

Virtual Tape Control System

Beyond the Basics: VTCS Leading Edge Techniques

MVS Software

Version 6.2

E22966-01



April 2011, Revision 01

Submit comments about this document to STP_FEEDBACK_US@ORACLE.COM

Copyright © 2006, 2010, 2011 Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related software documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Preface

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

Audience

This guide is for StorageTek or customer personnel who are responsible for installing configuring VTCS. See *VTCS Command and Utility Reference* for information about the following:

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

Prerequisites

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

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

About This Book

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

- ? [“Leading Edge Solutions with Management and Storage Classes” on page 1](#) is about Management and Storage Classes, which are the bread and butter of anything leading edge in VTCS land.
- ? [“Using Clustered VTSS Configurations” on page 19](#). Clustered VTSS is a powerful tool, and here’s why and how to use it.
- ? [“Doing Physical Exports and Imports” on page 81](#), which you do via the EXPORT and IMPORT utilities.

Enjoy...

What’s New in this Guide?

VTCS 6.2.0, Revisions F, G, and 01

The VTCS 6.2.0, Revisions F, G, and 01 of this guide contains technical updates and corrections.

VTCS 6.2.0, Revision E

The VTCS 6.2.0, Revision E of this guide contains information about the VTCS 6.2 enhancements described in [TABLE P-2](#).

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

This Enhancement...	...is described in...
TCP/IP CLINKs	“VTSS Native IP Support” on page 47

VTCS 6.2.0, Revision D

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

VTCS 6.2.0, Revision C

The VTCS 6.2.0, Revision C of this guide contains information about the VTCS 6.2 enhancements described in [TABLE P-2](#).

TABLE P-2 VTCS 6.2.0 Updates to VTCS Leading Edge Techniques, Revision C

This Enhancement...	...is described in...
VTSS Synchronous Replication	“Another Variation: Synchronous or Asynchronous Replication?” on page 45

Synchronous replication, which applies to only VSM4s and VSM5s, has the requirements described in [TABLE P-3](#).

TABLE P-3 Synchronous Replication Requirements for VTCS/NCS 6.2

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

VTCS 6.2.0, Revision B

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

Contents

Preface iii

Audience iii

Prerequisites iii

About This Book iv

What's New in this Guide? iv

VTCS 6.2.0, Revisions F, G, and 01 iv

VTCS 6.2.0, Revision E iv

VTCS 6.2.0, Revision D iv

VTCS 6.2.0, Revision C v

VTCS 6.2.0, Revision B v

1. Leading Edge Solutions with Management and Storage Classes 1

The Advanced Management Feature and VTCS CDS Levels 2

What are Management and Storage Classes? 3

Creating and Using VTCS Management and Storage Classes: The Basics 4

 ? To create and use VSM Management and Storage Classes: 4

Maintaining Management and Storage Classes 5

Leading Edge Techniques You Can Do with Management and Storage Classes 6

 Grouping Multiple Workloads on Shared MVCs 7

 Segregating Individual Workloads on Separate Sets of MVCs 9

 Archiving Data 11

 Archive Usage Notes 11

 Reconciling VTV Media and Location 13

 RECONcil Example 13

 RECONcil Usage Notes 14

Named MVC Pools or Not?	16
? Creating and Using Named MVC Pools	17
2. Using Clustered VTSS Configurations	19
What is Clustered VTSS?	20
Clustered VTSS Requirements	21
How Clustered VTSS Configurations Work	23
How VTSS Reconciliation Works	25
Uni-Directional and Bi-Directional CLINKs	26
How Uni-Directional VTSS Clusters Work	28
How Bi-Directional VTSS Clusters Work	29
Example: Uni-Directional Clustered VTSS - Dual ACS Configuration, 8 VCF Cards, 4 CLINKs, FICON Directors for 8 RTDs	30
? Configuring and Managing a Uni-Directional Clustered VTSS System	32
Example: Bi-Directional Clustered VTSS - Dual ACS Configuration, 8 VCF Cards, FICON Directors for 24 RTDs and 4 CLINKs	36
? Configuring and Managing a Bi-Directional Clustered System	39
Variation on a Theme: Uni-Directional or Bi-Directional?...	44
Another Variation: Synchronous or Asynchronous Replication?	45
? Implementing Synchronous Replication	45
3. VTSS Native IP Support	47
Satisfying Prerequisites	48
The TCP/IP Environment	50
Replacing IFF2 Cards with IFF3 Cards	51
Configuring the IFF3 IP Addresses	56
Configuring VTCS	65
CONFIG Utility Changes	65
Example: Clustered VTSS with TCP/IP CLINKs	66
Other VTCS Changes	69
DECOM	69
DISPLAY CLINK	69
XML Processing	69
Messages	70
Changing the IP Address of an IFF3 Card	71

4. Doing Physical Exports and Imports 81

Exporting and Importing by Management Class 82

? Example: Exporting by Management Class from the Source VSM System 82

? Example: Importing by Management Class into the Target VSM System 84

Exporting and Importing by Storage Class 86

? Example: Exporting by Storage Class from the Source VSM System 86

? Example: Importing by Storage Class into the Target VSM System 87

Leading Edge Solutions with Management and Storage Classes

The Advanced Management Feature and VTCS CDS Levels

This is a key discussion for your toolkit, focusing on two essential items you need for advanced projects. Any discussion of VTCS Leading Edge work starts with the The Advanced Management Feature, which is an optional VTCS feature, but required for the features described in [TABLE 1-1](#).

TABLE 1-1 Features Enabled by the Advanced Management Feature

For this feature...	...use this interface...	...which is described in...
Management and Storage Classes	STORclas statement MGMTclas statement MIGpol, RESTIME, CONSRC, CONTGT, REPLICAT, ARCHAGE, and ARCHPOL parameters	? “What are Management and Storage Classes?” on page 3 ? “Creating and Using VTCS Management and Storage Classes: The Basics” on page 4 ? “Leading Edge Techniques You Can Do with Management and Storage Classes” on page 6
Export/Import	EXPORT command/utility IMPORT command/utility	<i>ELS Disaster Recovery and Offsite Data Management Guide</i>
Clustered VTSS configurations	MGMTclas statement REPLICAT parameter CONFIG CLUSTER, CONFIG CLINK statements MGMTclas and STORclas statements (typical but not required)	<i>ELS Disaster Recovery and Offsite Data Management Guide</i>

The next critical item is “know your CDS VTCS level, and what it can and cannot do.” [TABLE 1-2](#) describes the CDS levels and the features they enable.

TABLE 1-2 CDS Levels for Supported VTCS Versions

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

What are Management and Storage Classes?

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

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

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

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

Let’s start with “[Creating and Using VTCS Management and Storage Classes: The Basics](#)” on [page 4](#), which is a basic procedure you can adjust for any of your business’s needs. Next, it’s on to “[Leading Edge Techniques You Can Do with Management and Storage Classes](#)” on [page 6](#). Think of this section as a gallery where you get to see a variety of options, then pick out the one that works best for you...

Creating and Using VTCS Management and Storage Classes: The Basics

You'll see this pattern a lot (TAPERREQ --> POLICY --> MGMTclas --> STORclas), so get to know it, because it's the foundation for all the information in [“Leading Edge Techniques You Can Do with Management and Storage Classes”](#) on page 6.

? To create and use VSM Management and Storage Classes:

1. **Enable the VSM Advanced Management Feature via the HSC FEATURES control statement.**
2. **Determine the definition data set that contains the STORclas and MGMTclas statements.**
MGMTclas and STORclas statements **must** reside in the same data set for cross-validation.
3. **Define Storage Classes via the STORclas control statement.**
4. **As desired, adjust the migration policies with MIGRSEL statements.**
5. **Define Management Classes with the MGMTclas control statement.**
Note that the MGMTclas control statement specifies Storage Classes on various parameters.
6. **Load the control statements with the HSC MGMTDEF command.**
7. **Specify a tape policy on the SMC POLICY command.**
8. **Specify the Policy name to VTCS on either of the following:**
 - ? The SMC TAPERREQ statement.
 - ? SMS routines that you write to the StorageTek DFSMS interface.

Maintaining Management and Storage Classes

Note the following:

- ? Always use an SMC POLICY command to assign the Management Class to mounts.
- ? You can specify a policy with either the TAPEREQ statement or SMS routines.
- ? You can use the VTVMAINT utility to change a VTV's Management Class. Also note that while you **cannot** use VTVMAINT to directly change a VTV's Storage Class, you **can** use VTVMAINT to change a VTV's Management Class, which can reference a different Storage Class.
- ? Use only the minimum Storage Classes required to define the policies you want to implement. Excessive Storage Classes can impact VSM performance due to the MVC mount/dismount overhead incurred. In addition, an MVC can only contain VTVs in a single Storage Class, so excessive Storage Classes can underuse MVC space.
- ? **If you decide to delete a Management Class definition**, run a VTV Report to make sure that the Management Class is no longer assigned to any VTVs, otherwise unpredictable results will occur!

