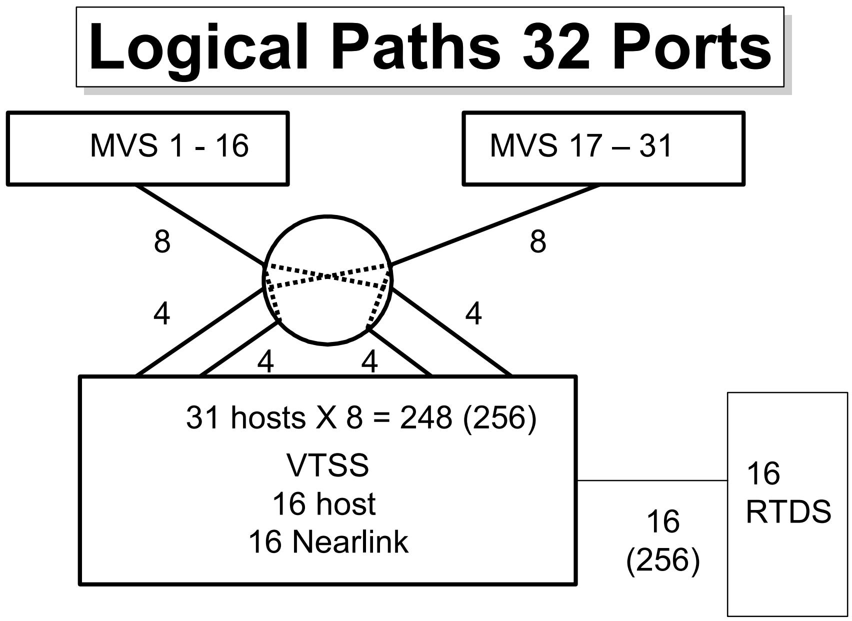


Figure 45. ***Logical Paths for VSM 4 with 32 Ports, 31 Hosts, 16 RTDs***

In Figure 45:

- The 16 RTDs consume 16 x 16 or 256 logical paths.
- Looking back at "VSM2 and VSM3 Logical Path Planning and Configuration Example" on page 135, the **maximum** logical paths we allocated for a VSM2/3 was 4 to a host requiring maximum throughput (which also satisfied the redundancy/connectivity requirements). Therefore, if we allocated **double** that number, or 8 logical paths, for each of the 31 hosts in this configuration, we only consume 248, or 8 less than the logical paths remaining for host connections.

Therefore, logical path allocation isn't an issue, as it was with VSM2s and VSM3s.

Appendix E. VSM4 FICON Front-End and Back-End Configuration

The VSM4 FICON Back-End connectivity feature adds value to the previously available FICON front-end connectivity. Table 47 summarizes the supported card configurations for VSM4 FICON Front-End plus Back-End connectivity.

Table 47. Supported Card Configurations for VSM4 FICON Front-End plus Back-End Connectivity

VCF Cards	FICON Ports	ICE Cards	ESCON Ports	Total Ports	Total Logical Paths (16 per ICE Port, 64 per VCF Port)
2	4	6	24	28	640
4	8	4	16	24	768
6	12	2	8	20	896
8	16	0	0	16	1024

VSM4 FICON VCF Card Options

VSM4 supports the following FICON VCF card options:

- Figure 46 shows a VSM4 with 6 ICE cards, 2 VCF cards.
- Figure 47 on page 176 shows a VSM4 with 4 ICE cards, 4 VCF cards.
- Figure 48 on page 177 shows a VSM4 with 2 ICE cards, 6 VCF cards.
- Figure 49 on page 177 shows a VSM4 with 8 VCF cards.

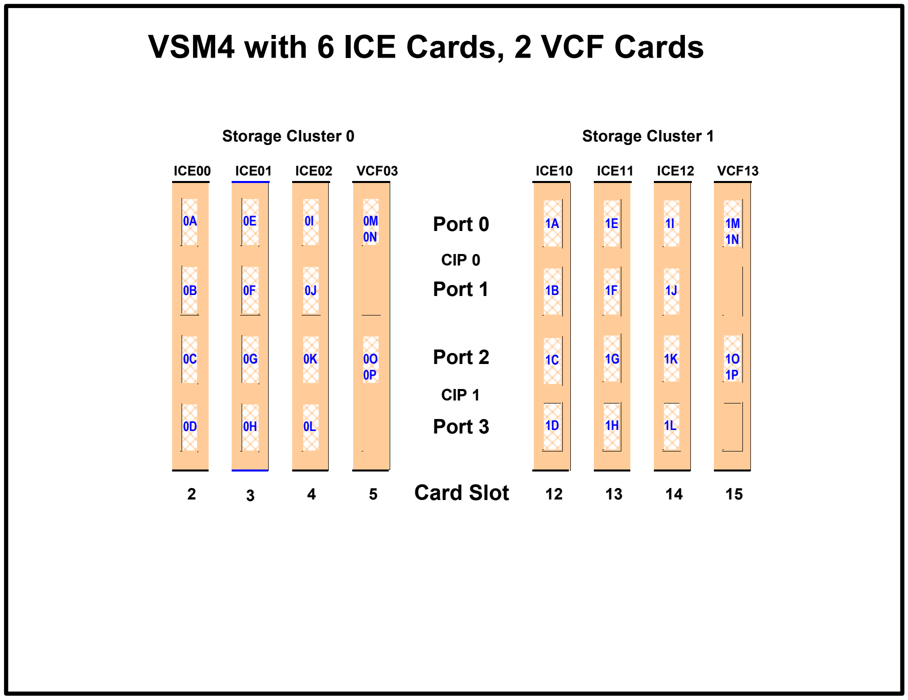


Figure 46. **VSM4 with 6 ICE cards, 2 VCF cards**

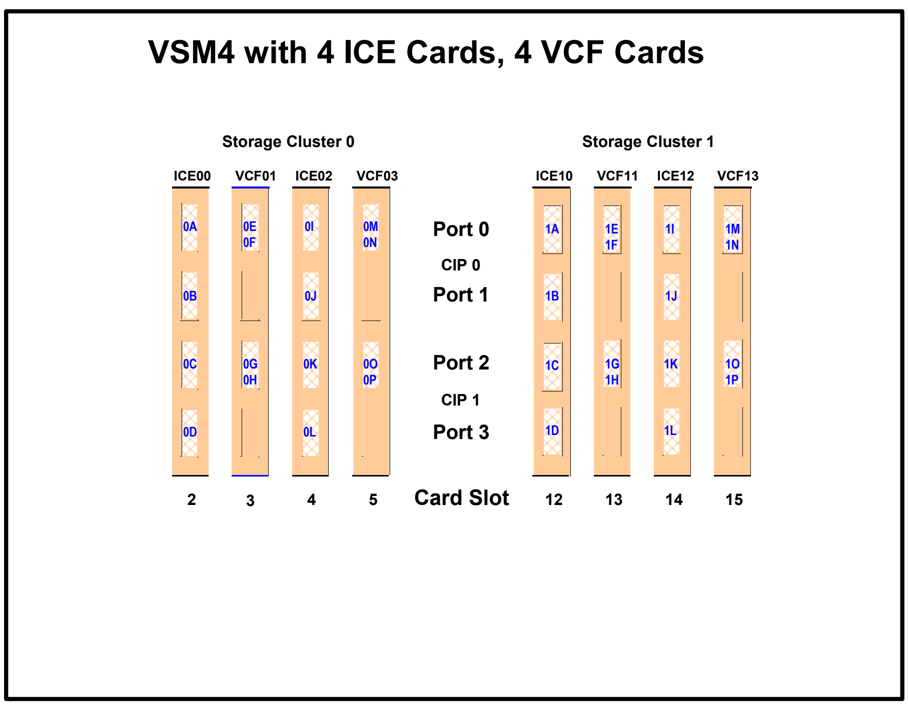


Figure 47. **VSM4 with 4 ICE cards, 4 VCF cards**

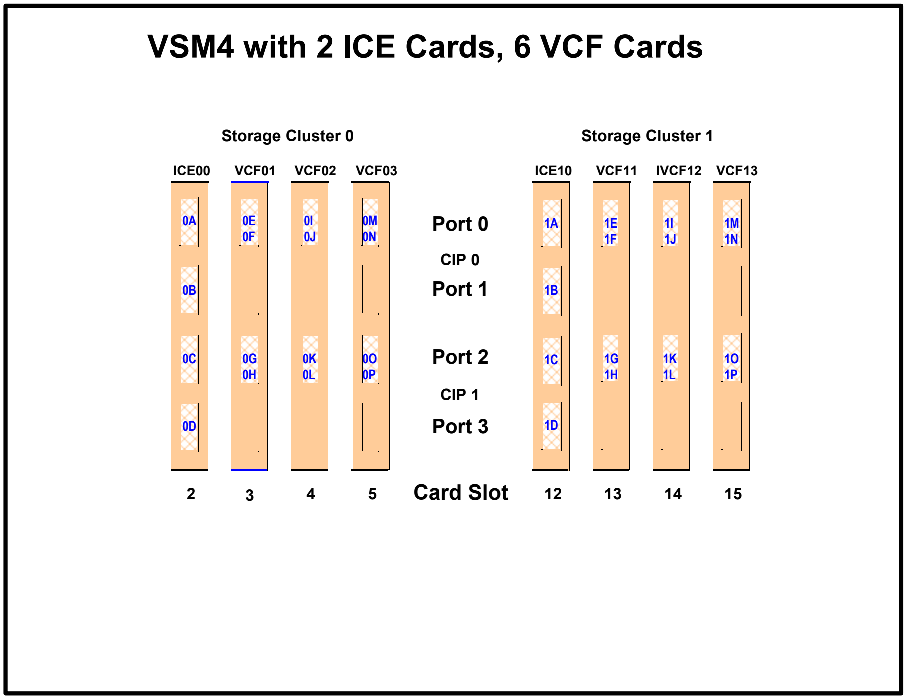


Figure 48. ***VSM4 with 2 ICE cards, 6 VCF cards***

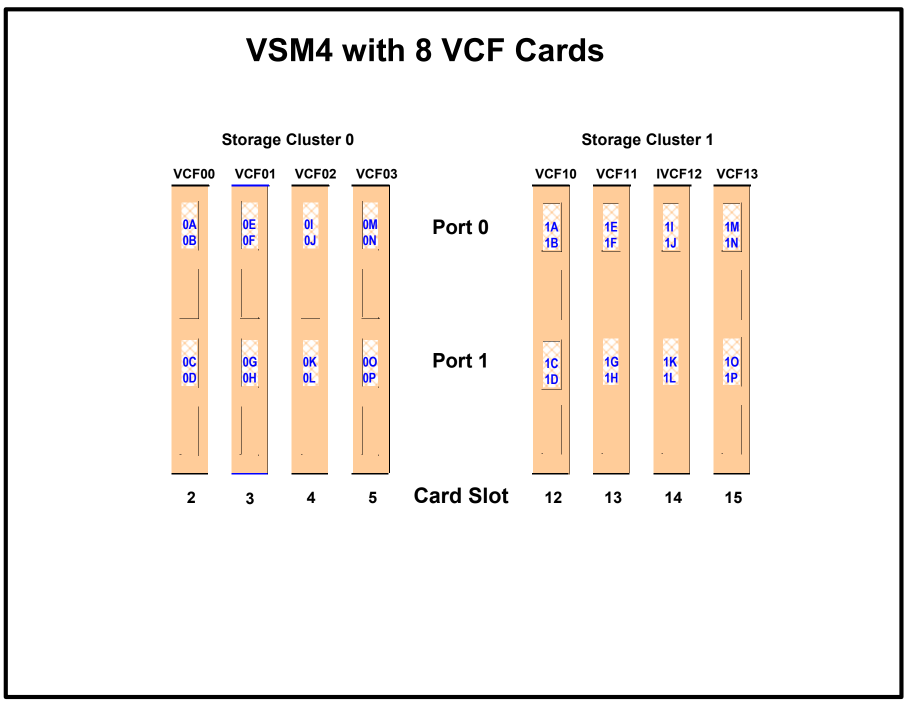


Figure 49. ***VSM4 with 8 VCF cards***

Note:

- In Figure 46 on page 176 through Figure 49 on page 177, the VCF cards must go in:
  - Slots 5 and 15 in a two-VCF card configuration
  - Slots 3, 5, 13, and 15 in a four-VCF card configuration.
  - Slots 3, 4, 5, 13, 14, and 15 in a six-VCF card configuration.
  - All slots in an eight-VCF card configuration.
- FICON ports are controlled by a FICON Interface processor (FIP), ESCON ports are controlled by a CIP. Regardless of the card configuration, there can be only a total of 8 Nearlink FIPs and/or CIPs.
- All FICON ports can be configured as either a Host port or Nearlink (RTD/CLINK origination) port. All ESCON ports continue to be configurable as host or Nearlink ports in pairs on a per CIP basis.
- As shown in Figure 46 on page 176 through Figure 49 on page 177, the ports are shown with their channel interface identifiers where **all ports are enabled**. These channel interface identifiers are the values that are required for the `CHANIF` values that you code for the `CONFIG` utility. Each value is two characters in length and has a value from 0A to 1O. The first digit is the VTSS cluster ID (valid values are 0 or 1). The second digit is the group or adapter ID (valid values are A to P).

  Each FICON port can attach to two RTDs, or two CLINKs, or an RTD/CLINK combination via a FICON director or supported switch (in FICON mode). **Note that**, as shown in these figures, **for RTDs only**, each FICON port has two `CHANIF` values **only if** the port is connected to a FICON director which is then connected to two RTDs. Nearlink RTD connections that are paired via a FICON switch or director on the same port dynamically alternate between both RTDs for atomic operations such as mount, migrate VTV, recall VTV, etc.
- **Each ICE card** contains two pairs of ESCON ports. Each pair is controlled by its own Channel Interface Processor (CIP). Each CIP switches between the two ports, so that **only one port** can transfer data at a time, which emulates a FICON port attached to a director attached to RTDs.
- Each host FICON channel supports 64 logical paths (times 16 logical units). However, in HCD:
  - From a single MVS host, you can only define 8 channels (CHPIDs) running to a single control unit (single VSM4).
  - You use the CNTLUNIT statement to define each VSM4 as 16 3490 control unit images.
  - You use the IODEVICE statement to define the 16 VTDs that are associated with each 3490 control unit image.

- For a VSM4, each ESCON CIP or FICON FIP can operate with only *one* of two “personalities”, which is set at the VTSS LOP:
  - *Host Mode*. In Host Mode, ports can connect to the host CPU channels, including via Director(s) or channel extenders. A port in Host Mode can also serve as a CLINK terminator.

    Also note that for ESCON ports, you can have two physical paths from the same LPAR to the same CIP, as long as the two physical paths address different (not overlapping) logical control units. For example, a single host LPAR can address logical control units 0-7 on one CIP port, and 8-F on the other CIP port of the same CIP.
  - *Nearlink Mode*. In Nearlink Mode, ports can connect to an RTD. A port in Nearlink Mode can also serve as a CLINK originator.
  - **For clustering**, you need an originator port in Nearlink mode on one VTSS connected via a CLINK to a terminator port in Host mode on the other VTSS.

    For example, Figure 50 shows 2 CLINK ports on each VTSS configured for Uni-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**.

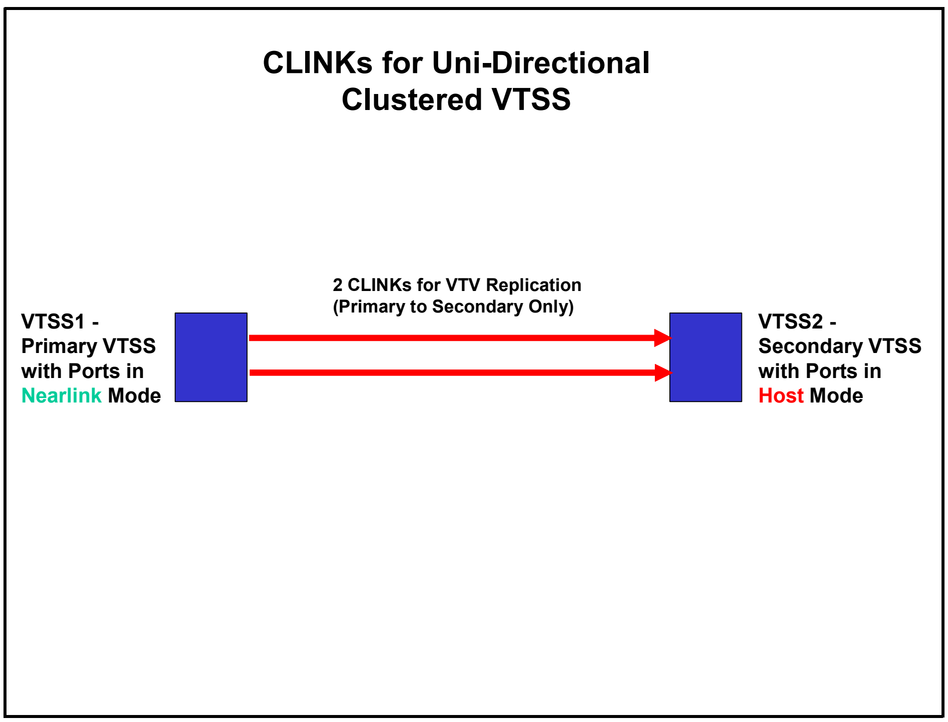


Figure 50. ***CLINKs for Uni-Directional Clustered VTSS***

Figure 51 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/FIPSs configured so that the data flows in **opposite directions** on the CLINKs.

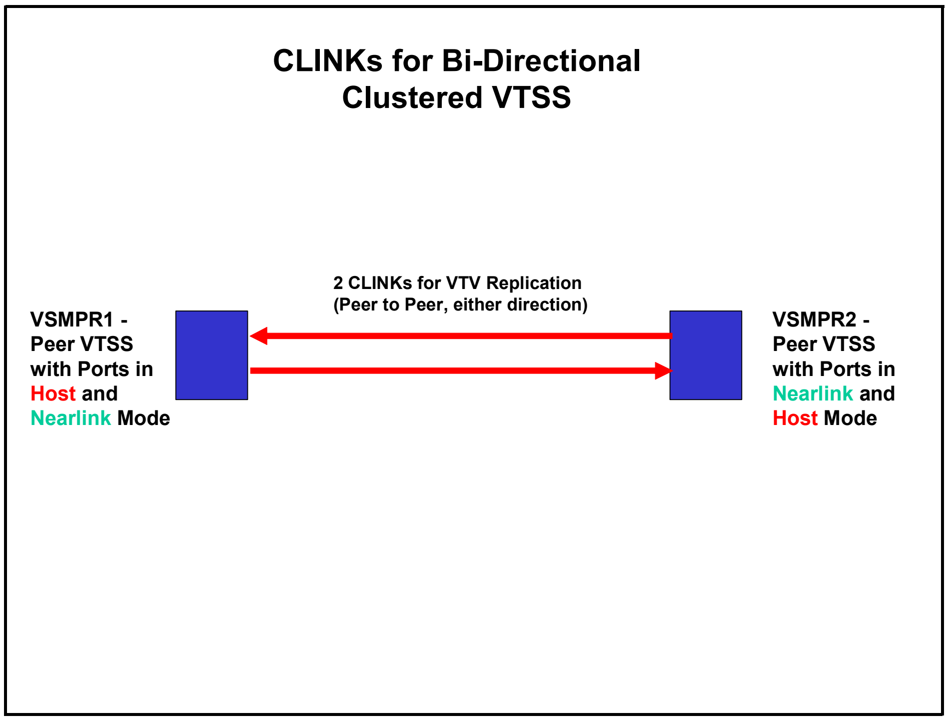


Figure 51. ***CLINKs for Bi-Directional Clustered VTSS***

In both FICON and ESCON, what are Best Practices for optimizing port operations? See Table 48...

Table 48. ***Optimizing VSM4 FICON/ESCON Port Operations***

Configuration - Two ESCON Ports on a CIP (ICE) or FICON port attached to a FICON Director (VCF)	**Best Practices**
Two CLINKs	**Don't use**....because only one port can be active at a time. If you're doing Clustered VTSS, you want all CLINK connections to be active all the time.
CLINK and RTD	**An advantage** in Degraded Cluster Mode. You normally have fewer RTDs on the Primary VTSS because the Secondary is doing most of the migrations. If you have an offline RTD on the same CIP as an active CLINK, if the Secondary fails you can vary the CLINK offline and bring the RTD online to handle more workload on the Primary.

Table 48. ***Optimizing VSM4 FICON/ESCON Port Operations***

Configuration - Two ESCON Ports on a CIP (ICE) or FICON port attached to a FICON Director (VCF)	**Best Practices**
Two RTDs	**An advantage** for the following: • **Optimize use of local and remote RTDs**. During busy shifts, use only the local RTD on the CIP. During quiet periods, switch to the remote RTD for deep archive and DR work. • **Optimize use of different drive technologies**. As described in the previous bullet, use a T9840 as a local RTD, then switch to a T9940 for deep archive. You can also use this feature to migrate from older drive technology (such as 9490) to newer technology (such as 9840). Use Management and Storage Classes to read in data from older media, then switch to the newer technology drive to place data on new media. This technique effectively gives you greater physical connectivity to different drive technologies without incurring the overhead of full time, real time ESCON connections to each drive type. **Note that** Because of the “only one active” rule, if an RTD on one port is migrating or recalling a VTV, the RTD on the second port cannot be accessed until the operation on the first port completes (the RTD on the second port is in “suspend” mode, as shown by the D RTD command/utility). Best Practices suggests, therefore, that RTDs that **must** be active simultaneously should connect to different CIPs.

VSM4 FICON Front-End and Back-End Configuration Examples

For VSM4s with both FICON Front-End and Back-End connectivity, let's look at two examples of VCF card configurations and implementation:

- "VSM4 Configuration Example: 8 VCF Cards, FICON Directors, 16 RTDs" on page 184
- "VSM4 Configuration Example: 8 VCF Cards, 4 CLINKs, FICON Directors for 8 RTDs" on page 186
- "IOCP Example for Single MVS Host Connected to a VSM4 Via FICON Directors" on page 189

For a VSM4 host gen example, see "IOCP Example for Single MVS Host Connected to a VSM4 Via FICON Directors" on page 189.

VSM4 Configuration Example: 8 VCF Cards, FICON Directors, 16 RTDs

Figure 52 shows CONFIG channel interface identifiers for a VSM4 with 8 VCF cards. In this configuration, we've allocated 8 ports to RTDs and 8 ports to host connections. 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 16 RTDs, although, as with ESCON, only one RTD per port/Director can be active at a time.

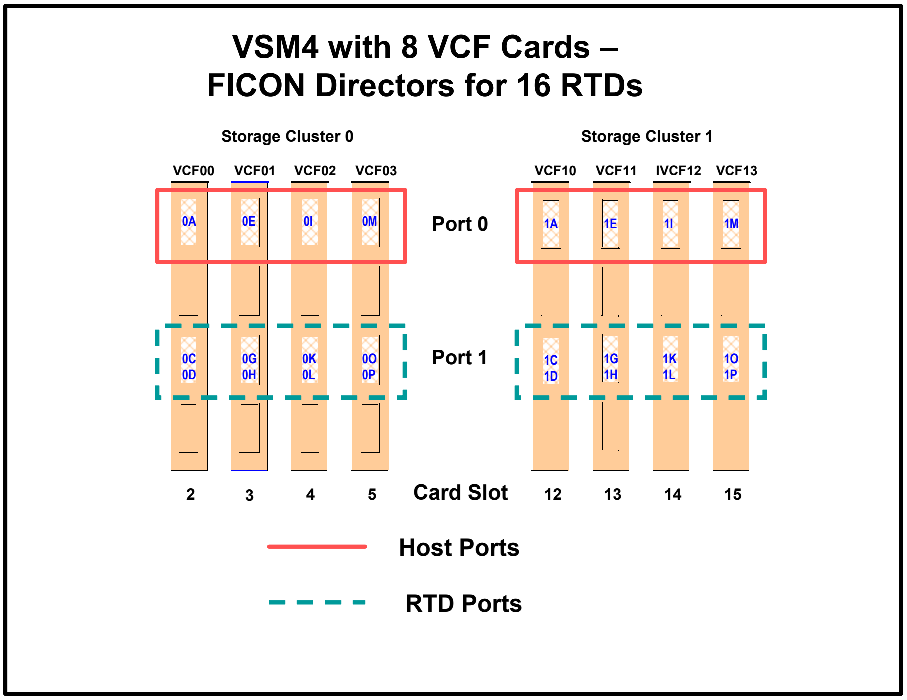


Figure 52. ***VSM4 with 8 VCF Cards, FICON Directors for 16 RTDs***

CONFIG Example for VSM4 FICON with 8 VCF Cards, FICON Directors, 16 RTDs

Figure 53 shows example CONFIG JCL to define the VSM4 configuration shown in Figure 52 on page 184.

```
//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
//SLSPRINT        DD   SYSOUT=*
//SLSIN           DD   *
 CONFIG
 GLOBAL           MAXVTV=32000   MVCFREE=40
 RECLAIM          THRESHLD=70    MAXMVC=40  START=35
 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=VSM401 LOW=70 HIGH=80 MAXMIG=8 RETAIN=5
 RTD  NAME=VSM42A00 DEVNO=2A00 CHANIF=0C
 RTD  NAME=VSM42A01 DEVNO=2A01 CHANIF=0D
 RTD  NAME=VSM42A02 DEVNO=2A02 CHANIF=0G
 RTD  NAME=VSM42A03 DEVNO=2A03 CHANIF=0H
 RTD  NAME=VSM42A04 DEVNO=2A04 CHANIF=0K
 RTD  NAME=VSM42A05 DEVNO=2A05 CHANIF=0L
 RTD  NAME=VSM42A06 DEVNO=2A06 CHANIF=0O
 RTD  NAME=VSM42A07 DEVNO=2A07 CHANIF=0P
 RTD  NAME=VSM42A08 DEVNO=2A08 CHANIF=1C
 RTD  NAME=VSM42A09 DEVNO=2A09 CHANIF=1D
 RTD  NAME=VSM42A0A DEVNO=2A0A CHANIF=1G
 RTD  NAME=VSM42A0B DEVNO=2A0B CHANIF=1H
 RTD  NAME=VSM42A0C DEVNO=2A0C CHANIF=1K
 RTD  NAME=VSM42A0D DEVNO=2A0D CHANIF=1L
 RTD  NAME=VSM42A0E DEVNO=2A0E CHANIF=1O
 RTD  NAME=VSM42A0F DEVNO=2A0F CHANIF=1P
 VTD LOW=9900 HIGH=99FF
```

Figure 53. `CONFIG` ***example: VSM4 with 8 VCF cards, FICON Directors, 16 RTDs***

VSM4 Configuration Example: 8 VCF Cards, 4 CLINKs, FICON Directors for 8 RTDs

Figure 52 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. As with ESCON, only one RTD per port/Director can be active at a time.
- 4 ports for CLINK connections to form a Bi-Directional VTSS Cluster, and 8 ports to host connections. To form the clustered VTSS, we'll have two VSM4s (`VSMPR1` and `VSMPR2`) configured identically as shown in Figure 52. As shown in Figure 51 on page 180, Bi-Directional Clustering requires pairs of Uni-Directional CLINKs with the FIPSs configured so that the data flows in **opposite directions** on the CLINKs. To make that happen, let's make 0G and 1G the sending (Nearlink Mode) ports on both VTSSs and 0O and 1O the receiving (Host Mode) on both VTSSs.

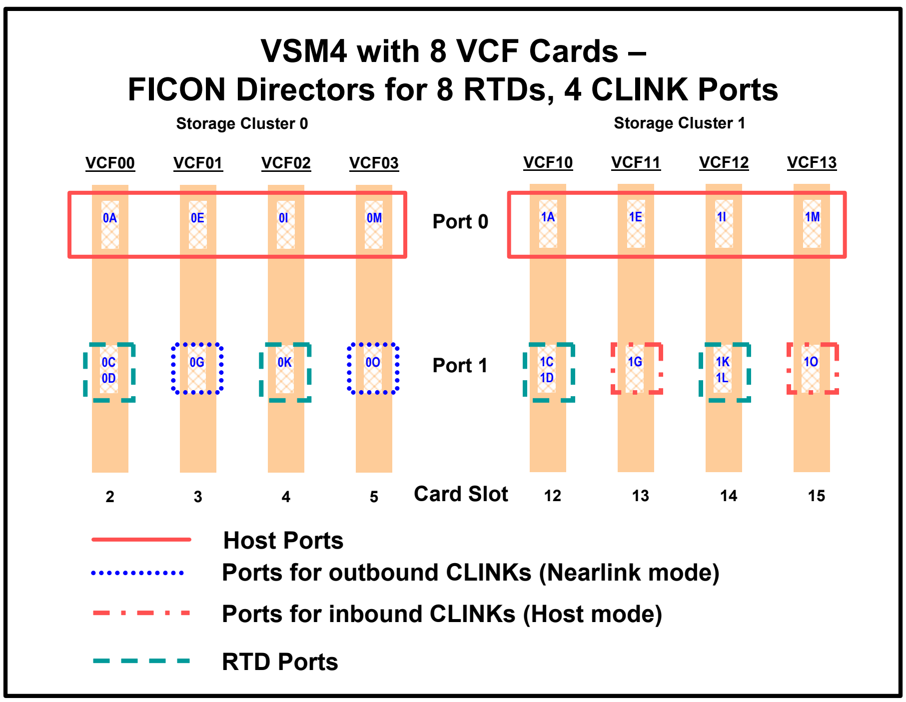


Figure 54. ***VSM4 with 8 VCF Cards, 8 Host Ports, FICON Directors for 8 RTDs, 4 CLINK Ports***

CONFIG Example for Bi-Directional Clustered VSM4 FICON Back-End

Figure 55 shows example CONFIG JCL to define a Bi-Directional Cluster of two VSM4s (VSMPR1 and VSMPR2) with identical VCF card configurations shown in Figure 54 on page 186.