Leading Edge Techniques You Can Do with Management and Storage Classes

The following isn't an exhaustive list, just the most common tasks you can do with Management and Storage Classes:

- ? [“Grouping Multiple Workloads on Shared MVCs” on page 7](#). This was our introductory example, and is a good one if your company owns the data center, and you want to optimize use of your available resources by:
 - ? Duplexing critical data to separate MVCs in the local and remote ACSs. In the local ACS, put it on 9840 media so I can get it back in a hurry if I need it. In the remote ACS, prefer 9940 over ZCART media for deep storage on high-capacity media.
 - ? Giving two critical jobstreams (payroll and accounting) access to these Management/Storage Classes. Result: all your payroll and accounting data is duplexed local and remote, grouped on the same set of MVCs of the appropriate media per the Storage Class specifications.
 - ? Production data is also critical, but we want it on a separate set of MVCs from the ones used for payroll and accounting data. No problem...just create another Management Class/Storage Class combination for production data, and you're done...
- ? [“Segregating Individual Workloads on Separate Sets of MVCs” on page 9](#). All you service groups, study this one carefully, because you'll probably use it a lot. Ever want to give each of your clients his/her own set of resources for billing/security purposes? Segregating workloads is the key...
- ? [“Archiving Data” on page 11](#). In this scenario, you can use VTCS to mimic HSM...in the StorageTek automated tape/virtual tape environment only. That is, you can use the ARCHAge and ARCHPol parameters of the MGMTclas statement to set an *Archive Policy* for VTVs in a Management Class.

Information Lifecycle Management (ILM), a StorageTek storage management strategy, has as its central concept the idea that data should be stored on media that matches its importance to the business enterprise and its pattern of reuse. Active, important data goes on fast-access media, and has multiple copies, while inactive, less important data is archived to high-capacity, cheaper media. Automating this process is the most cost-effective way to manage data storage. Archiving implements ILM by letting you archive inactive data. Using VTCS archiving, you can move VTVs to different media (for example, from fast-access 9840 media to high-capacity 9940 media) and a different location (for example, from a local ACS to a remote ACS for ejection/vaulting). For the details, see [“Archiving Data” on page 11](#)...

- ? [“Reconciling VTV Media and Location” on page 13](#). Think of archiving as a proactive move. You put the data on the right media at the beginning of the ILM cycle, then move it to other media as the data ages. What happens if the data ends up on the wrong media? Answer: use the RECONcil utility to move it from one Storage Class to another...

Grouping Multiple Workloads on Shared MVCs

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

```
STORCLAS NAME (LOC1) ACS (00) MEDIA (STK1R)
STORCLAS NAME (LOC2) ACS (00) MEDIA (STK1R)
STORCLAS NAME (REM1) ACS (01) MEDIA (STK2P, ZCART)
STORCLAS NAME (REM2) ACS (01) MEDIA (STK2P, ZCART)
```

FIGURE 1-1 Storage Classes for Workload Grouping

[FIGURE 1-2](#) defines the following Management Classes:

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

```
MGMT NAME (PAY) MIGPOL (LOC1, REM1)
MGMT NAME (ACCOUNT) MIGPOL (LOC1, REM1)
MGMT NAME (PROD) MIGPOL (LOC2, REM2)
```

FIGURE 1-2 Management Classes for Workload Grouping

[FIGURE 1-4](#) defines tape policies that specify virtual media and respectively assign Management Classes PAY, ACCOUNT, and PROD.

```
POLICY NAME (PPAY) MEDIA (VIRTUAL) MGMT (PAY)
POLICY NAME (PACCOUNT) MEDIA (VIRTUAL) MGMT (ACCOUNT)
POLICY NAME (PPROD) MEDIA (VIRTUAL) MGMT (PROD)
```

FIGURE 1-3 POLICY commands for Workload Grouping

Finally, [FIGURE 1-4](#) consists of TAPEREQ statements that assign policies as follows:

- ? Data sets with qualifiers of PAYROLL.** are assigned policy PPAY.
- ? Data sets with qualifiers of ACCOUNTS.** are assigned policy PACCOUNT.
- ? All other data sets are assigned policy PPROD.

```
TAPEREQ DSN(PAYROLL.** ) POLICY(PPAY)
TAPEREQ DSN(ACCOUNTS.** ) POLICY(PACCOUNT)
TAPEREQ DSN(** ) MEDIA(VIRTUAL) POLICY(PPROD)
```

FIGURE 1-4 TAPEREQ Statements for Workload Grouping

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

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

Segregating Individual Workloads on Separate Sets of MVCs

You can use Storage and Management Classes to segregate individual workloads on separate sets of MVCs. For example, the STORclas statements in [FIGURE 1-5](#) define Storage Classes LOC, CUSTA, and CUSTB1, and CUSTB2.

```
STORCLAS NAME(LOC) ACS(00) MEDIA(STK1R)
STORCLAS NAME(CUSTA) ACS(00) MEDIA(STK1R)
STORCLAS NAME(CUSTB1) ACS(00) MEDIA(STK1R)
STORCLAS NAME(CUSTB2) ACS(01) MEDIA(STK2P)
```

FIGURE 1-5 Storage Classes for Workload Segregation

[FIGURE 1-6](#) defines the following Management Classes:

- 2 Management Class CUSTA specifies the CUSTA Storage Class on the MIGPOL parameter. VTCS simplexes VTVs in this Management Classes to **only** the CUSTA Storage Class (9840 media in the local ACS), because that's what this customer wants.
- 2 Customer B wants more protection, namely duplexing to the local and remote ACSs, so Management Class CUSTB points to **both** the CUSTB1 and CUSTB2 Storage Classes.
- 2 Finally, local ACS/9840 media is just fine for your own production data, so that's what Management Class PROD does. What I'll probably also do is set up an Archive Policy for this Management Class (see "[Archiving Data](#)" on [page 11](#)) so I can move it to Deep Storage eventually.

```
MGMT NAME(CUSTA) MIGPOL(CUSTA)
MGMT NAME(CUSTB) MIGPOL(CUSTB1,CUSTB2)
MGMT NAME(PROD) MIGPOL(LOC)
```

FIGURE 1-6 Management Classes for Workload Segregation

[FIGURE 1-7](#) defines tape policies that specify virtual media and respectively assign Management Classes PAY, ACCOUNT, and PROD.

```
POLICY NAME(PCUSTA) MEDIA(VIRTUAL) MGMT(CUSTA)
POLICY NAME(PCUSTB) MEDIA(VIRTUAL) MGMT(CUSTB)
POLICY NAME(PPROD) MEDIA(VIRTUAL) MGMT(PROD)
```

FIGURE 1-7 Policies for Workload Segregation

Finally, [FIGURE 1-8](#) shows the corresponding TAPEREQ statements and policy assignments:

- ? Data sets with HLQ CUSTA are assigned policy PCUSTA.
- ? Data sets with HLQ CUSTB are assigned policy PCUSTB.
- ? All other data sets are assigned policy PPROD.

```
TAPEREQ DSN(CUSTA.** ) POLICY(PCUSTA)
TAPEREQ DSN(CUSTB.** ) POLICY(PCUSTB)
TAPEREQ DSN(** ) POLICY(PPROD)
```

FIGURE 1-8 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.

Archiving Data

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

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

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

```
TAPEREQ DSN(COMPLY.***) POLICY(PCOMPLY)
POLICY NAME(PCOMPLY) MEDIA(VIRTUAL) MGMT(COMPLY)
MGMT NAME(COMPLY) IMMED(DELETE) MIGPOL(LOC1) -
          ARCHAGE(365) ARCHPOL(REMDEEP)
STOR NAME(LOC1) ACS(00) MEDIA(STK1R)
STOR NAME(REMDEEP) ACS(01) MEDIA(STK2P)
```

FIGURE 1-9 Archiving Data

For this scenario, I did the following:

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

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

Archive Usage Notes

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

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

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

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

- 7 By not specifying the MOVEVTV parameter, you can get a report (only) that is a valuable “what if” picture of how many VTVs, MVCs, and total MBs you will process with an archive request. StorageTek **strongly recommends**, therefore, that you first run ARCHIVE **without** MOVEVTV, then adjust the job as needed **before** specifying MOVEVTV!. For more information, see *ELS Command, Control Statement, and Utility Reference*.
- 7 Because demand archive can be resource intensive, you typically run ARCHIVE during non-peak processing periods. You can also use the ARCHIVE utility to override the CONFIG RECLAIM THRESHLD, MAXMVC, and CONMVC settings to optimize archive performance. You can also specify the maximum time for the archive in minutes on the ELAPSE parameter. Note that there are several limiting factors that influence archives (for example, MAXMVC and ELAPSE). VTCS enforces the strictest limiting factor. For example, if you run ARCHIVE and specify ELAPSE equal to 5 hours and MAXMVC equal to 10 *and* VTCS archives 10 MVCs in one hour, then VTCS terminates the archive before the ELAPSE value expires.
- 7 VTCS and HSC must be active to process a ARCHIVE request except when you specify the POLICYdd parameter. POLICYdd (which forces “report only” mode) also provides an enhanced “what if” capability. You can create one or more alternate MGMTclas statements with different Archive Policies (different ARCHAge and ARCHPol values), and use POLICYdd to view the Archive Policy and resource use for each scenario.
- 7 The RECONcil utility looks like ARCHIVE because RECONcil also moves VTVs from one Storage Class to another (that is, moves them from one MVC media to another and/or moves them from one ACS to another). Think of ARCHIVE as proactive and RECONcil as reactive, and you’ll see the difference, as described in [“Reconciling VTV Media and Location” on page 13...](#)

So let’s say the 365 days is up, and the internal auditors haven’t shown up. Time to go to archive. [FIGURE 1-10](#) shows example JCL to run ARCHIVE as follows:

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

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

FIGURE 1-10 Example JCL for the ARCHIVE utility

Tip – The MOVEVTV parameter also gives you a report, so you can see how well (or not) you did. If your tuning parameters didn’t archive the everything you wanted to archive, then adjust your job and rerun...

Reconciling VTV Media and Location

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

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

For details of the reconciling process, see [“RECONcil Example” on page 13](#).

RECONcil Example

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

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

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

FIGURE 1-11 Storage Classes/Management Classes for Workload Segregation

According to [FIGURE 1-11](#), everything in Management Class PROD should have ended up on 9840 media in the local ACS...but in fact, they're all on 9940 media in the remote ACS...almost as if they were in the wrong Storage Class...

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

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

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

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

Now I can run RECONcil as shown in [FIGURE 1-10](#) as follows:

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

```
//RECONCIL EXEC PGM=SLUADMIN
//STEPLIB DD DSN=hlq.SEALINK,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
      RECON MGMT (PROD) MAXMVC (60)
              CONMVC (8) ELAPSE (360) MOVEVTV
```

FIGURE 1-12 Example JCL for the RECONcil utility

RECONcil Usage Notes

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

- ? MGMTclas to move the VTVs to the Storage Class(es) specified by the MIGpol parameter. This is what we did in [“RECONcil Example” on page 13](#). Management Class points to the wrong Storage Class, get it pointed toward the right Storage Class, run RECONcil against the updated Management Class.
- ? STORclas to move the VTVs to the specified Storage Class(es). I’d probably use this when an ACS is unavailable for a considerable period of time.
- ? MVC to reconcile the VTVs on a list or range of MVCs. The VTVs are moved to the Storage Class(es) specified by the MIGpol parameter of the MGMTclas statement(s) for the VTVs. I’d use this or the VTV option as onesies/twosies kind of fix-ups.
- ? VTV to reconcile a list or range of VTVs. The VTVs are moved to the Storage Class(es) specified by the MIGpol parameter of the Management Class(es) for the VTVs.

Note –

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

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

- ? By not specifying the MOVEVTV parameter, you can get a report (only) that is a valuable “what if” picture of how many VTVs, MVCs, and total MBs you will process with a reconcile request. StorageTek **strongly recommends**, therefore, that you first run RECONcil **without** MOVEVTV, then adjust the job as needed **before** specifying MOVEVTV!. For more information, see *ELS Command, Control Statement, and Utility Reference*.

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

Named MVC Pools or Not?

Named MVC Pools are the right tool for the job for all of your service groups: You can use Named MVC Pools to give an application ownership of the MVCs in the named pool. For example, a service group might elect to use Named MVC Pools where its customers have legal requirements to buy and own a group of MVCs.

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

- ? [“Grouping Multiple Workloads on Shared MVCs” on page 7](#)
- ? [“Segregating Individual Workloads on Separate Sets of MVCs” on page 9](#)

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 only a single MVC pool.

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

If you **do** elect to use Named MVC Pools, go to [“Creating and Using Named MVC Pools” on page 17](#).

? Creating and Using Named MVC Pools

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

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

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

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

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

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

Caution – A Named MVC Pool *must* contain media of the type specified in the corresponding STORCLAS MEDIA parameter (see [Step 5](#)); otherwise, VTCS will issue a “no MVCs available” message during MVC selection.

2. **Run the VT MVCDEF command to activate the updated data set.**
3. **Enable the VSM Advanced Management Feature via the HSC FEATURES control statement.**

The Advanced Management Feature is required for the Storage Classes you define in [Step 4](#).

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

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

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

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

For more information, see “Creating and Using VTCS Management and Storage Classes: The Basics” on [page 4](#).

Using Clustered VTSS Configurations

Ever wish you could copy VTVs from one VTSS to another? Well, you can, thanks to the magic of Clustered VTSSs. Clustered VTSS is a powerful tool for applications such as but not limited to DR (Disaster Recovery) solutions. As you've probably guessed, however, with Clustered configurations, *Some Assembly is Required*. So let's start with the details of what VTSS Clusters are and how they work:

- ? [“What is Clustered VTSS?” on page 20](#)
- ? [“Clustered VTSS Requirements” on page 21](#)

Also note that you can now cluster VTSSs using TCP/IP as well as FICON. For more information, see [“VTSS Native IP Support” on page 47](#).

Next, we'll proceed to some real world examples:

- ? [“Example: Uni-Directional Clustered VTSS - Dual ACS Configuration, 8 VCF Cards, 4 CLINKs, FICON Directors for 8 RTDs” on page 30](#)
- ? [“Example: Bi-Directional Clustered VTSS - Dual ACS Configuration, 8 VCF Cards, FICON Directors for 24 RTDs and 4 CLINKs” on page 36](#)
- ? ...and, as added bonuses, [“Variation on a Theme: Uni-Directional or Bi-Directional?...” on page 44](#) and [“Another Variation: Synchronous or Asynchronous Replication?” on page 45](#).

What is Clustered VTSS?

A VTSS Cluster is two VTSSs connected via FICON or ESCON ports. A Cluster enables data (VTVs) to be replicated from one VTSS system to another VTSS within the same cluster. Clustered configurations, therefore, provide redundancy/high availability because if one VTSS within a cluster has an outage, workload automatically continues on the remaining VTSS(s) with minimal disruption (that is, all data that was replicated remains available without requiring recalls from MVCs).

A cluster can span geographic locations. A cluster, however, **must be within a single TapePlex** (controlled by a single CDS).

A VTV can be replicated (copied) from one VTSS to another either:

- ? Asynchronously to the VTV creation – scheduled to complete as soon as possible after the VTV dismount
- ? Synchronously with the VTV creation. The VTV dismount will not complete until the replication is complete.

Note – The VTSS estimates if the VTV can be synchronously replicated within 40 minutes. If this is not possible, the VTV is asynchronously replicated.

The connections between VTSS systems within a cluster can be either uni-directional, where data (VTVs) flows only one way or bi-directional, where data (VTVs) can flow in both directions. The CONFIG utility specifies whether a cluster is uni- or bi-directional, and a VTVs Management Class determines its replication policy, if any, and whether the replication is done synchronously or asynchronously.

So both vaulting MVCs and VTV replication can facilitate a Disaster Recovery/Business Continuity solution. VTV replication, however, is superior as a High Availability solution because with replication:

- ? Data can be backed up synchronously.
- ? Recent data, which has been replicated to a “Recovery” VTSS, can be restored more quickly because you don’t have to mount MVCs.

Clustered VTSS Requirements

TABLE 2-1 Clustered VTSS Requirements

Component	Requirement
2 VTSSs within a cluster (ESCON Interfaces)	The Primary and Secondary VTSSs can be any combination of VSM3 and VSM4 where the Secondary can be of any capacity. VSM5s do not have ESCON interfaces, and cannot be in a cluster with other VTSSs that use ESCON.
2 VTSSs within a cluster (FICON Interfaces)	The Primary and Secondary VTSSs can be any combination of VSM4 and VSM5 where the Secondary can be of any capacity. For example, all of the following are valid: ? Primary VSM5, Secondary VSM4 ? Primary VSM5, Secondary VSM5 ? Primary VSM4, Secondary VSM4 ? Primary VSM4, Secondary VSM5 (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	? The Advanced Management Feature (to enable the REPLICAT parameter of the MGMTclas statement)
VTDs reserved for clustering	In Clustered VTSS configurations, you must ensure that the first 16 VTDs in each VTSS (0-F) are reserved for clustering. These devices must be OFFLINE to MVS, and their paths must be online to each HSC server host. This also applies to any VTSSs involved in Cross TapePlex Replication. VTCS does not register the first 16 VTDs with SMC/HSC, which prevents mounting VTVs on these VTDs.
RTDs	In dual-ACS environments, the same device types must be represented in the RTDs attached to each ACS so that data migrated by one VTSS can be recalled by the other VTSS. The number of MVCs, the media type and location used for the migration is determined by the MIGPOL parameter of the MGMTclas statement. A minimum of two of each drive type in each ACS is strongly recommended to allow VTCS to swap MVCs if a transient drive or media failure occurs.

Synchronous replication, which applies to only VSM4s and VSM5s, has the requirements described in [TABLE 2-2](#).