Caution: Bi-Directional Clustering **requires** VTCS 6.1! You **cannot** configure a Bi-Directional Cluster at releases lower than VTCS 6.1! **Also note** that the Clustered VTSSs require the Advanced Management Feature.

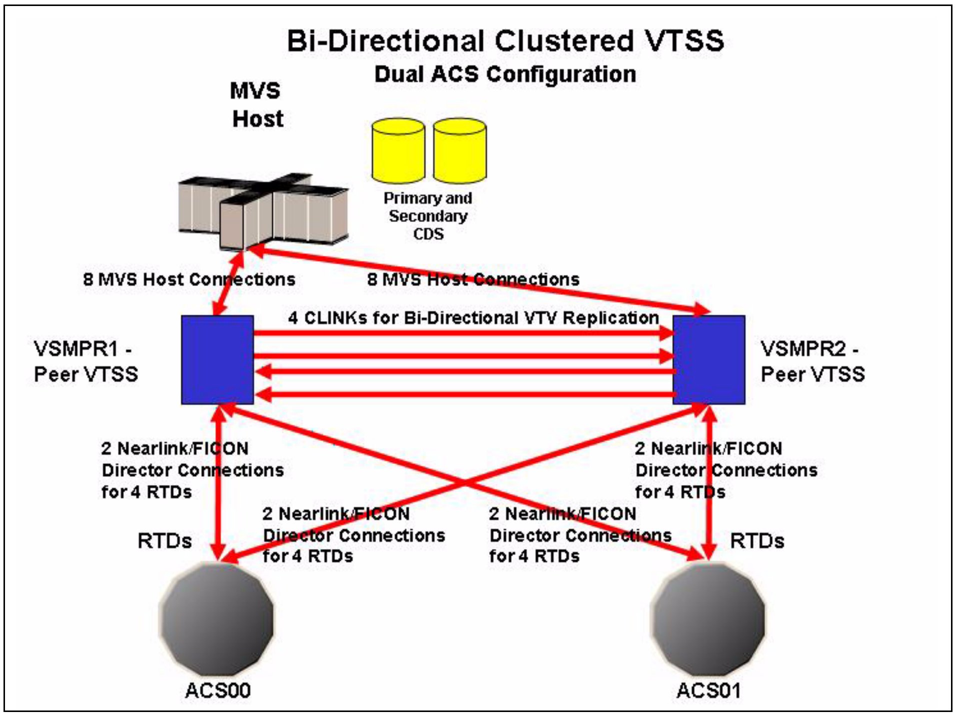


Figure 55. ***Dual ACS Bi-Directional Clustered VTSS Configuration***

Figure 56 shows example CONFIG JCL to define a Bi-Directional Cluster of two VSM4s (VSMPR1 and VSMPR2) as shown in Figure 55 on page 187. **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 186, the Nearlink ports are 0G and 1G on both VTSSs.

```
//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
//SLSPRINT      DD   SYSOUT=*
//SLSIN         DD   *
 CONFIG RESET CDSLEVEL(V61ABOVE)
 GLOBAL    MAXVTV=32000     MVCFREE=40
 RECLAIM   THRESHLD=70      MAXMVC=40  START=35
 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 56. CONFIG ***example: Dual ACS Bi-Directional Clustered VTSS System, VSM4 FICON Back-End***

IOCP Example for Single MVS Host Connected to a VSM4 Via FICON Directors

Figure 57 shows a configuration diagram for a single MVS host connected to a VSM4 via FICON Directors, and Figure 58 on page 190 shows example IOCP statements for this configuration. **Note that**:

- From MVSA, you define 8 CHPIDs, with each path switched in the FICON Director, for a total of 8 channels running to the VSM4.
- You code 16 CNTLUNIT statements to define the VSM4 as 16 3490 images.
- You code IODEVICE statement to define the 16 VTDs that are associated with each 3490 image.
- If ESCON and FICON channels are configured to the same logical control unit, MVS issues message CBDG489I, which indicates that mixing ESCON and FICON channel paths on a logical control unit should be used only for the migration from ESCON to native FICON, but should not be used permanently. This is a warning message only, and does not indicate an error.

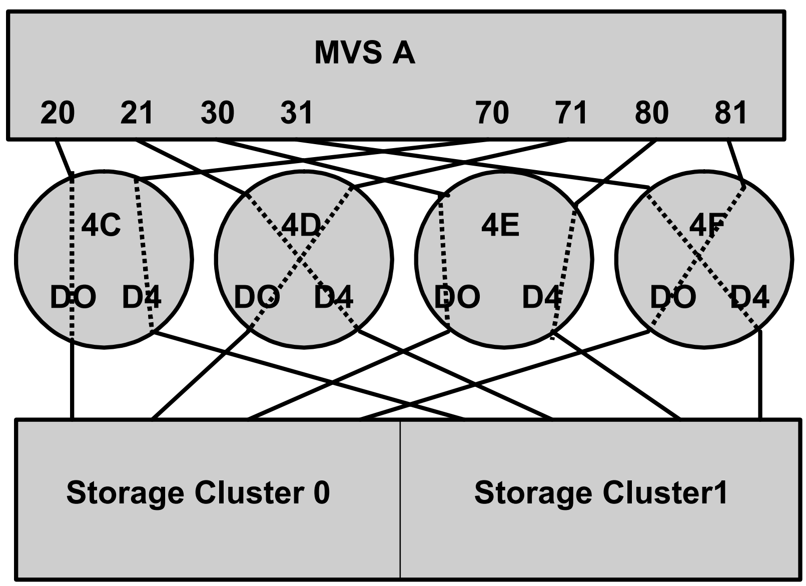


Figure 57. ***Configuration Diagram: Single MVS Host Connected to a VSM4 via FICON Directors***

```
ESCD4C   CHPID PATH=(20,70),TYPE=FC,SWITCH=4C
ESCD4D   CHPID PATH=(21,71),TYPE=FC,SWITCH=4D
ESCD4E   CHPID PATH=(30,80),TYPE=FC,SWITCH=4E
ESCD4F   CHPID PATH=(31,81),TYPE=FC,SWITCH=4F

CU1      CNTLUNIT CUNUMBR=001,
                  PATH=(20,21,30,31,70,71,80,81),
                  LINK=(D0,D4,D0,D4,D4,D0,D4,D0),
                  UNIT=3490,CUADD=0,
                  UNITADD=((00,16))

STRING1  IODEVICE ADDRESS=(0500,16),
                  CUNUMBER=(001),
                  UNIT=3490,
                  UNITADD=00,STADET=Y

CU2      CNTLUNIT CUNUMBR=002,
                  PATH=(20,21,30,31,70,71,80,81),
                  LINK=(D0,D4,D0,D4,D4,D0,D4,D0),
                  UNIT=3490,CUADD=1,
                  UNITADD=((00,16))

STRING2  IODEVICE ADDRESS=(0510,16),
                  CUNUMBER=(002),
                  UNIT=3490,
                  UNITADD=00,STADET=Y
                  .
                  .
                  .
CU15     CNTLUNIT CUNUMBR=015,
                  PATH=(20,21,30,31,70,71,80,81),
                  LINK=(D0,D4,D0,D4,D4,D0,D4,D0),
                  UNIT=3490,CUADD=E,
                  UNITADD=((00,16))

STRING15  IODEVICE ADDRESS=(05E0,16),
                  CUNUMBER=(015),
                  UNIT=3490,
                  UNITADD=00,STADET=Y

CU16     CNTLUNIT CUNUMBR=016,
                  PATH=(20,21,30,31,70,71,80,81),
                  LINK=(D0,D4,D0,D4,D4,D0,D4,D0),
                  UNIT=3490,CUADD=F,
                  UNITADD=((00,16))

STRING16  IODEVICE ADDRESS=(05F0,16),
                  CUNUMBER=(016),
                  UNIT=3490,
                  UNITADD=00,STADET=Y
```

Figure 58. ***IOCP Example: Single MVS Host Connected to a VSM4 via FICON Directors***

Hint: Unlike ESCON, FICON supports multiple active I/Os per channel. If the number of active VTDs is less than the number of channels configured to the VTSS, the I/Os to those VTDs may not be evenly spread across all the channels. As the number of active VTDs increases to be greater than the number of channels configured to the VTSS, the channel subsystem will spread the I/Os across all the channels. If it is desired to spread the I/Os across all of the channels even when only a few VTDs are active, it is necessary to use the preferred path feature to force the channel subsystem to spread the I/Os across the channels. The preferred path feature is specified via the `PATH=` parameter on the `IODEVICE` statement. When you specify preferred path on the `IODEVICE` statement, the channel subsystem always tries the preferred path first. If it is busy or unavailable, the channel subsystem next tries the channel path following the preferred path in the rotation order, and so on.

Figure 58 on page 190 (repeated in Figure 59) shows `IODEVICE` statements for `STRING1` **without** using preferred pathing.

```
STRING1  IODEVICE ADDRESS=(0500,16),
                  CUNUMBER=(001),
                  UNIT=3490,
                  UNITADD=00,STADET=Y
```

*Figure 59. **IODEVICE Statements for STRING 1 without Preferred Pathing***

Figure 60 on page 192 shows `IODEVICE` statements for `STRING1` **using** preferred pathing. If you're using preferred pathing, you need to use these kind of `IODEVICE` statements for **all** paths, such as `STRING2` through `STRING16` in Figure 58 on page 190.

```
STRING10 IODEVICE ADDRESS=(0500,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=20

STRING12 IODEVICE ADDRESS=(0502,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=21

STRING14 IODEVICE ADDRESS=(0504,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=30

STRING16 IODEVICE ADDRESS=(0506,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=31

STRING18 IODEVICE ADDRESS=(0508,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=70

STRING1A IODEVICE ADDRESS=(050A,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=71

STRING1C IODEVICE ADDRESS=(050C,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=80

STRING1E IODEVICE ADDRESS=(050E,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=81
```

Figure 60. ***IODEVICE Statements for STRING 1 Using Preferred Pathing***

Appendix F. VSM5 Configuration

VSM5, which provides greater throughput and capacity than its predecessors, is a significant step forward for the VSM solution. The following RTDs are supported (using 3490-emulation mode only): 9840B, 9840C, 9940B, T10000.

Note: PTF L1H12ZT (SWS6100) changes `QUERY/DISPLAY VTSS` to report the VTSS capacity in gigabytes (Gb).

VSM5 FICON VCF Card Options

VSM5 is available **only** with VCF (FICON) cards in the following configurations:

- Figure 61 shows a VSM5 with 8 VCF cards.
- Figure 62 on page 195 shows a VSM5 with 6 VCF cards, 2 empty card slots.
- Figure 63 on page 195 shows a VSM5 with 4 VCF cards, 4 empty card slots.

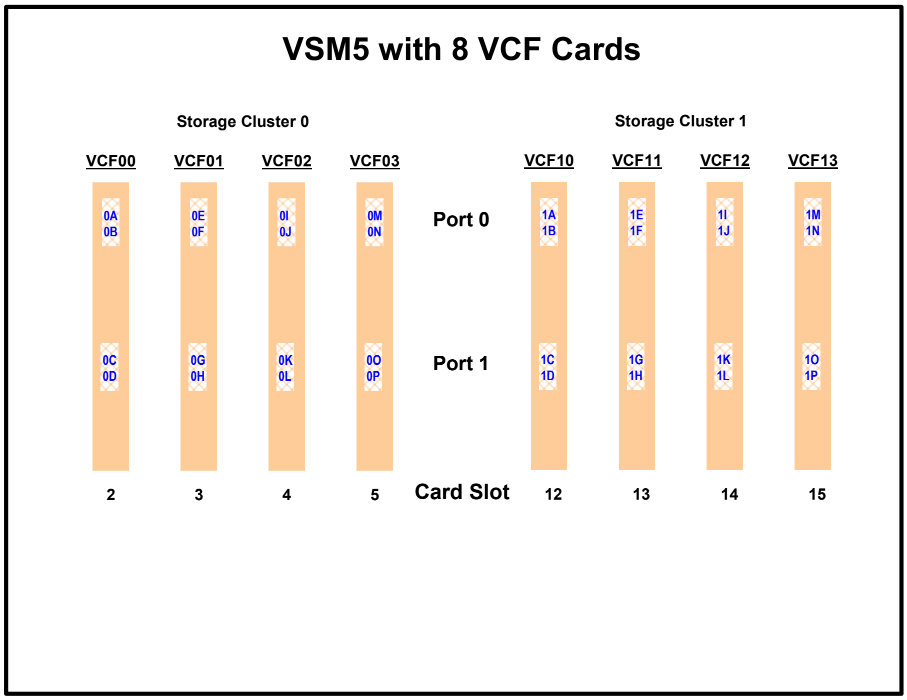


Figure 61. ***VSM5 with 8 VCF cards***

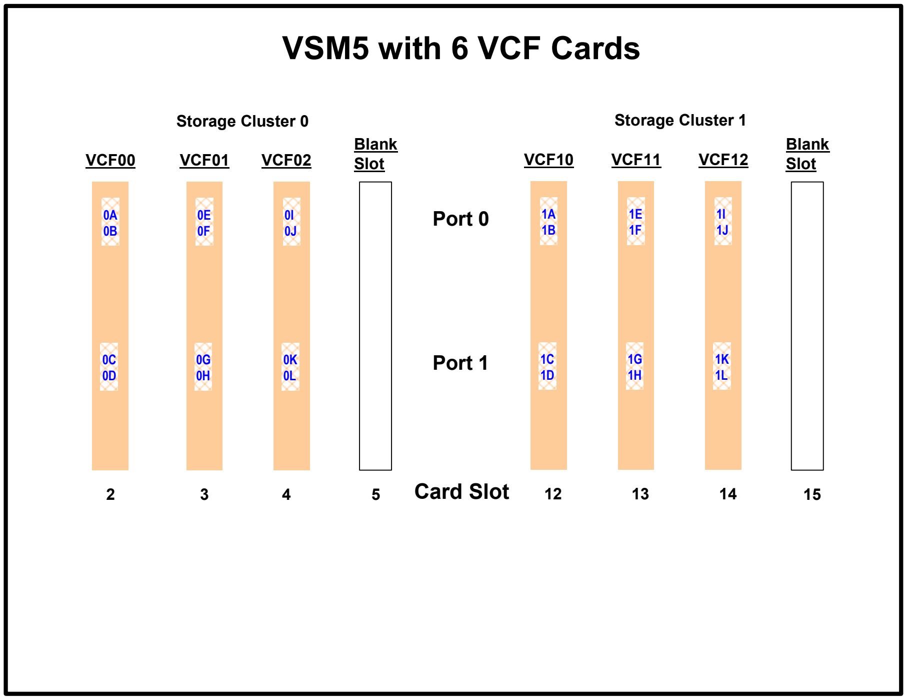


Figure 62. ***VSM5 with 6 VCF cards, 2 empty card slots***

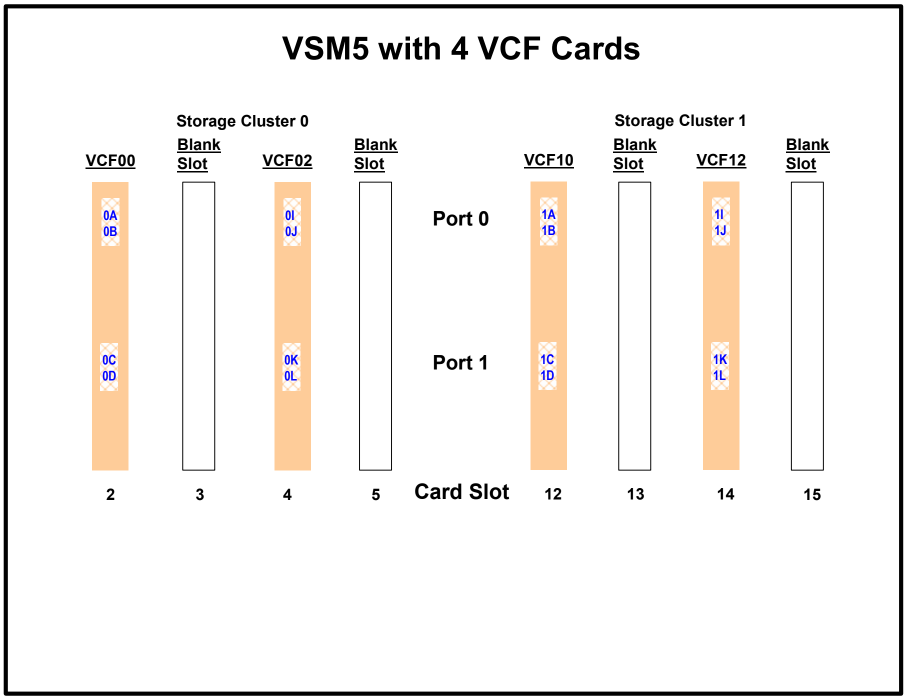


Figure 63. ***VSM5 with 4 VCF cards, 4 empty card slots***

Note:

- In Figure 61 on page 194 through Figure 63 on page 195, the VCF cards must go in:
  - All slots in an eight-VCF card configuration.
  - Slots 2, 3, 4, 13, 14, and 15 in a six-VCF card configuration.
  - Slots 2, 4, 14, and 15 in a four-VCF card configuration.
- FICON ports are controlled by a FICON Interface processor (FIP) and there can be only a total of 8 Nearlink FIPs.
- All FICON ports can be configured as either a Host port or Nearlink (RTD/CLINK origination) port.
- In Figure 61 on page 194 through Figure 63 on page 195, the ports are shown with their channel interface identifiers where **all ports are enabled**. These channel interface identifiers are the values that are required for the `CHANIF` values that you code for the `CONFIG` utility. Each value is two characters in length and has a value from 0A to 1O. The first digit is the VTSS cluster ID (valid values are 0 or 1). The second digit is the group or adapter ID (valid values are A to P).

  Each FICON port can attach to two RTDs, or two CLINKs, or an RTD/CLINK combination via a FICON director or supported switch (in FICON mode). **Note that**, as shown in these figures, **for RTDs only**, each FICON port has two `CHANIF` values **only if** the port is connected to a FICON director which is then connected to two RTDs. Nearlink RTD connections that are paired via a FICON switch or director on the same port dynamically alternate between both RTDs for atomic operations such as mount, migrate VTV, recall VTV, etc.
- Each host FICON channel supports 64 logical paths (times 16 logical units). However, in HCD:
  - From a single MVS host, you can only define 8 channels (CHPIDs) running to a single control unit (single VSM5).
  - You use the CNTLUNIT statement to define each VSM5 as 16 3490 control unit images.
  - You use the IODEVICE statement to define the 16 VTDs that are associated with each 3490 control unit image.

- For a VSM5, each FIP can operate with only *one* of two “personalities”, which is set at the VTSS DOP:
    - *Host Mode*. In Host Mode, ports can connect to the host CPU channels, including via Director(s) or channel extenders. A port in Host Mode can also serve as a CLINK terminator.
    - *Nearlink Mode*. In Nearlink Mode, ports can connect to an RTD. A port in Nearlink Mode can also serve as a CLINK originator.
    - **For clustering**, you need an originator port in Nearlink mode on one VTSS connected via a CLINK to a terminator port in Host mode on the other VTSS.

      For example, Figure 64 shows 2 CLINK ports on each VTSS configured for Uni-Directional Clustering. On the Primary VTSS (VTSS1), the CLINK FIPs are configured in **Nearlink Mode**, while on the Secondary VTSS (VTSS2), the FIPs are configured in **Host Mode**.

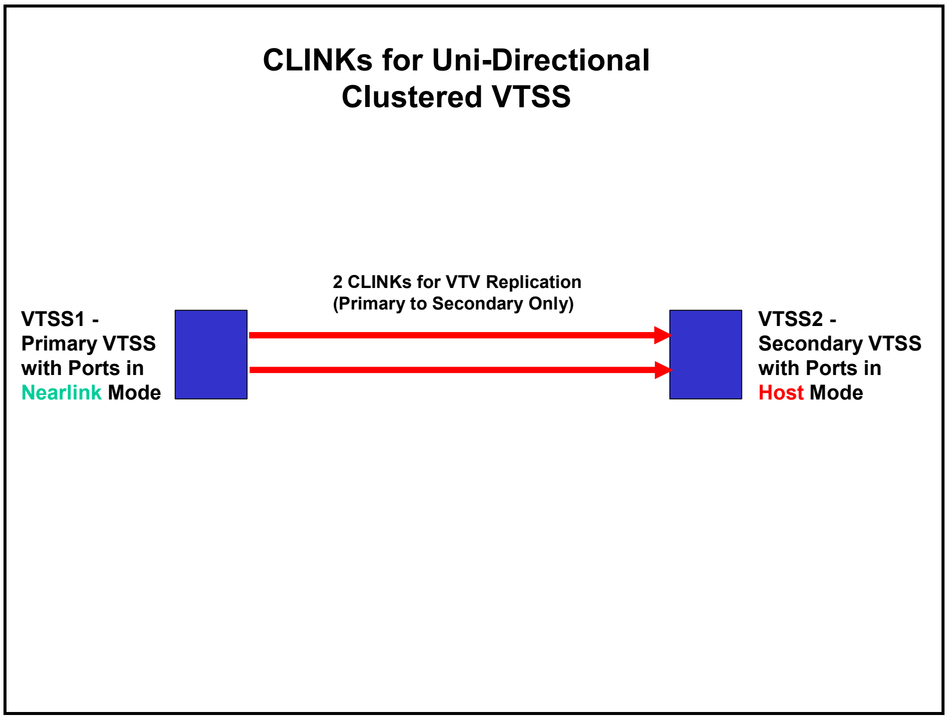


Figure 64. CLINKs for Uni-Directional Clustered VTSS

Figure 65 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 FIP configured in **Nearlink Mode** for replicating to the Peer.
- **One** CLINK FIP configured in **Host Mode** for receiving replicated VTVs from the Peer.

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

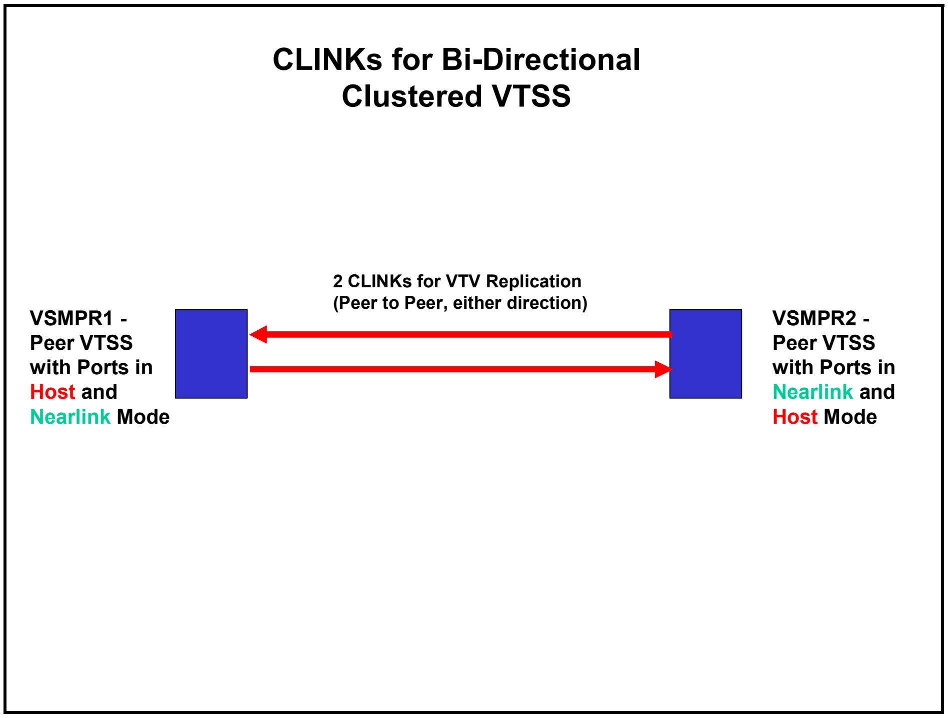


Figure 65. ***CLINKs for Bi-Directional Clustered VTSS***

For FICON, what are Best Practices for optimizing port operations? See Table 49...

Table 49. ***Optimizing VSM5 FICON Port Operations***

Configuration - FICON port attached to a FICON Director	**Best Practices**
Two CLINKs	**Don't use**....because only one port can be active at a time. If you're doing Clustered VTSS, you want all CLINK connections to be active all the time.
CLINK and RTD	**An advantage** in Degraded Cluster Mode. You normally have fewer RTDs on the Primary VTSS because the Secondary is doing most of the migrations. If you have an offline RTD on the same FIP as an active CLINK, if the Secondary fails you can vary the CLINK offline and bring the RTD online to handle more workload on the Primary.

Table 49. ***Optimizing VSM5 FICON Port Operations***

Configuration - FICON port attached to a FICON Director	**Best Practices**
Two RTDs	**An advantage** for the following: • **Optimize use of local and remote RTDs**. During busy shifts, use only the local RTD on the FIP. During quiet periods, switch to the remote RTD for deep archive and DR work. • **Optimize use of different drive technologies**. As described in the previous bullet, use a T9840 as a local RTD, then switch to a T9940 for deep archive. You can also use this feature to migrate from older drive technology (such as 9490) to newer technology (such as 9840). Use Management and Storage Classes to read in data from older media, then switch to the newer technology drive to place data on new media. This technique effectively gives you greater physical connectivity to different drive technologies without incurring the overhead of full time, real time FICON connections to each drive type. **Note that** Because of the “only one active” rule, if an RTD on one port is migrating or recalling a VTV, the RTD on the second port cannot be accessed until the operation on the first port completes (the RTD on the second port is in “suspend” mode, as shown by the D RTD command/utility). Best Practices suggests, therefore, that RTDs that **must** be active simultaneously should connect to different FIPs.

VSM5 FICON Front-End and Back-End Configuration Examples

For VSM5s, let's look at two examples of VCF card configurations and implementation:

- "VSM5 Configuration Example: 8 VCF Cards, FICON Directors, 16 RTDs" on page 202
- "VSM5 Configuration Example: 8 VCF Cards, 4 CLINKs, FICON Directors for 8 RTDs" on page 204
- "IOCP Example for Single MVS Host Connected to a VSM5 Via FICON Directors" on page 207

For a VSM5 host gen example, see "IOCP Example for Single MVS Host Connected to a VSM5 Via FICON Directors" on page 207.

VSM5 Configuration Example: 8 VCF Cards, FICON Directors, 16 RTDs

Figure 66 shows CONFIG channel interface identifiers for a VSM5 with 8 VCF cards. In this configuration, we've allocated 8 ports to RTDs and 8 ports to host connections. The RTD ports are all connected to FICON directors, each of which is attached to 2 RTDs, so the CHANIF identifiers for both RTDs are shown on each port. This allows Back-End connection to 16 RTDs, although only one RTD per port/Director can be active at a time.

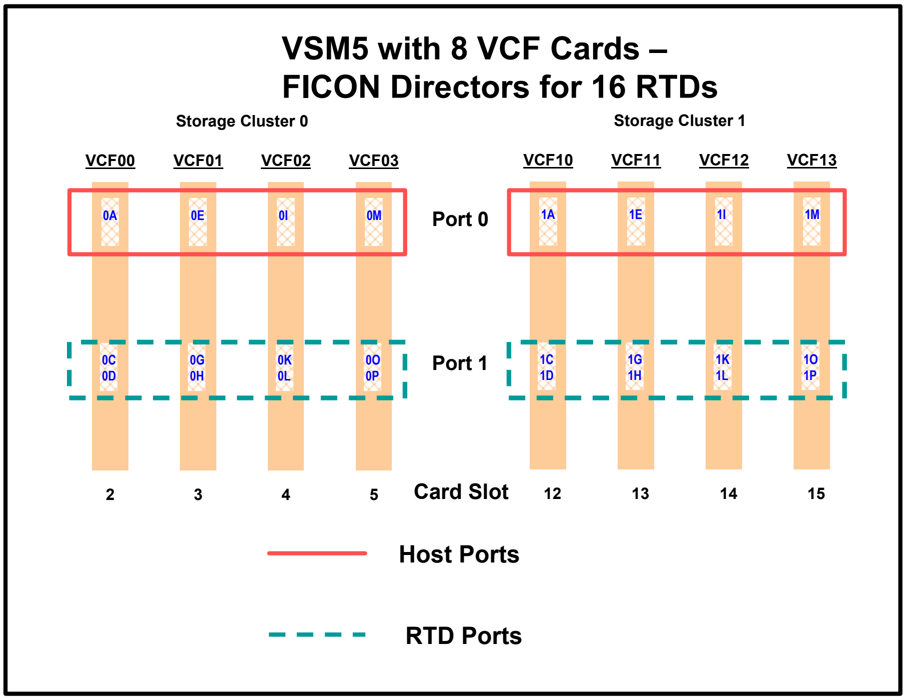


Figure 66. ***VSM5 with 8 VCF Cards, FICON Directors for 16 RTDs***

CONFIG Example for VSM5 FICON with 8 VCF Cards, FICON Directors, 16 RTDs

Figure 67 shows example CONFIG JCL to define the VSM5 configuration shown in Figure 66 on page 202.