TABLE 2-2 Synchronous Replication Requirements for VTCS/NCS

Synchronous replication requires...	..the following VSM4/VSM5 microcode...	...and CDS level...
FICON ports for the CLINKs	D02.03.00.00 or higher	“F” or higher

How Clustered VTSS Configurations Work

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

- ? Clusters can be either Uni-Directional or Bi-Directional depending on the CLINK statements.
- ? The Secondary VTSS (or the second Peer) can either be at the same physical location as the Primary (or first Peer) or at a remote location.
- ? The CONFIG CLUSTER statement specifies the VTSSs that form the Cluster.
- ? The CONFIG CLINK statement defines the CLINKs that connect the VTSSs. The way you write the CLINK statements determines whether the replication is uni-directional or bi-directional. For examples, see [FIGURE 2-5 on page 33](#) and [FIGURE 2-14 on page 40](#).
- ? The MGMTclas REPLICAT parameter (which requires the Advanced Management Feature) identifies the Management Class that contains the VTVs that VSM *replicates* (copies) from one VTSS in the Cluster to the other.

Note – The CONFIG GLOBAL REPLICat parameter now specifies when to replicate a VTV as follows:

REPLICat

specifies when VSM replicates the VTV.

ALWAYS

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

CHANGED

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

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

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

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

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

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

Because VTCS immediately migrates (with KEEP) replicated VTVs regardless of the MGMTclas IMMDELAY setting, StorageTek **strongly recommends** that you **do not** explicitly set a MGMTclas IMMDELAY policy for replicated VTVs. If you do, VTCS honors the explicit immediate migrate request, and immediately migrates the affected VTV from whichever VTSS is first capable of performing the migration (that is, the first VTSS that has

a resident VTV copy and an available RTD to satisfy the migrate). Setting an explicit MGMTclas IMMDELAY policy, therefore, is redundant and may interfere with optimal VTV replication and migration.

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

- ⌚ **Note that** the replication requirements of data is determined following a dismount, **not** a recall. Merely recalling a VTV will not cause a replicate – so demand recall, MVCdrain and reclaim will not cause a replicate. However, if the VTV is recalled and mounted on a VTD, at dismount time it will be replicated to the Secondary or Peer VTSS.
- ⌚ In dual-ACS environments, the same device types must be represented in the RTDs attached to each ACS so that data migrated by one VTSS can be recalled by the other VTSS. The number of MVCs, the media type and location used for the migration is determined by the MIGPOL parameter of the MGMTclas statement.
- ⌚ A Cluster can support different workloads in each of four operating modes. For example, only Full-Function Clusters can support active replication, but in Degraded Primary Mode, you can vary the Secondary’s VTDs online to MVS to take over the workload. You can use Query to display Cluster, Cluster link, VTV replication, and VTSS status. You can use VARY VTSS to change VTSS states and VARY CLink to change CLINK states.

Note – In a client/server environment (either MVS/CSC and LibraryStation or SMC client/server), VTCS cannot determine if long running jobs are active on the client hosts. Therefore, in addition to varying the Primary offline, you must also either explicitly vary the Primary’s VTDs offline to MVS or ensure that there are no tape jobs active on the client hosts. In Clustered VTSS configurations, the Clinks to the Primary VTSS should be varied offline to stop replication processing.

How VTSS Reconciliation Works

- ? Whenever a clustered VTSS-pair resumes the full function state, VTCS reconciles the contents of the two VTSSs. This occurs either during VTCS initialization or when a VTSS goes online and its partner VTSS is also online.
- ? Reconciliation consists of either deleting or migrating and deleting VTVs (or replicating a VTV if this had not previously completed successfully). That is, recall is not involved in reconciling VTSS contents.

For example, in a uni-directional cluster with a VTV resident in the receiver but not the sender VTSS, VTCS deletes the VTV from the receiver (after ensuring that all required MVC copies have been made). This avoids a recall to the sender.

Similarly, in a uni-directional cluster with a VTV resident in the sender but not the receiver, VTSS, VTCS replicates the VTV to the receiver instead of recalling it from an MVC.

- ? The reconcile process assumes that if a replicated VTV is resident on the sender VTSS, then it is a valid copy. If the copy on the receiver is different, VTCS deletes it.
- ? To maintain consistent reconcile actions in a bi-directional cluster, the VTSS in which the VTV is or was last resident (as indicated by the CDS VTV record), is considered to be the sender VTSS. Reconcile processing is as described above for uni-directional clusters.

Uni-Directional and Bi-Directional CLINKs

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

For example, [FIGURE 2-1](#) shows 2 CLINK ports on each VTSS configured for Bi-Directional Clustering. On the Primary VTSS (VTSS1), the CLINK CIPs/FIPs are configured in **Nearlink Mode**, while on the Secondary VTSS (VTSS2), the CIPs/FIPs are configured in **Host Mode**. Figure 11 on page 26 shows a Uni-Directional Cluster attached to a dual ACS. In a Uni-Directional Cluster, the Primary VTSS replicates VTVs to the Secondary VTSS.

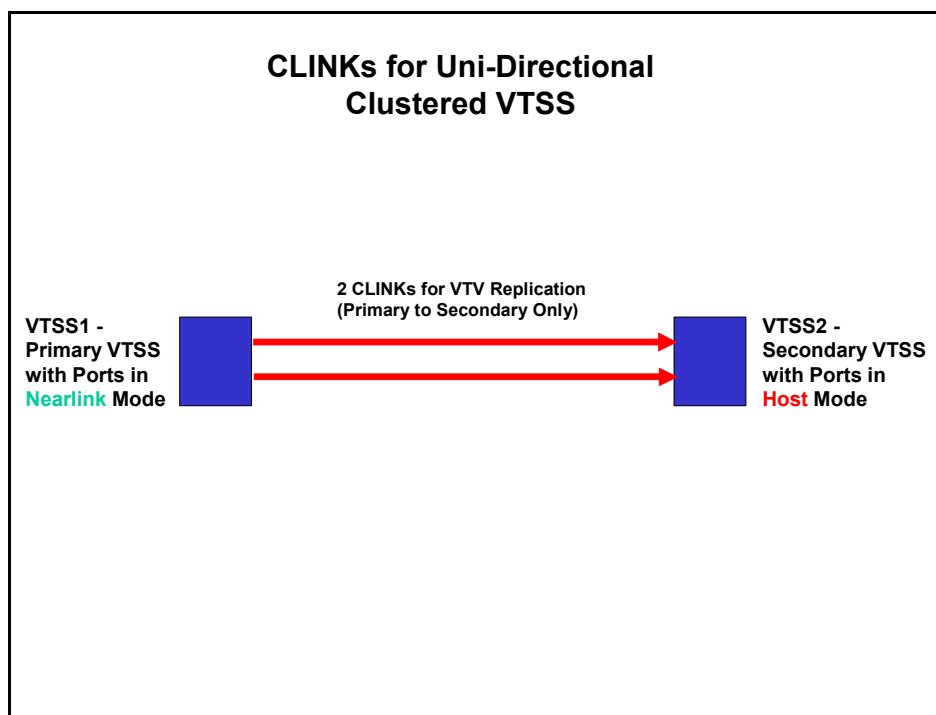


FIGURE 2-1 CLINKs for Uni-Directional Clustered VTSS

FIGURE 2-2 shows 2 CLINK ports on each VTSS configured for Bi-Directional Clustering. Each Peer VTSS (VSMR1 and VSMR2), must have **both** of the following:

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

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

Caution – In FIGURE 2-2, also note that each CLINK is attached to **the same Storage Cluster** on each VTSS, **which is a requirement**. Failure to configure in this manner can produce Replicate, Channel, and Communication errors!

Therefore, as shown, the sending (Nearlink mode) CLINK port on VSMR1 is on Storage Cluster 1, and it connects to a receiving (Host Mode) CLINK port, **also on Storage Cluster 1** on VSMR2. Similarly, a sending CLINK port on Storage Cluster 0 of VSMR2 connects to a receiving CLINK port on Storage Cluster 0 of VSMR1.