```
//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
//SLSPRINT        DD   SYSOUT=*
//SLSIN           DD   *
 CONFIG
 GLOBAL           MAXVTV=32000   MVCFREE=40
 RECLAIM          THRESHLD=70    MAXMVC=40  START=35
 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=VSM501 LOW=70 HIGH=80 MAXMIG=8 RETAIN=5
 RTD  NAME=VSM52A00 DEVNO=2A00 CHANIF=0C
 RTD  NAME=VSM52A01 DEVNO=2A01 CHANIF=0D
 RTD  NAME=VSM52A02 DEVNO=2A02 CHANIF=0G
 RTD  NAME=VSM52A03 DEVNO=2A03 CHANIF=0H
 RTD  NAME=VSM52A04 DEVNO=2A04 CHANIF=0K
 RTD  NAME=VSM52A05 DEVNO=2A05 CHANIF=0L
 RTD  NAME=VSM52A06 DEVNO=2A06 CHANIF=0O
 RTD  NAME=VSM52A07 DEVNO=2A07 CHANIF=0P
 RTD  NAME=VSM52A08 DEVNO=2A08 CHANIF=1C
 RTD  NAME=VSM52A09 DEVNO=2A09 CHANIF=1D
 RTD  NAME=VSM52A0A DEVNO=2A0A CHANIF=1G
 RTD  NAME=VSM52A0B DEVNO=2A0B CHANIF=1H
 RTD  NAME=VSM52A0C DEVNO=2A0C CHANIF=1K
 RTD  NAME=VSM52A0D DEVNO=2A0D CHANIF=1L
 RTD  NAME=VSM52A0E DEVNO=2A0E CHANIF=1O
 RTD  NAME=VSM52A0F DEVNO=2A0F CHANIF=1P
 VTD LOW=9900 HIGH=99FF
```

Figure 67. `CONFIG` ***example: VSM5 with 8 VCF cards, FICON Directors, 16 RTDs***

VSM5 Configuration Example: 8 VCF Cards, 4 CLINKs, FICON Directors for 8 RTDs

Figure 66 shows CONFIG channel interface identifiers for a VSM5 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 Bi-Directional VTSS Cluster, and 8 ports to host connections. To form the clustered VTSS, we'll have two VSM5s (VSMPR1 and VSMPR2) configured identically as shown in Figure 66. As shown in Figure 65 on page 198, Bi-Directional Clustering requires pairs of Uni-Directional CLINKs with the FIPSs configured so that the data flows in **opposite directions** on the CLINKs. To make that happen, let's make 0G and 1G the sending (Nearlink Mode) ports on both VTSSs and 0O and 1O the receiving (Host Mode) on both VTSSs.

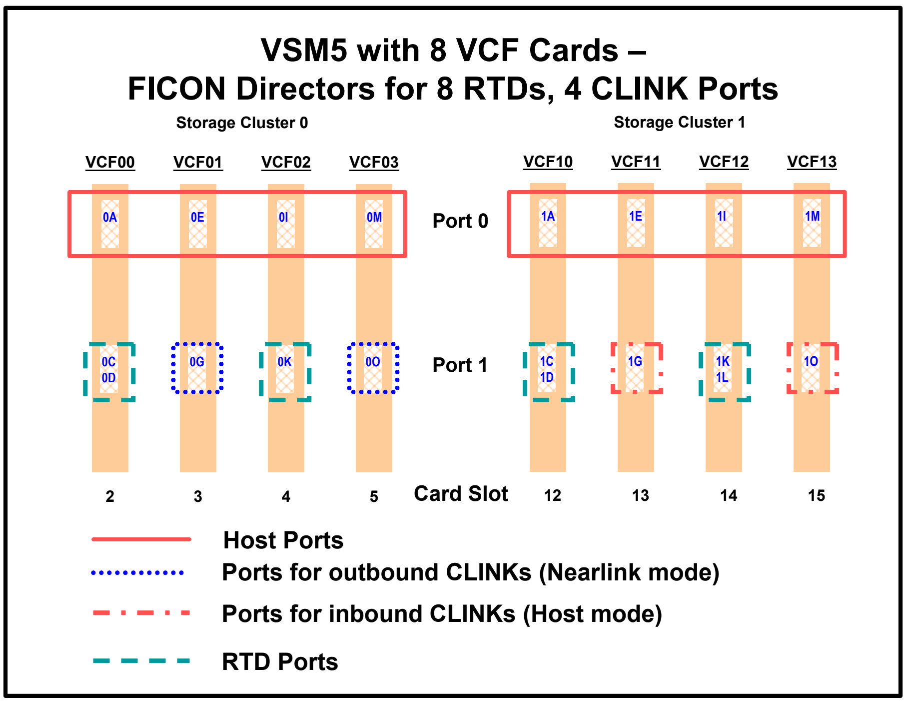


Figure 68. ***VSM5 with 8 VCF Cards, 8 Host Ports, FICON Directors for 8 RTDs, 4 CLINK Ports***

CONFIG Example for Bi-Directional Clustered VSM5 FICON Back-End

Figure 69 shows example CONFIG JCL to define a Bi-Directional Cluster of two VSM5s (VSMPR1 and VSMPR2) with identical VCF card configurations shown in Figure 68 on page 204.

Caution: Bi-Directional Clustering **requires** VTCS 6.1! You **cannot** configure a Bi-Directional Cluster at releases lower than VTCS 6.1! **Also note** that the Clustered VTSSs require the Advanced Management Feature.

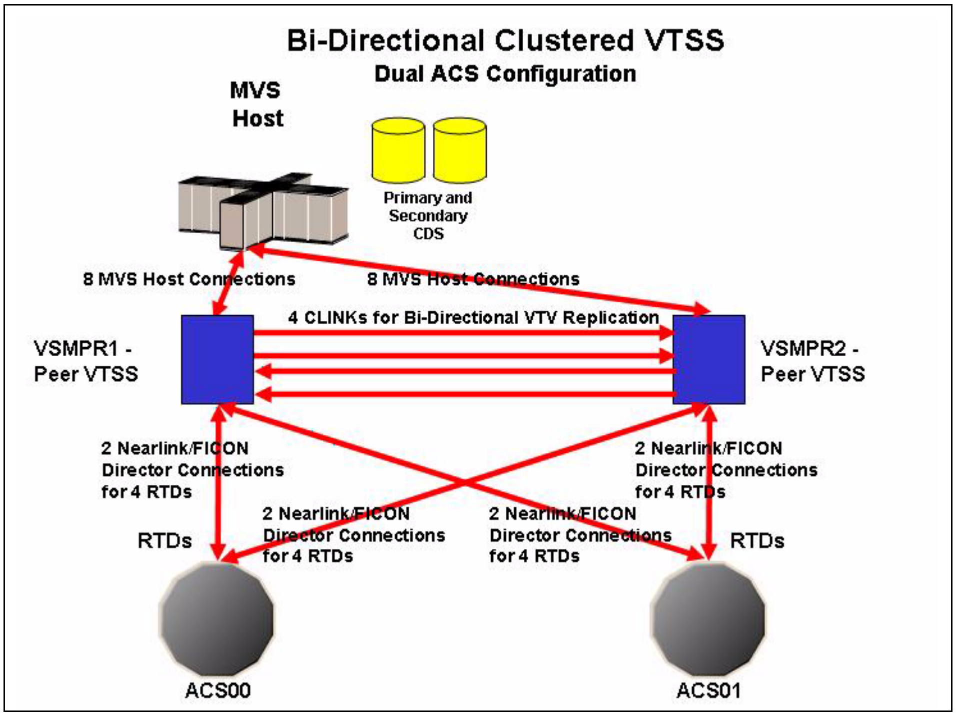


Figure 69. ***Dual ACS Bi-Directional Clustered VTSS Configuration***

Figure 70 shows example CONFIG JCL to define a Bi-Directional Cluster of two VSM5s (VSMPR1 and VSMPR2) as shown in Figure 69 on page 205. **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 204, the Nearlink ports are 0G and 1G on both VTSSs.

```
//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
//SLSPRINT      DD   SYSOUT=*
//SLSIN         DD   *
 CONFIG RESET CDSLEVEL(V61ABOVE)
 GLOBAL    MAXVTV=32000     MVCFREE=40
 RECLAIM   THRESHLD=70      MAXMVC=40  START=35
 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 70. CONFIG ***example: Dual ACS Bi-Directional Clustered VTSS System, VSM5 FICON Back-End***

IOCP Example for Single MVS Host Connected to a VSM5 Via FICON Directors

Figure 71 shows a configuration diagram for a single MVS host connected to a VSM5 via FICON Directors, and Figure 72 on page 208 shows example IOCP statements for this configuration. **Note that**:

- From MVSA, you define 8 CHPIDs, with each path switched in the FICON Director, for a total of 8 channels running to the VSM5.
- You code 16 CNTLUNIT statements to define the VSM5 as 16 3490 images.
- You code IODEVICE statement to define the 16 VTDs that are associated with each 3490 image.
- If ESCON and FICON channels are configured to the same logical control unit, MVS issues message `CBDG489I`, which indicates that mixing ESCON and FICON channel paths on a logical control unit should be used only for the migration from ESCON to native FICON, but should not be used permanently. This is a warning message only, and does not indicate an error.

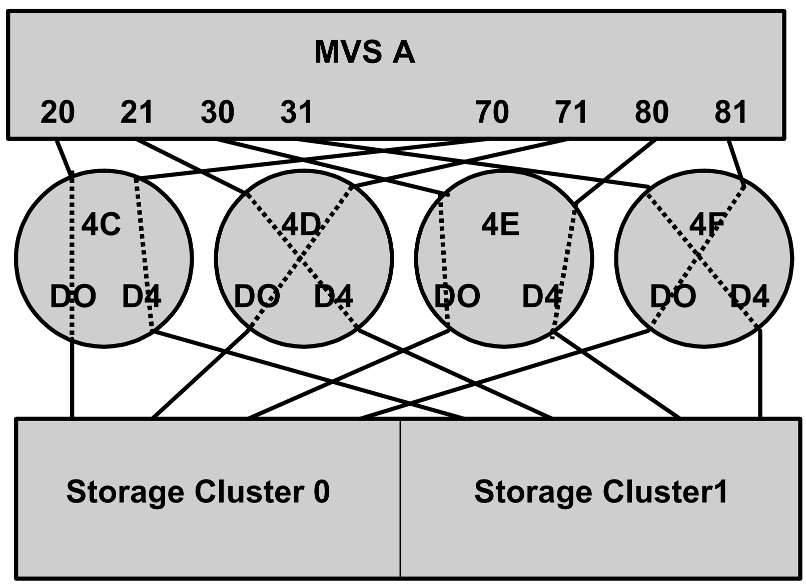


Figure 71. ***Configuration Diagram: Single MVS Host Connected to a VSM5 via FICON Directors***

```
ESCD4C   CHPID PATH=(20,70),TYPE=FC,SWITCH=4C
ESCD4D   CHPID PATH=(21,71),TYPE=FC,SWITCH=4D
ESCD4E   CHPID PATH=(30,80),TYPE=FC,SWITCH=4E
ESCD4F   CHPID PATH=(31,81),TYPE=FC,SWITCH=4F

CU1      CNTLUNIT CUNUMBR=001,
                  PATH=(20,21,30,31,70,71,80,81),
                  LINK=(D0,D4,D0,D4,D4,D0,D4,D0),
                  UNIT=3490,CUADD=0,
                  UNITADD=((00,16))

STRING1  IODEVICE ADDRESS=(0500,16),
                  CUNUMBER=(001),
                  UNIT=3490,
                  UNITADD=00,STADET=Y

CU2      CNTLUNIT CUNUMBR=002,
                  PATH=(20,21,30,31,70,71,80,81),
                  LINK=(D0,D4,D0,D4,D4,D0,D4,D0),
                  UNIT=3490,CUADD=1,
                  UNITADD=((00,16))

STRING2  IODEVICE ADDRESS=(0510,16),
                  CUNUMBER=(002),
                  UNIT=3490,
                  UNITADD=00,STADET=Y
                  .
                  .
                  .
CU15     CNTLUNIT CUNUMBR=015,
                  PATH=(20,21,30,31,70,71,80,81),
                  LINK=(D0,D4,D0,D4,D4,D0,D4,D0),
                  UNIT=3490,CUADD=E,
                  UNITADD=((00,16))

STRING15  IODEVICE ADDRESS=(05E0,16),
                  CUNUMBER=(015),
                  UNIT=3490,
                  UNITADD=00,STADET=Y

CU16     CNTLUNIT CUNUMBR=016,
                  PATH=(20,21,30,31,70,71,80,81),
                  LINK=(D0,D4,D0,D4,D4,D0,D4,D0),
                  UNIT=3490,CUADD=F,
                  UNITADD=((00,16))

STRING16  IODEVICE ADDRESS=(05F0,16),
                  CUNUMBER=(016),
                  UNIT=3490,
                  UNITADD=00,STADET=Y
```

Figure 72. ***IOCP Example: Single MVS Host Connected to a VSM5 via FICON Directors***

Hint: Unlike ESCON, FICON supports multiple active I/Os per channel. If the number of active VTDs is less than the number of channels configured to the VTSS, the I/Os to those VTDs may not be evenly spread across all the channels. As the number of active VTDs increases to be greater than the number of channels configured to the VTSS, the channel subsystem will spread the I/Os across all the channels. If it is desired to spread the I/Os across all of the channels even when only a few VTDs are active, it is necessary to use the preferred path feature to force the channel subsystem to spread the I/Os across the channels. The preferred path feature is specified via the `PATH=` parameter on the `IODEVICE` statement. When you specify preferred path on the `IODEVICE` statement, the channel subsystem always tries the preferred path first. If it is busy or unavailable, the channel subsystem next tries the channel path following the preferred path in the rotation order, and so on.

Figure 72 on page 208 (repeated in Figure 73) shows `IODEVICE` statements for `STRING1` **without** using preferred pathing.

```
STRING1  IODEVICE ADDRESS=(0500,16),
                  CUNUMBER=(001),
                  UNIT=3490,
                  UNITADD=00,STADET=Y
```

Figure 73. ***IODEVICE Statements for STRING 1 without Preferred Pathing***

Figure 74 on page 210 shows `IODEVICE` statements for `STRING1` **using** preferred pathing. If you're using preferred pathing, you need to use these kind of `IODEVICE` statements for **all** paths, such as `STRING2` through `STRING16` in Figure 72 on page 208.

```
STRING10 IODEVICE ADDRESS=(0500,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=20

STRING12 IODEVICE ADDRESS=(0502,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=21

STRING14 IODEVICE ADDRESS=(0504,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=30

STRING16 IODEVICE ADDRESS=(0506,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=31

STRING18 IODEVICE ADDRESS=(0508,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=70

STRING1A IODEVICE ADDRESS=(050A,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=71

STRING1C IODEVICE ADDRESS=(050C,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=80

STRING1E IODEVICE ADDRESS=(050E,2),
         CUNUMBER=(001),
         UNIT=3490,
         UNITADD=00,STADET=Y,
         PATH=81
```

Figure 74. ***IODEVICE Statements for STRING 1 Using Preferred Pathing***

Appendix G. Swapping VTSSs

This procedure covers **only** swapping out a VTSS and replacing it with another VTSS, a. k. a., a "push-pull swap". This procedure, therefore, **does not** cover changes to the VSM configuration such as adding or removing RTDs or any other changes that require using VTCS `CONFIG` or HSC `MERGEcds`.

Accordingly, **note that** a push-pull swap **requires** the following to ensure that no `CONFIG`/`MERGEcds` changes are needed:

- The RTD channel interfaces and MVS device addresses are the same on the new VTSS as they were on the old VTSS.
- The VTSSNAME is the same on the new VTSS as it was on the old VTSS.

Note: There is **one swap** where you'll have to make `CONFIG` changes, and that is when you're going from a VSM2/3 to a VSM4/5 or vice versa. VSM2/3s have 64 VTDs, whereas VSM4/5s can be configured with up to 256 VTDS. For more information, see *VTCS Command and Utility Reference* and *VTCS Installation and Configuration Guide*.

Swapping Out the Old VTSS

Let's start by swapping out the existing (old) VTSS.

To swap out the old VTSS:

1. **Stop sending work to the VTSS(s) you are replacing (old VTSS).**

   Basically, this means undoing allocation to VSM via any of the following methods:

   - The StorageTek DFSMS interface.
   - HSC, MVS/CSC, or SMC `TAPEREQ` statements.
   - HSC or MVS/CSC User Exits.

   What if you're replacing all your VTSSs? Or if you only have one? In this case, you may want to route tape jobs to real tape until your new VTSSs are up and running.

2. **For sites that have multiple VTSSs, change the SMC allocation parameters:**

   ```
   F SMC0,ALLOCDEF LIST
   F SMC0,ALLOCDEF MINLEVEL(1)
   ```

3. **MVS vary all VTDs offline to MVS...**

   ..to further ensure that nothing new gets sent to the old VTSS:

   ```
   V xxxx-yyyy,OFF
   ```

   Note: For those sites that use allocation software such as CA-MIM or IBM-STAR, use the commands for those packages, not MVS vary commands.

4. **VTCS vary quiesced the old VTSS(s):**

   .VT V VTSS(*vtssname*) QUIESCED

   Varying to quiesced state accomplishes a couple of important items:

   - In quiescing state (a transitional state to quiesced), VTCS does not direct any DD allocation to the VTSS, which still accepts pending mounts to allow those long running jobs with unit=aff chains to complete. When all VTDs are no longer in use (their UCBs are not allocated on MVS), the VTSS goes to quiesced state.
   - In quiesced state, the VTSS continues to accept and process back-end work; for example, migrates, recalls, and audits...which is a good thing, because you'll need these capabilities in succeeding steps.

5. **Use** `Display VTSS` **to monitor the quiesce process:**

   .VT D VTSS

   You have the high sign when you see `QUIESCED` in the `State` column.

6. **Use** `SET MIGOPT` **to ramp up your migration throughput to the maximum.**

   For example, if you have a VTSS with 8 RTDs, the following will work:

   .VT SET MIGOPT MAX(8) MIN(8)

   This looks a little dicey because you're not allowing any RTDs for recalls or reclaims. That's okay in this case, however, because you're about to migrate the old VTSS to zero in Step 7.

7. **Migrate the old VTSS to zero:**

   .VT MIGRATE VTSS(*vtssname*) THRESHLD(0)

   You'll get console messages when the migrate to zero succeeds, but to confirm, once again use `Display VTSS`.

8. **Audit the old VTSS and run a VTV report:**

   .VT AUDIT VTSS(*vtssname*)
   .VT VTVRPT

   You're just double-checking that there are no VTVs left on the old VTSS. On the VTV Report, if there is anything in the `RESD` column and it's owned by the old VTSS, then you have to put the old VTSS into a quiesced state and either execute another migrate to zero with associated checks (Step 4 through Step 7) or explicitly migrate the VTVs that are still resident.

   If there are no unavailable VTVs, the old VTSS is clean and all data is on MVCs.

9. **VTCS Vary offline all RTDs connected to the old VTSS.**

   .VT V RTD(*xxxx-yyyy*) OFF

10. **Ensure the RTDs are offline:**

    `.VT Display RTD`

    So we now have:

    - The old VTSS should be Quiesced and contain zero VTVs.
    - All connected RTDs offline.
    - All VTDs offline to MVS.

    Time to pull the plug...

11. **If all this is true then vary the old VTSS offline:**

    .VT Vary VTSS(*vtssname***) OFF**

12. **Ensure that the VTSS is offline (not just offline pending):**

    `.VT Display VTSS Detail`

13. **MVS vary offline all paths from all hosts to the old VTSS.**

    V PATH(*xxxx-yyyy,zz***),OFF**

14. **To ensure that all paths are offline to the old VTSS, from each system, enter:**

    D M=DEV(*aaaa***)**

    Where *aaaa* is a VTD address.

    Verify that all paths show “offline”.

15. **Physically uncable/remove the old VTSS.**

    This completes the swap-out part of the procedure, so continue with “Swapping In the New VTSS” on page 215.

Swapping In the New VTSS

As you can imagine, the swap-in process is pretty much the opposite of the swap-out process, with a few significant tweaks. You now know why you're doing most of the steps in the following procedure, so we'll give you a bare bones procedure.

To swap in the new VTSS:

1. **Physically install/cable up the new VTSS.**

1. **MVS vary online all paths from all hosts to the new VTSS.**

   `V PATH(`*`xxxx-yyyy,zz`***`),ON`**

2. **To ensure that all paths are online to the new VTSS, from each system, enter:**

   `D M=DEV(`*`aaaa`***`)`**

   Where *aaaa* is a VTD address.

   Verify that at least one path shows "online".

3. **Stop HSC on one of the hosts physically connected to the new VTSS:**

   `P` ***`HSC`***

   Make sure that HSC comes all the way down.

4. **Start HSC back up on the host in Step 3:**

   `S` ***`HSC`***

   As HSC starts up there must be **no** ECAM errors accessing the new VTSS as all of the paths are online. If there are any ECAM errors then the VTD tables may be built incorrectly.

5. **Make sure that HSC has reached full initialization complete:**

   `.D SRV`

6. **VTCS vary online the new VTSS:**

   `.VT VARY VTSS(`*`vtssname`***`) ON`**

7. **Ensure that the new VTSS comes online globally:**

   `.VT Display VTSS Detail`

   If the new VTSS goes to online pending there are only two valid reasons:

   - One or more hosts where HSC is down.
   - Hosts that are configured but not physically connected to the new VTSS.

   Any other reason requires investigation and if necessary reporting to StorageTek software support.

8. **Vary the new VTSS RTDs online**

   `.VT Vary RTD(`*`xxxx-yyyy`***`) ONline`**

9. **Ensure the new VTSS RTDs come online:**

   `.VT D RTD`

10. **Vary online the new VTSS VTDs to MVS:**

    V *xxxx-yyyy*,**ONline**

    The new VTSS should now be online and usable to MVS.

11. **Return the migration parameters to their previous settings (*x* and *y*):**

    .VT SET MIGOPT MAXmig(*x*) MINmig(*y*)

12. **Return the SMC** ALLOCDEF MINLEVEL **to what it was before the change (*x*):**

    F SMCO,ALLOCDEF MINLEVEL(*x*)

13. **Start sending work to the new VTSS.**

    Basically, this means reviewing/redoing allocation to VSM via any of the following methods:

    - The StorageTek DFSMS interface.
    - HSC, MVS/CSC, or SMC TAPEREQ statements.
    - HSC or MVS/CSC User Exits.

Appendix H. Using T10000 Drives as RTDs

Prerequisites for T10000 Drives for NCS/VTCS

Table 50. Prerequisites for T10000 Drives

Description	Requirement
NCS/VTCS - Base T10000 Support	6.0 with the following PTFs: • L1H12ZI (SOS6000) • L1H12ZJ (SWS6000) 6.1 with the following PTFs: • L1H12ZN (SOS6100) • L1H12ZO (SWS6100) Base support integrated into 6.2
NCS/VTCS - T10000 Encryption Support	6.1 with the following PTFs: • L1H13AA (SOS6100) • L1H136K (SWS6100) 6.2 with the following PTFs: • L1H139D (SOS6200) • L1H139C (SWS6200)
LSMs	9310 and SL8500 at LMU Compat Level 13
T10000 drive protocol	FICON
VTSSs	VSM4 and VSM5
T10000 drive media	T10000T1 (full capacity 500GB cartridge) T10000TS (120GB sport cartridge) T10000E1 (full capacity 500GB encryption cartridge) T10000ES (120GB encryption sport cartridge) **Note:** The encryption media types (T10000E1 and T10000ES) should be used **only** on the STORCLAS MEDIA parameter. Using them in other control cards (VOLATTR, TAPEREQ, etc.) does not produce an error, but these media types **will not** affect the implied RECTECH. For example, media T10000ES on TAPEREQ MEDIA **will not** direct an allocation to an encrypting drive type.

Defining T10000 Drives to NCS/VTCS

New T10000 RTDs, which are Nearline transports, require common 4-digit hexadecimal MVS unit addresses for the following:

- The `SLIDRIVS` macro to define RTD device addresses during the HSC `LIBGEN` update.
- `UNITATTR` statements **for HSC 6.0 only**. For NCS 6.1, The `UNITATTR` statement has been moved from HSC to SMC is required **only** to set the real transport model type for non-library transports (which are not supported for VSM). For more information, see *SMC Configuration and Administration Guide*.

  For HSC 6.0, you must use `UNITATTR`s to define T10000 drives as 3490E image for use as RTDs, where the valid `MODEL` values are `T1A` (any T10000 drive) and `T1A34` (T10000 drive as 3490E). For example:

  ```
  UNITATTR ADDR(8800-8803) MODEL(T1A34)
  ```

- The VTCS `CONFIG VTSS RTD DEVNO` parameter.

  Hint: You also specify the RTD identifier on the `CONFIG VTSS RTD NAME` parameter. To help identify the RTDs connected to each VTSS, StorageTek recommends that you choose RTD identifiers that reflect the VTSS identifier (specified on the `VTSS NAME` parameter) and the RTD's MVS device number (specified on the `RTD DEVNO` parameter).

  For example:

  ```
  RTD    NAME=VTS18800 DEVNO=8800 CHANIF=0A
  RTD    NAME=VTS18801 DEVNO=8801 CHANIF=0I
  RTD    NAME=VTS18802 DEVNO=8802 CHANIF=1A
  RTD    NAME=VTS18803 DEVNO=8803 CHANIF=1I
  ```

  In configurations where multiple VTSSs are connected to and dynamically share the same RTD, in each VTSS definition you can either assign unique RTD identifiers or use the same RTD identifier.

- The `HCD` facility to assign MVS device numbers to these transports.

Caution: Note the following:

- StorageTek **strongly recommends** that you define your RTDs to MVS (as normal 3490 tape drives), even if you do not intend to vary them online to MVS. This prevents the RTD addresses used in `CONFIG` and `LIBGEN` from accidentally being used for other devices. If you do not do this, and subsequently use the addresses for other MVS devices, you will cause problems with LOGREC processing, because VTCS will write records using the RTD addresses, and MVS will write records for other devices with those same addresses.

- StorageTek **requires** that RMM users define their RTDs to MVS. RMM causes problems if it sees the `IEC501` mount message generated for RTDs by VTCS and if the device in the mount message is not defined to MVS.

☞ **Note:**

- You can specify that Nearline transports can only be used as RTDs in the `LIBGEN`. For more information, see "Creating or Updating the HSC LIBGEN" on page 80.
- Ensure that you use the T10000 Virtual Operator Panel (VOP) to enable the SL PROT (Standard Label Protect) function on the RTDs.

Defining T10000 Media to NCS/VTCS

You must define T10000 media as VTCS MVCs. You define MVCs as described in the following sections:

- "Define and Select Nearline Volumes" on page 222
- "Define Available MVCs with CONFIG" on page 223
- "Define the MVC Pool" on page 224
- "Protect MVCs and Nearline Volumes" on page 225

Define and Select Nearline Volumes

First, to define and select Nearline volumes for MVCs, use these guidelines:

- MVCs require `VOLATTR` statements to ensure that VTCS will select the correct RTD device type for each MVC. Select volumes for MVCs that are compatible with your system's RTD transport types. For T10000 media, the valid `VOLATTR MEDIA` value for a full capacity cartridge is `T10000T1` (or `T1`) and for the sport cartridge, T10000TS (or TS). For example:

```
VOLATTR SERIAL(N35000-N35999) MEDIA(T10000T1)
```

- For mixed–media VSM systems, select volumes that include at least one media type compatible with each of your system's RTD transport types.

  Note that VSM selects media for migration processing and reclaim processing according to the media types of volumes in your system's MVC pool.

- If you define new Nearline volumes as MVCs, you must create MVS volsers for these volumes and initialize STANDARD, ECART, and ZCART volumes as 36-track format standard label volumes.
- As described in "Protect MVCs and Nearline Volumes" on page 225, if possible, create a new and separate volser range for MVCs. Ensure that if you define new volumes, you do not overlap existing TMS ranges.
- Note that the VTCS Display and MVC report outputs are updated to denote the `T10000T1` and T1000TS media types.

Define Available MVCs with CONFIG

Second, use VTCS `CONFIG` to define all MVCs *available* to VTCS. `CONFIG` reserves space for these volumes in the HSC CDS. The `MVCPool` statements define the *MVC pool*, which contains the MVCs that VTCS actually *uses*. For example:

```
MVCVOL LOW=N35000 HIGH=N35999
```

For an initial `CONFIG` definition, consider defining only enough MVCs for reasonable growth of your MVC pool. This method allows you to expand your MVC pool without rerunning `CONFIG` (you only have to change your `MVCPool` statements) but does not reserve unnecessary space in the CDS, which can impact MVC processing performance. Note that if the CDS does not contain sufficient space to run VTCS `CONFIG`, you will also have to run HSC `RECONFIG`.

For example, if you currently need 300 MVCs but will need to add 150 more MVCs within the next 6 months, define an MVC range of 450 volsers with `CONFIG`, but only apply `MVCPool` statements to the first 300 "in use" MVCs. As your MVC space requirements increase, update and reapply your `MVCPool` statements to add the second 150 MVCs.

If your MVC space requirements expand beyond the second 150 MVCs, then rerun `CONFIG` to define additional MVC ranges and update and reapply your `MVCPool` statements.

Note:

- You can only add new MVC ranges. A range can consist of a single volume. You cannot delete or modify existing ranges.
- A VSM audit of all MVCs will audit all MVCs defined with `CONFIG` including those that are *not* specified in the `MVCPool` statements.

Define the MVC Pool

Third, create `MVCPool` statements, which specify the pool of MVCs available for migration and consolidation requests, using the following guidelines:

- Because `MVCPool` statements specify the "in use" MVCs, `MVCPool` statements can (and typically do) define a subset of the available MVCs you defined via `CONFIG`. `MVCPool` statements, however, can only specify MVCs you already defined with `CONFIG`. For example:

  `MVCPOOL VOLSER(N35000-N35999)`

- StorageTek recommends that you use identical `MVCPool` statements on all hosts. A host can automigrate any VTV on any VTSS to which the host is connected, including VTVs created by another host. If your VSM configuration consists of hosts cross-connected to multiple VTSSs, therefore, separate MVC pools do not guarantee that a host automigrates only VTVs it creates to only its MVC pool. To most effectively segregate VTVs on groups of MVCs, see "Creating and Using VSM Management and Storage Classes" on page 96.

- Ensure that your MVC pool consists of volumes that are physically located in ACS that contains your system's RTDs.

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

- To redefine your MVC pool, change your `MVCPool` statements and reload them via the `VT MVCDEF` command.