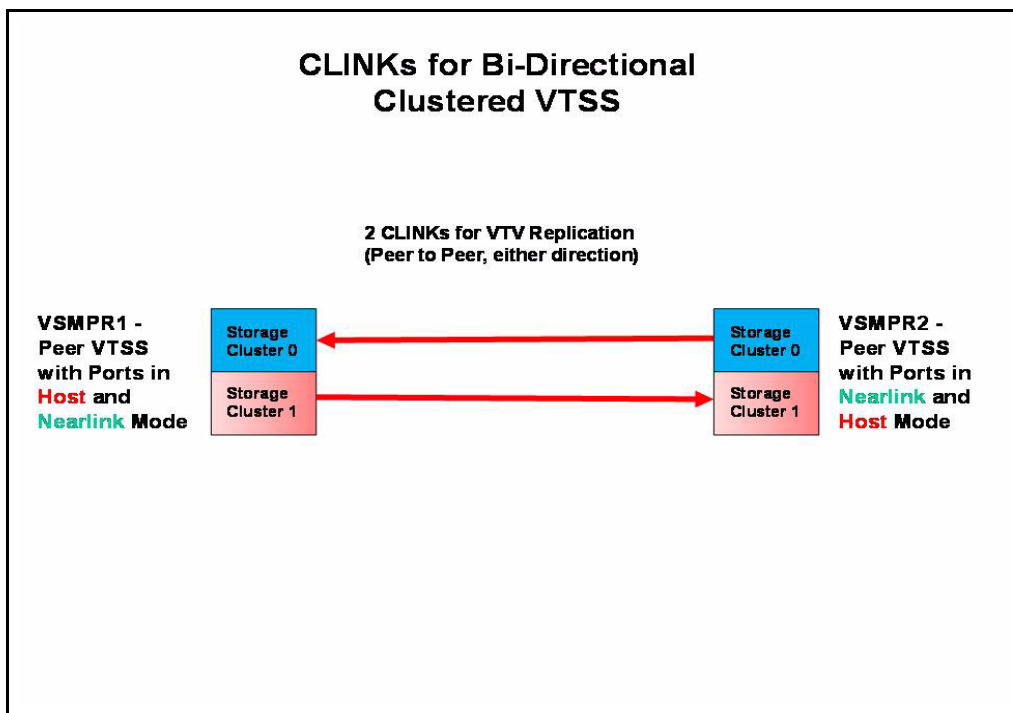


FIGURE 2-2 CLINKs for Bi-Directional Clustered VTSS

How Uni-Directional VTSS Clusters Work

- ? The Secondary can receive both replicated VTVs from the Primary and non-replicate production workload by any of the standard routing methods (for example, TAPEREQs). You need to vary the VTDs in the Secondary online to MVS so that the Secondary can accept production work. You **cannot** vary online to MVS the VTD addresses used by the CLINK terminations as described in [“How Clustered VTSS Configurations Work” on page 23](#).
- ? A VTV with replication enabled is allocated to an online Primary VTSS unless none are available; in that case, the VTV is allocated to an online Secondary VTSS. If no online Secondary VTSSs are available, the VTV is allocated to a non-cluster VTSS. A VTV without replication can be allocated to any online VTSS including the Secondary of a Full-Function Cluster.
- ? At dismount time, a VTV with replication enabled that resides on a Full-Function Cluster is queued for replication to the Secondary VTSS. If a VTV with replication enabled is dismounted from a VTD in a VTSS that is not part of a Full-Function Cluster, the VTV is queued for immediate migration.

When the Secondary VTSS receives a replicated VTV from the Primary VTSS, the VTV is then immediately migrated (with the KEEP option) regardless of Immediate Migrate Management Class settings for this VTV.

- ? **Both the Primary and the Secondary VTSS** can manage all space reclamations.

How Bi-Directional VTSS Clusters Work

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

- ? In a Bi-Directional Cluster, each of the Peer VTSSs can receive production work via the standard routing methods (for example, TAPEREQs). You need to vary the VTDs in both VTSSs online to MVS so that each can accept production work. However, **note that** you **cannot** vary online to MVS the VTD addresses used by the CLINK connections as described in [“How Clustered VTSS Configurations Work” on page 23](#).
- ? **In a Bi-Directional Cluster**, a VTV with replication enabled is allocated to either of the Peer VTSSs. If one of the two Peer VTSSs is either offline or quiesced, production workload can run on the remaining online VTSS. VTVs requiring replication, however, are allocated to the remaining VTSS only if no other Full-Function clusters are available and suitable. In this case, replicate VTVs are migrated immediately with keep and queued for replication when the other VTSS comes online.
- ? **In a Bi-Directional Cluster**, at dismount time, a VTV with replication enabled that resides on a Full-Function Cluster is queued for replication to the other Peer VTSS. If a VTV with replication enabled is dismounted from a VTD in a VTSS that is not part of a Full-Function Cluster, the VTV is queued for immediate migration. **Note that** the replication requirements of data is determined following a dismount, **not** a recall. Merely recalling a VTV will not cause a replicate – so demand recall, MVCdrain and reclaim will not cause a replicate. However, if the VTV is recalled and mounted on a VTD, at dismount time it will be replicated to the Secondary VTSS unless you specify REPLICAT(CHANGED) (the recommended option), which will cause the VTV to be replicated again only if the data is changed.
- ? **Both Peer VTSSs** can manage all space reclamations.

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

FIGURE 2-3 shows an example of a Uni-Directional Clustered VTSS Dual ACS system. In this example, I only have one MVS host, but it's putting out a lot of critical data that I want protected to the max using two brand new VSM4s that I just purchased. No problem: VTSS1 is the Primary VTSS, and it's connected to the Secondary (VTSS2) via Cluster Links (CLINKs). If the Management Class for a VTV specifies replication, presto, when the VTV arrives in VTSS1, it is *replicated* (copied) to VTSS2, and also immediately migrated (with KEEP). Result: I have increased data availability (there's a copy of the VTV in each VTSS in case one fails) *and* data protection (the VTV is also on square tape in case both VTSSs go offline). Clustered VTSS is a great solution, therefore, for business continuity and business resumption.

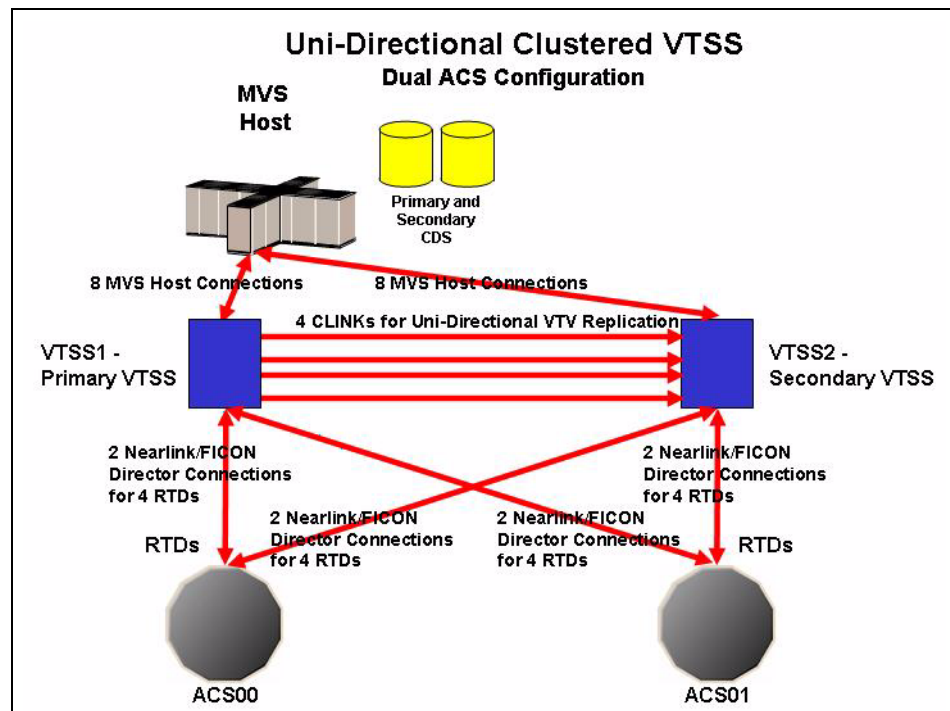


FIGURE 2-3 Dual ACS Uni-Directional Clustered VTSS Configuration

Now it's time to take a look at the hardware for this Clustered configuration. FIGURE 2-4 shows CONFIG channel interface identifiers for a VSM4 with 8 VCF cards. In this configuration, we've allocated:

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

- 7 4 ports for CLINK connections to form a Uni-Directional VTSS Cluster, and 8 ports to host connections. To form the clustered VTSS, we'll have two VSM4s (VTSS1 and VTSS2) configured identically as shown in [FIGURE 2-4](#).

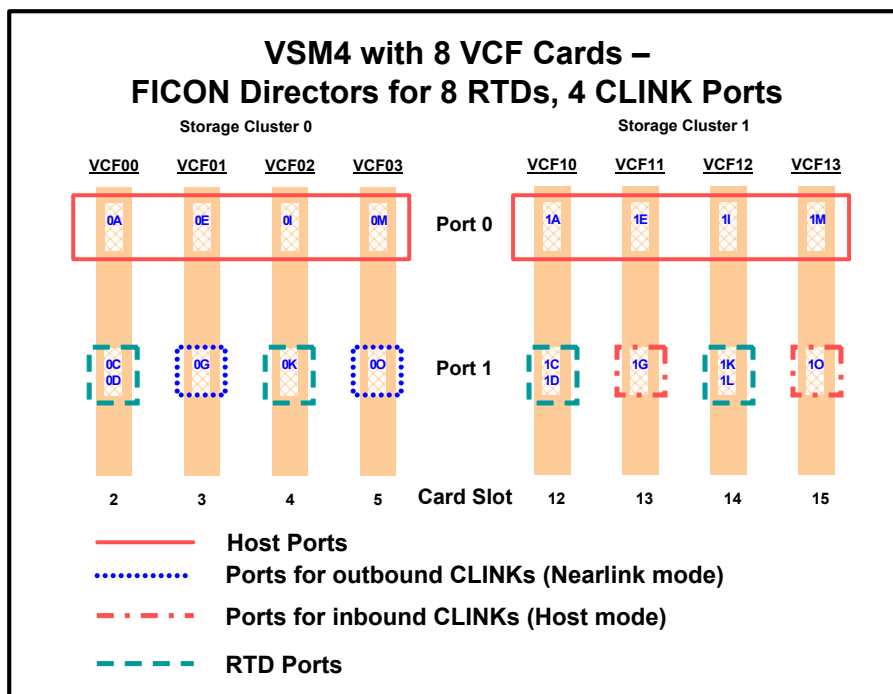


FIGURE 2-4 VSM4 with 8 VCF Cards, 8 Host Ports, FICON Directors for 8 RTDs, 4 CLINK Ports

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

? Configuring and Managing a Uni-Directional Clustered VTSS System

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

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

FIGURE 2-5 shows example CONFIG JCL to define a Uni-Directional Cluster of two VSM4s (VTSS1 and VTSS2) as shown in Figure 11 on page 26. **Note that:**

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

```

//CREATECFG                                EXEC PGM=SWSADMIN, PARM= 'MIXED'
//STEPLIB                                DD DSN=hlq.SLSLINK, DISP=SHR
//SLSCNTL                                DD DSN=FEDB.VSMLMULT.DBASEPRM, DISP=SHR
//SLSCNTL2                              DD DSN=FEDB.VSMLMULT.DBASESEC, DISP=SHR
//SLSSTBY                                DD DSN=FEDB.VSMLMULT.DBASESTBY, DISP=SHR
//CFG22202                              DD DSN=FEDB.VSMLMULT.CFG22202, DISP=SHR
//SLSPRINT                              DD SYSOUT=*
//SLSIN                                  DD *

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

```

FIGURE 2-5 CONFIG example: Dual ACS Uni-Directional Clustered VTSS System, VSM4 FICON Back-End

3. Enable the Advanced Management Feature.

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

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

```
CONFIG GLOBAL REPLICAT=CHANGED
```

FIGURE 2-6 CONFIG GLOBAL Setting for VTV Replication

In [FIGURE 2-7](#), CONFIG GLOBAL REPLICAT=CHANGED specifies:

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

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

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

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

FIGURE 2-7 Management Class for VTV Replication

Note –

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

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

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

FIGURE 2-8 Storage Classes for Replicated, Migrated VTVs

In [FIGURE 2-8](#), the STORclas statement defines Storage Classes REPLSTR1 and REPLSTR2 referenced in the MIGPOL parameter in [Step 5](#) on page 30. **Also note** that the MIGRATE parameters on the Storage Classes specify that the VTSS receiving the replicated VTV...in this case VTSS2, the Secondary, does the migration to both ACSs. This is a handy way of ensuring that the Secondary functions as the “migrate engine.”

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

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

FIGURE 2-9 MGMTDEF Command to Load Statements

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

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

FIGURE 2-10 TAPEREQ Statement to Route Critical Data, Assign Management Class VSMREPL

In [FIGURE 2-10](#), the TAPEREQ statement specifies:

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

Caution – To replicate VTVs, **both VTSS1 and VTSS2** must be varied online to VTCS so that VTCS can send control commands to both VTSSs. See [“How Clustered VTSS Configurations Work” on page 23](#) for more information.

Note – Also note the following:

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

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

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

...and you're home free!

Example: Bi-Directional Clustered VTSS - Dual ACS Configuration, 8 VCF Cards, FICON Directors for 24 RTDs and 4 CLINKs

FIGURE 2-11 shows an example of a Bi-Directional Clustered VTSS Dual ACS system. This system is very similar to the uni-directional example, but goes one step further: There are two MVS hosts sharing a CDS, and everything in the picture is cross-connected. I basically have sites mirroring each other for the ultimate in data availability and protection. To make this happen...that is, to make it bi-directional...I have to configure the two VTSSs as peers via the CLINK statements.

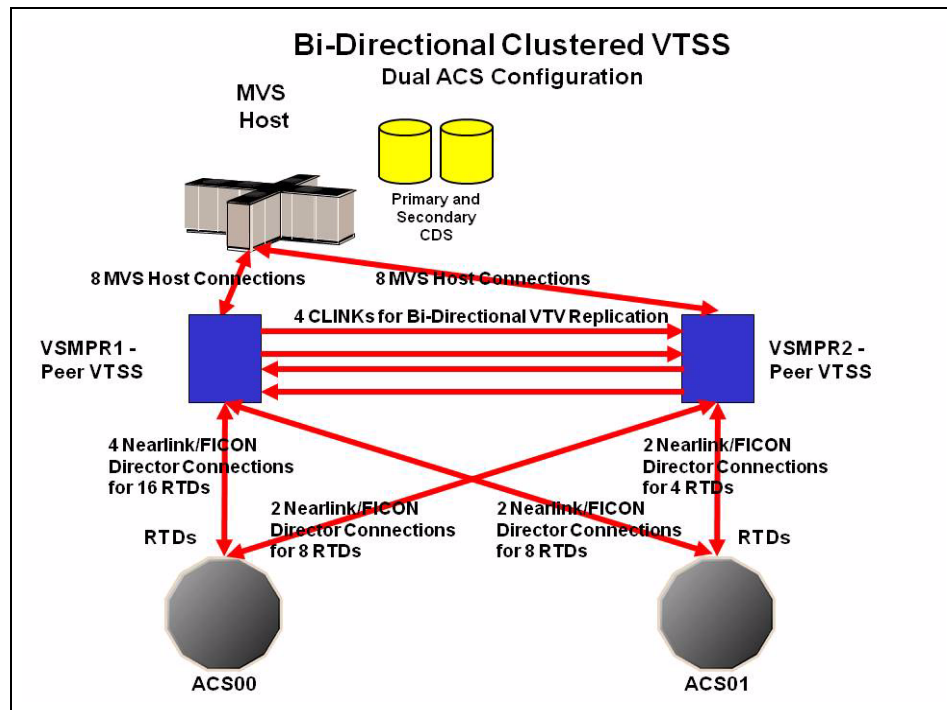


FIGURE 2-11 Dual ACS Bi-Directional Clustered VTSS Configuration

Note –

- 2 Bi-Directional Clustering **requires** VTCS 6.1 and above! 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.
- 2 This configuration is shown with the feature that enables up to a total of 16 simultaneous NearLink I/O transfers, which can be spread across multiple targets on as many as 14 NearLink ports, and up to a total of 2 simultaneous NearLink I/O transfers per port. This feature requires VTSS microcode D02.06.00.00 or higher.

FIGURE 2-12 shows CONFIG channel interface identifiers for VSMRP1 shown in FIGURE 2-11 on page 36. In this configuration, we've allocated:

- ? 8 Host ports.
- ? 6 ports for RTDs. The RTD ports are all connected to FICON directors, each of which is attached to 4 RTDs, so the CHANID identifiers for all 4 RTDs are shown on each port. This allows Back-End connection to 24 RTDs, although only one RTD per port/Director can be active at a time.
- ? 4 ports via FICON directors. Two are Nearlink for the originator, Two are host mode for the terminator for CLINK connections to form a Bi-Directional VTSS Cluster.

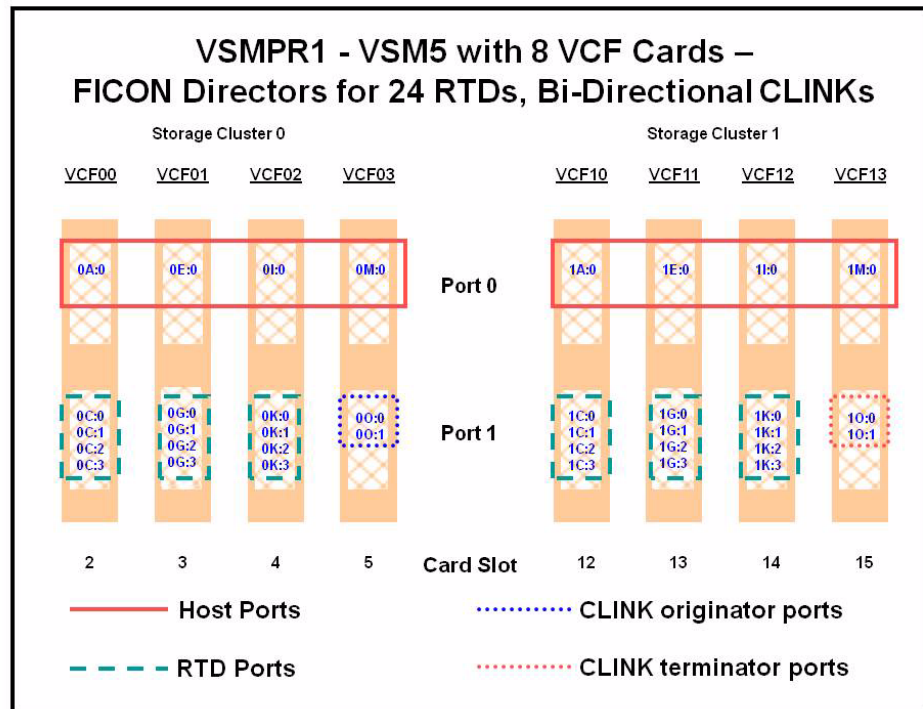


FIGURE 2-12 VSMRP1 - VSM5 with 8 VCF Cards, 8 Host Ports, FICON Directors for 24 RTDs, 4 CLINKs

FIGURE 2-13 shows CONFIG channel interface identifiers for a VSMR1, a VSM5 in a Bi-directional Cluster with 8 VCF cards and the Maximum 32 RTDs feature enabled. In this configuration, we've allocated:

- ? 8 Host ports.
- ? 6 ports for RTDs. The RTD ports are all connected to FICON directors, each of which is attached to 4 RTDs, so the CHANID identifiers for all 4 RTDs are shown on each port. This allows Back-End connection to 24 RTDs, although only one RTD per port/Director can be active at a time.
- ? 4 ports via FICON directors. Two are Nearlink for the originator, Two are host mode for the terminator for CLINK connections to form a Bi-Directional VTSS Cluster.