Protect MVCs and Nearline Volumes

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

- If possible, create a new and separate volser range for MVCs to prevent HSC from writing to MVCs and to prevent VSM from writing to conventional Nearline volumes.
- VTCS, not MVS, controls access to MVCs. The tape management system does not control VSM access to an MVC volume and does not record its usage. If you choose to define MVCs to the tape management system, to ensure that the tape management system does not accidentally access MVCs, follow the guidelines in *VTCS Installation and Configuration*.
- Use your security system to restrict access to MVCs.
- HSC automatically marks newly entered MVC volumes as non-scratch. If you define existing Nearline volumes as MVCs, ensure that these volumes do not contain data you need, then run the HSC `UNSCratch` Utility to unscratch them. For more information, see *HSC System Programmer's Guide for MVS*.

Changed VTCS Migrate and Reclaim Hierarchies

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

1. Standard - 400 Mb
2. ECART - 800 Mb
3. ZCART - 1600 Mb
4. 9840 - 20 Gb
5. 9840C - 40 Gb
6. 9940A - 60 Gb
7. T10000 Sport - 120GB
8. 9940B - 200 Gb
9. T10000 Full - 500 Gb

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

1. T10000 Full - 500 Gb
2. 9940B - 200 Gb
3. T10000 Sport - 120GB
4. 9940A - 60 Gb
5. 9840C - 40 Gb
6. 9840 - 20 Gb
7. ZCART -1600 Mb
8. ECART - 800 Mb
9. Standard - 400 Mb

Glossary

A

access method A technique for moving data between processor storage and input/output devices.

ACS *See* Automated Cartridge System.

ACSid A method used to identify an ACS. An ACSid is the result of defining the SLIALIST macro during the library generation (LIBGEN) process. The first ACS listed in this macro acquires a hexadecimal identifier of 00, the second ACS listed acquires a hexadecimal identifier of 01, and so forth, until all ACSs are identified.

ACS routine An SMS term, referring to automatic class selection routine. Not to be confused with the HSC term, ACS, referring to automatic cartridge system.

AMT automatic migration threshold.

APF Authorized Program Facility.

APPL VTAM APPLID definition for the HSC.

archiving The storage of backup files and associated journals, usually for a given period of time.

audit A VSM audit (which is not the same as an HSC audit) reconstructs VTV and MVC information.

Automated Cartridge System (ACS) The library subsystem consisting of one or two LMUs, and from 1 to 16 attached LSMs.

automated library *See* library.

automatic mode A relationship between an LSM and all attached hosts. LSMs operating in automatic mode handle cartridges without operator intervention. This is the normal operating mode of an LSM that has been modified online.

automatic migration Migrating VTVs to MVCs that is automatically initiated and controlled by VSM.

automatic migration threshold (AMT) AMT values are percentage values that determine when virtual tape volume migration begins and ends. VTV migration begins when the VTSS buffer reaches the high AMT and ends when the buffer reaches or falls below the low AMT. These thresholds apply to all VTSSs.

automatic recall Recalling VTVs to the VTSS that is automatically initiated and controlled by VSM.

automatic reclaim Reclaiming MVC space that is automatically initiated and controlled by VSM.

B

back-end capacity The capacity of the VTSS disk buffer, in bytes, as defined in disk arrays excluding space for system overhead.

block A collection of contiguous records recorded as a unit. Blocks are separated by interblock gaps, and each block may contain one or more records.

buffer A routine or storage used to compensate for a difference in rate of data flow, or time of occurrence of events, when transferring data from one device to another.

C

CA–1 (TMS) Computer Associates Tape Management System. Third–party software by Computer Associates International, Inc.

CAP *See* Cartridge Access Port.

capacity *See* media capacity.

CAPid A CAPid uniquely defines the location of a CAP by the LSM on which it resides. A CAPid is of the form *AAL:CC* where *AA* is the ACSid, *L* is the LSM number, and *CC* is the CAP number. Some commands and utilities permit an abbreviated CAPid format of *AAL*.

cartridge The plastic housing around the tape. It is approximately 4 inches (100 mm) by 5 inches (125 mm) by 1 inch (25 mm). The tape is threaded automatically when loaded in a transport. A plastic leader block is attached to the tape for automatic threading. The spine of the cartridge contains a Tri–Optic label listing the VOLSER (tape volume identifier).

Cartridge Access Port (CAP) An assembly which allows an operator to enter/eject cartridges during automated operations. The CAP is located on the access door of an LSM. (*see also,* standard CAP, enhanced CAP, WolfCreek CAP, WolfCreek optional CAP.)

Cartridge Scratch Loader An optional feature for the Cartridge Drive. It allows the automatic loading of premounted tape cartridges or the manual loading of single tape cartridges.

cartridge system tape The basic tape cartridge media that is used with 4480, 4490, or 9490 Cartridge Subsystems. They are visually identified by a one–color cartridge case.

CAW *See* Channel Address Word.

CDRM Cross Domain Resource Manager definition (if not using existing CDRMs).

CDRSC Cross Domain Resource definition.

CDS *See* control data set.

CE Channel End.

cell A storage slot in the LSM that is used to store a tape cartridge.

Central Support Remote Center (CSRC) *See* Remote Diagnostics Center.

CFT Customer field test.

channel A device that connects the host and main storage with the input and output control units.

Channel Address Word (CAW) An area in storage that specifies the location in main storage at which a channel program begins.

channel command A command received by a CU from a channel.

Channel Status Word (CSW) An area in storage that provides information about the termination of input/output operations.

check Detection of an error condition.

CI Converter/Interpreter (JES3).

Clink (cluster link). The path between a primary VTSS and secondary VTSS in a cluster. The Clink path is used to copy replicate VTVs from the primary to the secondary.

Cluster. Two VTSSs which are physically cabled together by Clink paths and are defined in CONFIG as a cluster. A cluster consists of a primary and a secondary VTSS. VTVs with the replicate attribute attached will be copied from the primary to the secondary as soon as possible after dismount time.

connected mode A relationship between a host and an ACS. In this mode, the host and an ACS are capable of communicating (at least one station to this ACS is online).

control data set (CDS) The HSC database. In addition to the current information in the CDS, VSM keeps all its persistent data in the CDS as well.

control data set allocation map A CDS subfile that marks individual blocks as used or free.

control data set data blocks CDS blocks that contain information about the library and its configuration or environment.

control data set directory A part of the CDS that maps its subdivision into subfiles.

control data set pointer blocks CDS blocks that contain pointers to map data blocks belonging to a subfile.

control data set recovery area A portion of the CDS reserved for maintaining integrity for updates that affect multiple CDS blocks.

control data set subfile A portion of the CDS consisting of Data Blocks and Pointer Blocks containing related information.

Control Unit (CU) A microprocessor–based unit situated logically between a host channel (or channels) and from two to sixteen tape transports. It functions to translate channel commands into tape transport commands, send transport status to the channel(s), and pass data between the channel(s) and transport(s).

conventional Nearline transport An HSC–controlled transport that is not defined to VSM as an RTD.

cross–host recovery The ability for one host to perform recovery for another host that has failed.

CSE Customer Service Engineer.

CSI Consolidated System Inventory.

CSL Cartridge Scratch Loader.

CSRC Central Support Remote Center (*See* Remote Diagnostics Center)

CSW Channel Status Word.

CU *See* Control Unit.

D

DAE Dump Analysis Elimination.

DASD Direct access storage device.

data Any representations such as characters or analog quantities to which meaning is, or might be, assigned.

data class A collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set.

data compaction An algorithmic data–reduction technique that encodes data from the host and stores it in less space than unencoded data. The original data is recovered by an inverse process call decompaction.

data–compaction ratio The number of host data bytes divided by the number of encoded bytes. It is variable depending on the characteristics of the data being processed. The more random the data stream, the lower the opportunity to achieve compaction.

Data Control Block (DCB) A control block used by access routines in storing and retrieving data.

data set The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data streaming A continuous stream of data being transmitted in character or binary–digit form, using a specified format.

DBU disk buffer utilization.

DCB Data Control Block.

demand allocation An MVS term meaning that a user has requested a specific unit.

demand migration Migrating VTVs to MVCs that an administrator does with the MIGRATE command or utility.

demand recall Recalling VTVs to the VTSS that an administrator does with the RECALL command or utility.

demand reclaim Reclaiming MVC space that an administrator does with the RECLAIM command or utility.

device number A four–digit hexadecimal number that uniquely identifies a device attached to a processor.

device separation The HSC function which *forces* the MVS device selection process to choose either a nonlibrary transport or a transport in a particular ACS, based on the location of the volume (specific requests) or the given subpool rules in effect (nonspecific request).

DFP Data Facility Product. A program that isolates applications from storage devices, storage management, and storage device hierarchy management.

DFSMS Refers to an environment running MVS/ESA SP and DFSMS/MVS, DFSORT, and RACF. This environment helps automate and centralize the management of storage through a combination of hardware, software, and policies.

DFSMS ACS routine A sequence of instructions for having the system assign data class, storage class, management class, and storage group for a data set.

directed allocation The HSC function of *influencing* MVS's selection of library transports. For a specific request, the HSC influences MVS to choose a transport requiring the fewest number of pass–thrus; for a nonspecific (scratch) request, HSC's influencing is based on the given subpool rules in effect.

disconnected mode A relationship between a host and an ACS. In this mode, the host and an ACS are not capable of communicating (there are no online stations to this ACS).

disk buffer utilization (DBU). The ratio of used to total VTSS buffer capacity.

DOMed Pertaining to a console message that was previously highlighted during execution, but is now at normal intensity.

drain The deletion of data from an MVC. May be accompanied by a "virtual" eject to prevent the MVC from being reused.

drive loaded A condition of a tape drive in which a tape cartridge has been inserted in the drive, and the tape has been threaded to the beginning–of–tape position.

DSI Dynamic System Interchange (JES3).

dual LMU A hardware/u–software feature that provides a redundant LMU capability.

dual LMU HSC release 1.1.0 or later that automates a switchover to the standby LMU in a dual LMU configuration.

dump To write the contents of storage, or of a part of storage, usually from an internal storage to an external medium, for a specific purpose such as to allow other use of storage, as a safeguard against faults or errors, or in connection with debugging.

Dynamic Device Reconfiguration (DDR) A facility that allows a demountable volume to be moved, and repositioned if necessary, without abnormally terminating the job or repeating the initial program load procedure.

E

Ecart Cartridge system tape with a length of 1100 feet that can be used with 4490 cartridge drives. These tapes are visually identified by a two–tone colored case.

EDL *See* eligible device list.

eligible device list A group of tape drives that are available to satisfy an allocation request.

enhanced CAP An enhanced CAP contains two forty–cell magazine–style CAPs and a one–cell priority CAP (PCAP). Each forty–cell CAP holds four removable magazines of ten cells each. An LSM access door with an enhanced CAP contains no cell locations for storing cartridges. An enhanced CAP is ordered as Feature Number CC80. (*see also,* Cartridge Access Port (CAP), standard CAP, WolfCreek CAP, WolfCreek optional CAP.)

Effective Recording Density The number of user bytes per unit of length of the recording medium.

eject The LSM robot places a cartridge in a Cartridge Access Port (CAP) so the operator can remove it from the LSM.

ExPR Expert Performance Reporter.

Expert Performance Reporter Expert Performance Reporter collects performance data and generates reports about StorageTek Nearline ACSs and VTSS status and performance. It has an MVS component and a PC component.

Enhanced Capacity Cartridge System Tape Cartridge system tape with increased capacity that can be used with 4490 and 9490 Cartridge Drives. These tapes are visually identified by a two–tone colored case.

EOT End–of–Tape marker.

EPO Emergency Power Off.

ERDS Error Recording Data Set.

EREP Environmental Recording, Editing, Printing.

ERP Error recovery procedures.

error recovery procedures (ERP) Procedures designed to help isolate and, where possible, to recover from errors in equipment.

ExtendedStore Library One or more LSMs with no cartridge drives (CDs) that are attached by pass–thru ports to other LSMs (with CDs) in an ACS. These LSMs provide archive storage for cartridges containing less active data sets. Cartridges can be entered and ejected directly into and out of this LSM though either a standard CAP or an enhanced CAP.

F

file protected Pertaining to a tape volume from which data can be read only. Data cannot be written on or erased from the tape.

format The arrangement or layout of data on a data medium.

G

GB 1,073,741,824 bytes of storage.

GDG Generation Data Group. An MVS data set naming convention. Sequence numbers are appended to the basic data set name to track the generations created for that data set.

GTF Generalized Trace Facility. An MVS facility used to trace software functions and events.

H

HDA Head/disk assembly.

Host Software Component (HSC) That portion of the Automated Cartridge System which executes on host systems attached to an automated library. This component acts as the interface between the operating system and the rest of the automated library.

host system A data processing system that is used to prepare programs and the operating environments for use on another computer or controller.

HSC Host Software Component.

HSM Hierarchical Storage Manager.

HWS High Watermark Setup. Relates to chains set up for tape transport allocation in JES3.

I

ICRC See Improved Cartridge Recording Capability.

Improved Cartridge Recording Capability (ICRC) An improved data recording mode that, when enabled, can increase the effective cartridge data capacity and the effective data rate when invoked.

ID Identifier or identification.

IDAX Interpreter Dynamic Allocation Exit. This is a subfunction of the DFSMS/MVS subsystem request (SSREQ 55) that the MVS JCL Interpreter and dynamic allocation functions issue for calling

DFSMS ACS routines for management of the data set requested.

IML *See* Initial Microprogram Load.

index a function performed by the cartridge loader that moves cartridges down the input or output stack one cartridge position. A loader can perform multiple consecutive indexes.

Initial Microprogram Load (IML) A process that activates a machine reset and loads system programs to prepare a computer system for operation. Processors having diagnostic programs activate these programs at IML execution. Devices running u–software reload the functional u–software usually from a floppy diskette at IML execution.

Initial Program Load (IPL) A process that activates a machine reset and loads system programs to prepare a computer system for operation. Processors having diagnostic programs activate these programs at IPL execution. Devices running u–software reload the functional u–software usually from a floppy diskette at IPL execution.

initial value A value assumed until explicitly changed. It must then be explicitly specified in another command to restore the initial value. An initial value for the HSC is the value in effect when the product is installed.

inline diagnostics Diagnostic routines that test subsystem components while operating on a time–sharing basis with the functional u–software in the subsystem component.

input stack The part of the cartridge loader where cartridges are premounted.

intervention required Manual action is needed.

ips Inches per second.

IVP Installation Verification Programs. A package of programs that is run by a user after the library is installed in order to verify that the library is functioning properly.

J

JCL See Job Control Language.

Job Control Language Problem–oriented language designed to express statements in a job that are used to identify the job or describe its requirements to an operating system.

journal The log associated with journaling. The log (stored in a data set) contains a record of completed work and changes to the control data set since the last backup was created.

journaling A technique for recovery that involves creating a backup control data set and maintaining a log of all changes (transactions) to that data set.

K

KB Kilobyte, thousand bytes, or 1024 bytes.

kb kilobit, or thousand bits (10^3 bits).

keyword parameter In command and utility syntax, operands that include keywords and their related values (*see* “positional parameter”). Values are concatenated to the keyword either by an equal sign, “KEYWORD=value,” or by parentheses, “KEYWORD(value).” Keyword parameters can be specified in any order. The HSC accepts (tolerates) multiple occurrences of a keyword. The value assigned to a keyword reflects the last occurrence of a keyword within a command.