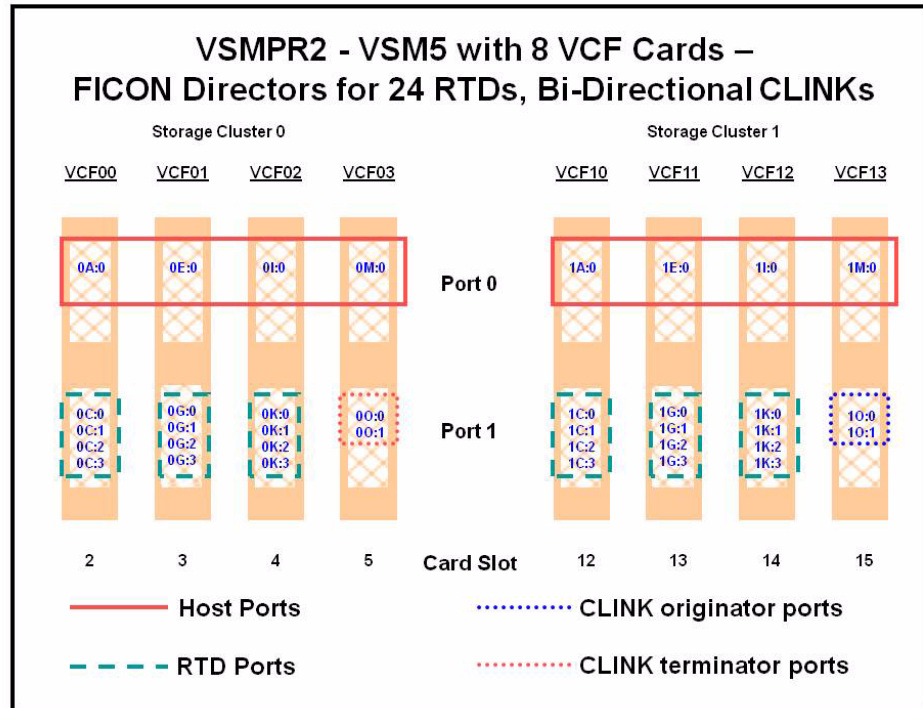


FIGURE 2-13 VSMR2 - VSM5 with 8 VCF Cards, 8 Host Ports, FICON Directors for 24 RTDs, 4 CLINKs

Caution – As shown in FIGURE 2-2 on page 27 each CLINK must be attached to **the same Storage Cluster** on each VTSS, **which is a requirement**. Failure to configure in this manner can produce Replicate, Channel, and Communication errors!

So as shown, the Nearlink ports (CLINK originators) on VSMR1 are on Storage Cluster 0 and the Host ports (CLINK terminators) on VSMR2 are also on Storage Cluster 0. The same is true for the CLINK connections for data flowing in the other direction; they are both on Storage Cluster 1.

? Configuring and Managing a Bi-Directional Clustered System

To configure and manage the Bi-Directional Clustered system shown in [FIGURE 2-11 on page 36](#), do the following:

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

[FIGURE 2-14](#) shows example CONFIG JCL to define a Bi-Directional Cluster of two VSM4s (VSMPR1 and VSMPR2) as shown in Figure 11 on page 26. **Note that:**

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

- There are CLINK statements using the sending (Nearlink Mode) ports of **both VTSSs** to enable the Cluster as Bi-Directional and they follow the rule of “connect using the same Storage Cluster on each VTSS for the sending and receiving ports of each CLINK”.

```
//CREATECF 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 VTVATTR=SCRATCH RECALWER=YES LOCKSTR=VTCS_LOCKS
REPLICAT=ALWAYS VTVPAGE=LARGE SYNCHREP=YES MAXRTDS=32
RECLAIMTHRESHLD=70 MAXMVC=40 START=35
RECLAIMTHRESHLD=70MAXMVC=40 START=35
VTVOL LOW=905000 HIGH=999999 SCRATCH
VTVOL LOW=C00000 HIGH=C25000 SCRATCH
VTVOL 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=VPR12A00 DEVNO=2A00 CHANIF=0C:0
RTD NAME=VPR12A01 DEVNO=2A01 CHANIF=0C:1
RTD NAME=VPR12A02 DEVNO=2A02 CHANIF=0C:2
RTD NAME=VPR12A03 DEVNO=2A03 CHANIF=0C:3
RTD NAME=VPR12A04 DEVNO=2A04 CHANIF=0G:0
RTD NAME=VPR12A05 DEVNO=2A05 CHANIF=0G:1
RTD NAME=VPR12A06 DEVNO=2A06 CHANIF=0G:2
RTD NAME=VPR12A07 DEVNO=2A07 CHANIF=0G:3
RTD NAME=VPR12A08 DEVNO=2A08 CHANIF=0K:0
RTD NAME=VPR12A09 DEVNO=2A09 CHANIF=0K:1
RTD NAME=VPR12A0A DEVNO=2A0A CHANIF=0K:2
RTD NAME=VPR12A0B DEVNO=2A0B CHANIF=0K:3
RTD NAME=VPR13A00 DEVNO=3A00 CHANIF=1C:0
RTD NAME=VPR13A01 DEVNO=3A01 CHANIF=1C:1
RTD NAME=VPR13A02 DEVNO=3A02 CHANIF=1C:2
RTD NAME=VPR13A03 DEVNO=3A03 CHANIF=1C:3
RTD NAME=VPR13A04 DEVNO=3A04 CHANIF=1G:0
RTD NAME=VPR13A05 DEVNO=3A05 CHANIF=1G:1
RTD NAME=VPR13A06 DEVNO=3A06 CHANIF=1G:2
RTD NAME=VPR13A07 DEVNO=3A07 CHANIF=1G:3
RTD NAME=VPR13A08 DEVNO=3A08 CHANIF=1K:0
RTD NAME=VPR13A09 DEVNO=3A09 CHANIF=1K:1
RTD NAME=VPR13A0A DEVNO=3A0A CHANIF=1K:2
RTD NAME=VPR13A0B DEVNO=3A0B CHANIF=1K:3
VTD LOW=9900 HIGH=99FF
```

FIGURE 2-14 CONFIG example: Dual ACS Bi-Directional Clustered VTSS System (Part 1)

```

VTSS NAME=VSMR2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=VPR22B00 DEVNO=2B00 CHANIF=0C:0
RTD NAME=VPR22B01 DEVNO=2B01 CHANIF=0C:1
RTD NAME=VPR22B02 DEVNO=2B02 CHANIF=0C:2
RTD NAME=VPR22B03 DEVNO=2B03 CHANIF=0C:3
RTD NAME=VPR22B04 DEVNO=2B04 CHANIF=0G:0
RTD NAME=VPR22B05 DEVNO=2B05 CHANIF=0G:1
RTD NAME=VPR22B06 DEVNO=2B06 CHANIF=0G:2
RTD NAME=VPR22B07 DEVNO=2B07 CHANIF=0G:3
RTD NAME=VPR22B08 DEVNO=2B08 CHANIF=0K:0
RTD NAME=VPR22B09 DEVNO=2B09 CHANIF=0K:1
RTD NAME=VPR22B0A DEVNO=2B0A CHANIF=0K:2
RTD NAME=VPR22B0B DEVNO=2B0B CHANIF=0K:3
RTD NAME=VPR23B00 DEVNO=3B00 CHANIF=1C:0
RTD NAME=VPR23B01 DEVNO=3B01 CHANIF=1C:1
RTD NAME=VPR23B02 DEVNO=3B02 CHANIF=1C:2
RTD NAME=VPR23B03 DEVNO=3B03 CHANIF=1C:3
RTD NAME=VPR23B04 DEVNO=3B04 CHANIF=1G:0
RTD NAME=VPR23B05 DEVNO=3B05 CHANIF=1G:1
RTD NAME=VPR23B06 DEVNO=3B06 CHANIF=1G:2
RTD NAME=VPR23B07 DEVNO=3B07 CHANIF=1G:3
RTD NAME=VPR23B08 DEVNO=3B08 CHANIF=1K:0
RTD NAME=VPR23B09 DEVNO=3B09 CHANIF=1K:1
RTD NAME=VPR23B0A DEVNO=3B0A CHANIF=1K:2
RTD NAME=VPR23B0B DEVNO=3B0B CHANIF=1K:3
VTD LOW=9900 HIGH=99FF
CLUSTER NAME=CLUSTER1 VTSSs (VSMR1,VSMR2)
CLINK VTSS=VSMR1 CHANIF=00:0
CLINK VTSS=VSMR1 CHANIF=00:1
CLINK VTSS=VSMR2 CHANIF=10:0
CLINK VTSS=VSMR2 CHANIF=10:1