L

LAN Local Area Network.

LCU *See* Library Control Unit.

LED *See* Light Emitting Diode.

LIBGEN The process of defining the configuration of the automated library to the host software.

library An installation of one or more ACSs, attached cartridge drives, volumes placed into the ACSs, host software that controls and manages the ACSs and associated volumes, and the library control data set that describes the state of the ACSs.

library control data set *See* control data set.

Library Control Unit (LCU) The portion of the LSM that controls the picking, mounting, dismounting, and replacing of cartridges.

Light Emitting Diode (LED) An electronic device used mainly as an indicator on status panels to show equipment on/off conditions.

LMU Library Management Unit. The portion of the ACS that manages from one to sixteen LSMs and communicates with the host CPU.

loader *See* Cartridge Scratch Loader.

load point The beginning of the recording area on magnetic tape.

Local Area Network (LAN) A computer network in which devices within the network can access each other for data transmission purposes. The LMU and attached LCUs are connected with a local area network.

logical ejection The process of removing a volume from the control data set without physically ejecting it from its LSM location.

LSM Library Storage Module. Provides the storage area for cartridges plus the robot necessary to move the cartridges. The term LSM often means the LCU and LSM combined.

LSMid An LSMid is composed of the ACSid concatenated with the LSM number.

LSM number A method used to identify an LSM. An LSM number is the result of defining the SLIACS macro LSM parameter during a LIBGEN. The first LSM listed in this parameter acquires the LSM number of 0 (hexadecimal), the second LSM listed acquires a hexadecimal number of 1, and so forth, until all LSMs are identified (maximum of sixteen or hexadecimal F).

M

machine initiated maintenance See ServiceTek.

magnetic recording A technique of storing data by selectively magnetizing portions of a magnetizable material.

magnetic tape A tape with a magnetizable surface layer on which data can be stored by magnetic recording.

magnetic tape drive A mechanism for moving magnetic tape and controlling it's movement.

maintenance facility Hardware contained in the CU and LMU that allows a CSE and the RDC to run diagnostics, retrieve status, and communicate with respective units through their control panels.

management class A collection of management attributes, assigned by the storage administrator, that are used to control the allocation and use of space by a data set. Note that SMS Management Classes are different from VSM Management Classes.

manual mode A relationship between an LSM and all attached hosts. LSMs operating in manual mode have been modified offline and require human assistance to perform cartridge operations.

master LMU The LMU currently controlling the functional work of the ACS in a dual LMU configuration.

MDS Main Device Scheduler (JES3).

media capacity The amount of data that can be contained on storage media and expressed in bytes of data.

micro–software *See* υ–software under Symbols.

migration The movement of VTVs from the VTSS to the RTD where the VTVs are stacked onto MVCs. See *automatic migration* and *demand migration*.

MIM Multi–Image Manager. Third–party software by CA Corporation.

mixed configurations Installations containing cartridge drives under ACS control and cartridge drives outside of library control. These configurations cause the Host Software Component to alter allocation to one or the other.

modem Modulator/demodulator. An electronic device that converts computer digital data to analog data for transmission over a telecommunications line (telephone line). At the receiving end, the modem performs the inverse function.

monitor A device that observes, records, and verifies selected system activities to determine significant departure from expected operation.

Multi-Volume Cartridge (MVC) A physical tape cartridge residing in an LSM that either contains migrated virtual tape volumes (VTVs) or is identified as a volume that can be selected for VTV stacking.

MVCPool Statement An HSC control statement that is contained in the definition data set specified by the `VT MVCDEF` command. An `MVCPool` statement specifies the MVCs that VTCS uses.

MVCDEF An HSC command that is used to load the definition data set that contains `MVCPool` statements.

N

O

output stack The part of the cartridge loader that receives and holds processed cartridges.

P

paired–CAP mode The two forty–cell CAPs in an enhanced CAP function in paired–CAP mode as a single eighty–cell CAP.

PARMLIB control statements Parameter library (PARMLIB) control statements allow you statically specify various operation parameters which take effect at HSC initialization. Identifying your system requirements and then specifying the appropriate control statements permits you to customize the HSC to your data center.

Pass–Thru Port (PTP) A mechanism that allows a cartridge to be passed from one LSM to another in a multiple LSM ACS.

physical end of tape A point on the tape beyond which the tape is not permitted to move.

positional parameter In command and utility syntax, operands that are identified by their position in the command string rather than by keywords (*see* “keyword parameter”). Positional parameters must be entered in the order shown in the syntax diagram.

POST *See* Program for Online System Testing.

PowderHorn A high–performance LSM (model number 9310) featuring a high–speed robot. The PowderHorn has a capacity of up to approximately 6000 cartridges.

Primary. One of two VTSSs in a cluster which is designated in CONFIG as the primary. During normal operations the primary services the host workload and copies replicate VTVs to the secondary.

Program for Online System Testing (POST) A program in a host computer that allows it to test an attached subsystem while the subsystem is online.

Program Temporary Fix A unit of corrective maintenance delivered to a customer to repair a defect in a product, or a means of packaging a Small Programming Enhancement (SPE).

Program Update Tape A tape containing a collection of PTFs. PUTs are shipped to customers on a regular basis under the conditions of the customer's maintenance license.

PTF *See* Program Temporary Fix.

PTP *See* pass–thru port.

PUT *See* Program Update Tape.

R

RACF See Resource Access Control Facility.

Real Tape Drive (RTD) The physical transport attached to the LSM. The transport has a data path to a VTSS and may optionally have a data path to MVS or to another VTSS.

RDC *See* Remote Diagnostic Center.

recall The movement of VTVs from the MVC back to the VTSS. May be automatic or on demand.

reclaim Refers to MVC space reclamation. For automatic and demand reclamation, VTCS uses the amount of fragmented free space on the MVC and the amount of VTV data that would have to be moved to determine if space reclamation is justified.

Reconciliation. An automatic process initiated when a cluster is reestablished after the primary or secondary has been offline. Reconciliation ensures that the contents of the primary and secondary are identical with respect to replicate VTVs.

Recording Density The number of bits in a single linear track measured per unit of length of the recording medium.

Remote Diagnostic Center (RDC) The Remote Diagnostic Center at StorageTek. RDC operators can access and test StorageTek systems and software, through telecommunications lines, from remote customer installations. Also referred to as the Central Support Remote Center (CSRC).

Replication. Copying a replicate VTV from the primary VTSS to the secondary VTSS in a cluster. When replication completes, there are two copies of the VTV, one in the primary and one in the secondary.

Replicate VTV. A VTV which has had the replicate attribute attached to it by a management class statement.

Resource Access Control Facility (RACF) Security software controlling access to data sets.

RTD *See* real tape drive.

S

SCP See System Control Program.

scratch tape subpool A defined subset of all scratch tapes. Subpools are composed of one or more ranges of VOLSERs with similar physical characteristics (type of volume {reel or cartridge}, reel size, length, physical location, etc.). Some installations may also subdivide their scratch pools by other characteristics, such as label type (AL, SL, NSL, NL). The purpose of subpooling is to ensure that certain data sets are built only within particular ranges of volumes (for whatever reason the user desires). If a volume which does not belong to the required subpool is mounted for a particular data set, it is dismounted and the mount reissued.

Secondary. One of two VTSSs in a cluster which is designated in CONFIG as the secondary. During normal operations the secondary receives copies of replicate VTVs, stores them, and makes a migration copy on an MVC as soon as possible.

secondary recording A technique for recovery involving maintaining both a control data set and a copy (secondary) of the control data set.

SER Software Enhancement Request.

ServiceTek (machine initiated maintenance) A unique feature of the ACS in which an expert system monitors conditions and performance of subsystems and requests operator attention before a potential problem impacts operations. Customers can set maintenance threshold levels.

servo A device that uses feedback from a sensing element to control mechanical motion.

Small Programming Enhancement (SPE) A supplement to a released program that can affect several products or components.

SMF System Management Facility. An MVS facility used to record system actions which affect system functionality.

SMP System Modification Program.

SMP/E System Modification Program Extended.

SMS System Managed Storage.

SPE Small Programming Enhancement.

standard CAP A standard CAP has a capacity of twenty–one cartridges (three rows of seven cells each). An LSM access door with a standard CAP contains cell locations for storing cartridges. (*see also,* Cartridge Access Port (CAP), enhanced CAP.)

standard LSM A model 4410 LSM which a has a storage capacity of up to approximately 6000 cartridges.

standby The status of a station that has been varied online but is connected to the standby LMU of a dual LMU ACS.

standby LMU The redundant LMU in a dual LMU configuration that is ready to take over in case of a master LMU failure or when the operator issues the SWitch command.

station A hardware path between the host computer and an LMU over which the HSC and LMU send control information.

storage class A named list of storage attributes that identify performance goals and availability requirements for a data set. Note that SMS Storage Classes are different from VSM Storage Classes.

storage group A collection of storage volumes and attributes defined by the storage administrator. Note that this is an SMS concept, not a VSM concept.

switchover The assumption of master LMU functionality by the standby LMU.

System Control Program The general term to describe a program which controls access to system resources, and allocates those resources among executing tasks.

system–managed storage Storage that is managed by the Storage Management Subsystem, which attempts to deliver required services for availability, performance, space, and security applications.

System Modification Program Extended An IBM–licensed program used to install software and software maintenance.

T

tape cartridge A container holding magnetic tape that can be processed without separating it from the container.

tape drive A device that is used for moving magnetic tape and includes the mechanisms for writing and reading data to and from the tape.

TAPEREQ An HSC control statement that is contained in the definition data set specified by the TREQDEF command. A TAPEREQ statement defines a specific tape request. It is divided into two parts, the input: job name, step name, program name, data set name, expiration date or retention period, and an indication for specific requests or nonspecific (scratch) requests; and the output: media type and recording technique capabilities. You can use TAPEREQ statements to direct data sets to VSM.

tape unit A device that contains tape drives and their associated power supplies and electronics.

Timberwolf (9740) LSM A high performance LSM that provides a storage capacity of up to 494 cartridges. Up to 10 drives (STD, 4490, 9490, 9490EE, 9840, and SD-3) can be configured. Timberwolf LSMs can only attach to other Timberwolfs.

TMS Tape Management System.

TP Tape–to–Print.

transaction A short series of actions with the control data set. These actions are usually related to a specific function (e.g., Mount, ENter).

transport An electromechanical device capable of threading tape from a cartridge, moving the tape across a read/write head, and writing data onto or reading data from the tape.

TREQDEF An HSC command that is used to load the definition data set that contains TAPEREQ control statements.

Tri–Optic label An external label attached to the spine of a cartridge that is both human and machine readable.

TT Tape–to–Tape.

U

UNITATTR An HSC control statement that is contained in the definition data set specified by the UNITDEF command. A UNITATTR statement defines to the HSC the transport's media type and recording technique capabilities. For VSM, the UNITATTR statements define the VTD addresses to VSM as virtual and associate them with a VTSS.

UNITDEF An HSC command that is used to load the definition data set that contains UNITATTR control statements.

utilities Utility programs. The programs that allow an operator to manage the resources of the library and to monitor overall library performance.

V

Virtual Storage Manager (VSM) A storage solution that virtualizes volumes and transports in a VTSS buffer in order to improve media and transport use. The hardware includes VTSS, which is the DASD buffer, and RTDs. The software includes VTCS, an HSC–based host software, and VTSS microcode.

Virtual Tape Control System (VTCS) The primary host code that controls activity and information about VTSSs, VTVs, RTDs, and MVCs.

Virtual Tape Drive (VTD) An emulation of a physical transport in the VTSS that looks like a physical tape transport to MVS. The data written to a VTD is really being written to DASD. The VTSS has 64 VTDs that do virtual mounts of VTVs.

Virtual Tape Storage Subsystem (VTSS) The DASD buffer containing virtual volumes (VTVs) and virtual drives (VTDs). The VTSS is a STK RAID 6 hardware device with microcode that enables transport emulation. The RAID device can read and write "tape" data from/to disk, and can read and write the data from/to an RTD.

Virtual Tape Volume (VTV) A portion of the DASD buffer that appears to the operating system as a real tape volume. Data is written to and read from the VTV, and the VTV can be migrated to and recalled from real tape.

virtual thumbwheel An HSC feature that allows read–only access to a volume that is not physically write–protected.

VOLATTR An HSC control statement that is contained in the definition data set specified by the VOLDEF command. A VOLATTR statement defines to the HSC the media type and recording technique of the specified volumes. For VSM, the VOLATTR statements define the volsers for volumes that will be used as MVCs.

VOLDEF An HSC command that is used to load the definition data set that contains VOLATTR control statements.

VOLSER A six–character alphanumeric label used to identify a tape volume.

volume A data carrier that is mounted or demounted as a unit. (*See* cartridge).

VSM *See* Virtual Storage Manager.

VTCS *See* Virtual Tape Control System.

VTD *See* virtual tape drive.

W

WolfCreek A smaller capacity high–performance LSM. WolfCreek LSMs are available in 500, 750, and 1000 cartridge capacities (model numbers 9360–050, 9360–075, and 9360–100 respectively). WolfCreek LSMs can be connected by pass–thru ports to 4410, 9310, or other WolfCreek LSMs.

WolfCreek CAP The standard WolfCreek CAP contains a 20–cell magazine–style CAP and a priority CAP (PCAP). (*see also,* Cartridge Access Port (CAP), Enhanced CAP, standard CAP, WolfCreek optional CAP.)

WolfCreek optional CAP The WolfCreek optional CAP contains a 30–cell magazine–style CAP which is added to the standard WolfCreek CAP. (*see also,* Cartridge Access Port (CAP), Enhanced CAP, standard CAP, WolfCreek CAP.)

Write Tape Mark (WTM) The operation performed to record a special magnetic mark on tape. The mark identifies a specific location on the tape.

WTM *See* Write Tape Mark.

WTO Write–to–Operator.

WTOR Write–to–Operator with reply.

Symbols

υ **-software.** Microprogram. A sequence of microinstructions used to perform preplanned functions and implement machine instructions.

Numerics

4410 LSM *See* standard LSM.

9310 LSM *See* Powderhorn LSM.

9360 LSM See Wolfcreek *LSM.*

9490 Cartridge Subsystem Cartridge tape transports that provide read/write capability for 36–track recording format and extended capacity tape and provide improved performance over the 4490 Cartridge Subsystem. 9490 transports can also read data recorded in 18–track format. The StorageTek 9490 Cartridge Subsystem offers better performance (faster data transfer rate, faster load/unload) than a 3490E device.

9490EE Cartridge Subsystem A high performance tape transport that provides read/write capability for Extended Enhanced (EEtape) cartridges. It is functionally equivalent to the IBM 3490E device.

9**740 LSM** See Timberwolf *LSM.*

9840 Cartridge Subsystem A high performance tape transport system for Enterprise and Open Systems environments that reads and writes 9840 cartridges. 9840s can be defined in 10-drive and 20-drive panel configurations. The 9840 can perform as a stand-alone subsystem with a cartridge scratch loader installed, or it can be attached to a StorageTek ACS.

Index

P

R

S

T

U

V