```

FIGURE 2-15 CONFIG example: Dual ACS Bi-Directional Clustered VTSS System (Part 2)

3. Enable the Advanced Management Feature.

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

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

```
CONFIG GLOBAL REPLICAT=CHANGED
```

FIGURE 2-16 CONFIG GLOBAL Setting for VTV Replication

As with the uni-directional example, in [FIGURE 2-16](#), we use CONFIG GLOBAL REPLICAT=CHANGED.

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

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

FIGURE 2-17 Management Class for VTV Replication

[FIGURE 2-17](#) should look familiar...replicate VTVs only if changed and not in the other VTSS in the Cluster, migrate duplexed to ACSs 01 and 00 by Storage Classes you will create in [Step 6](#).

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

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

FIGURE 2-18 Storage Classes for Replicated, Migrated VTVs

In [FIGURE 2-18](#), the STORclas statement defines Storage Classes REPLSTR1 and REPLSTR2 referenced in the MIGPOL parameter in [Step 5](#). **Also note** that, to optimize VTSS and RTD resources, the MIGRATE parameters on the Storage Classes allow migrates to come from either VTSS. This is a typical strategy for Bi-Directional, or Peer to Peer VTSS Clusters.

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

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

FIGURE 2-19 MGMTDEF Command to Load Statements

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

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

FIGURE 2-20 TAPEREQ Statement to Route Critical Data, Assign Management Class VSMREPL

In [FIGURE 2-20](#), the TAPEREQ statement specifies:

- ? Route data sets with HLQ mask *.PAYROLL.** to VSM...
- ?and assign Management Class VSMREPL that you enabled in [Step 5](#).

Caution – To replicate VTVs, **both VSMRP1 and VSMRP2** must be varied online to VTCS so that VTCS can send control commands to both VTSSs. See [“How Clustered VTSS Configurations Work” on page 23](#) for more information.

Note – Also note the following:

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

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

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

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

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

...the choice is yours! This is a Way Cool variation on the theme we showed with Bi-Directional Clustering. We're going to use VTSSLST and VTSSSEL statements, however, to make a Bi-Directional Cluster Uni-Directional. Why would I want to do this? What if I wanted to switch the roles of the Primary and Secondary VTSSs? Easy, you just start with the same setup as described in the procedure beginning on [“Configuring and Managing a Bi-Directional Clustered System” on page 39](#). After you complete [Step 6](#), you throw in a subtle change with the following VTSSLST and VTSSSEL statements.

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

FIGURE 2-21 VTSSLST/VTSSSEL Statements - VSMPR1 Primary, VSMPR2 Secondary

In [FIGURE 2-21](#):

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

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

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

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

FIGURE 2-22 VTSSLST/VTSSSEL Statements - VSMPR2 Primary, VSMPR1 Secondary

In [FIGURE 2-22](#):

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

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

Now *that's* flexibility!

Another Variation: Synchronous or Asynchronous Replication?

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

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

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

? Implementing Synchronous Replication

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

1. **Ensure that your system has the Synchronous Replication requirements described in [TABLE 2-2 on page 22](#).**

2. **With all HSC/VTCS systems down, use CONFIG GLOBAL to enable Synchronous Replication:**

```
CONFIG GLOBAL SYNCHREP=YES
```

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

ALWAYS

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

CHANGED

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

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

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

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


VTSS Native IP Support

The VTSS native IP connection feature lets you use TCP/IP protocol to connect two VTSSs for VTV replication. Each VTSS has IFF3 cards with Ethernet ports for connection to the TCP/IP network. Previously, you were limited to ESCON or FICON connections for replication. Using TCP/IP for CLINKs can provide improved replication performance over ESCON or FICON protocols and, if so desired, allows the existing ESCON or FICON ports to be used exclusively for RTD and host connections.

Native IP applies to only VSM5s. Configuring native IP consists of the following:

- ? [“Satisfying Prerequisites” on page 48](#)
- ? [“The TCP/IP Environment” on page 50](#)
- ? [“Replacing IFF2 Cards with IFF3 Cards” on page 51](#)
- ? [“Configuring the IFF3 IP Addresses” on page 56](#)
- ? [“Configuring VTCS” on page 65](#)

Also see [“Changing the IP Address of an IFF3 Card” on page 71](#) for this procedure.

Satisfying Prerequisites

The following are prerequisites and recommendations for Native IP:

- ? VTSS microcode level H02.08 (for Models VSM5-45TB-IFF3, VSM5-68TB-IFF3, and VSM5-90TB-IFF3) or D02.08 (for all other models) is required.
- ? Network Infrastructure Requirements:
 - ? Gigabit Ethernet protocol is required on all network switches and routers that are directly attached to the IFF3 cards. The IFF3 card will only do speed negotiation to the 1 Gb speed.
 - ? StorageTek recommends a private Ethernet network for native IP connections.
 - ? Switches and Routers should support Jumbo(mtu=9000) packets for best performance.
 - ? Check that you are using the proper Ethernet cables:
 - ? CAT5 cables and below are **not** acceptable for GigE transmission from a VTSS.
 - ? CAT5E cable 90 meters is acceptable if run through a patch panel, 100 meters if straight cable.
 - ? CAT6 cable 100 meters is acceptable regardless of patch panel configuration.
- ? Each VTSS must have installed 4 IFF3 cards, each of which supports a 1gigabit RJ-45 ethernet connector. Each IFF3 can be configured to support up to 4 IP CLINK connections up to a maximum of 16 IP CLINKs per VTSS. Configurations can also include a mixture of ESCON/FICON and Native IP CLINKs.
- ? CDSLEVEL F and above is required, with the following PTFs:
 - ? For 6.1:
 - ? L1H14II - SMS6100
 - ? L1H14IJ - SOS6100
 - ? L1H14IK - SWS6100
 - ? For 6.2:
 - ? L1A00P7 - SMC6200
 - ? L1H14IM - SMS6200
 - ? L1H14O2 - SOS6200
 - ? L1H14IL - SWS6200
 - ? For 7.0, L1H150G (SES7000)
 - ? For 7.1, support is included in the base.

Note –

- ? The term “Jumbo” indicates supporting frame sizes greater than 1500 bytes Maximum Transmission Unit (MTU). MTU refers to the number of bytes of the largest protocol data unit (PDU) that a communications protocol layer can pass onwards to the other layers. A higher MTU means better efficiency since each packet carries more user data while protocol overheads (headers, and so forth) remain the same for each packet. So if the link supports larger packets with Jumbo MTU that means higher throughput. To achieve the optimal performance, the transfer of data requires frame sizes greater than 1500. Initially for native IP, it will be using an MTU size of 4880 bytes. To further increase performance the MTU size must increase to 9000 bytes.

- 7 The Path Maximum Transmission Unit Discovery (PMTUD) feature is required to let the VTSS IP know if a component in the network infrastructure can or cannot support the greater than 1500 frame size. For a router that does **not** support Jumbo/PMTUD, VTSS would send the router a Jumbo frame and the router would **not** let VTSS know that it cannot process that size and the frame would be discarded by the router. If the router supports PMTUD, it returns a message indicating that it cannot process that particular large frame size and VTSS resizes the Jumbo frame into the 1500 bytes (standard) size. The result is that the transmission is successful but at a lower performance. So for VTSS Native IP, the switch/router in the configuration must support JUMBO frames (Ethernet sends frames and IP sends packets) and PMTUD.
-

The TCP/IP Environment

TCP/IP attached CLINKs perform the same function as FICON or ESCON channel attached CLINKs, TIP/IP CLINKs originate from an Ethernet port on the IFF3 interface card instead of the ESCON or FICON ports on an ICE or VCF channel interface card. The TCP/IP connection is a standard copper Ethernet connector and must be attached either directly to another IFF3 card or a 1 Gb port on an Ethernet switch or router (connections at less than 1 Gb cannot perform replication). [FIGURE 3-1](#) shows Peer VTSSs, each with 3 IFF3 cards. The Ethernet cables from the IFF3 cards attach to Local Area Networks (LANs, one for each VTSS) and the LANs are connected via a Wide Area Network (WAN).

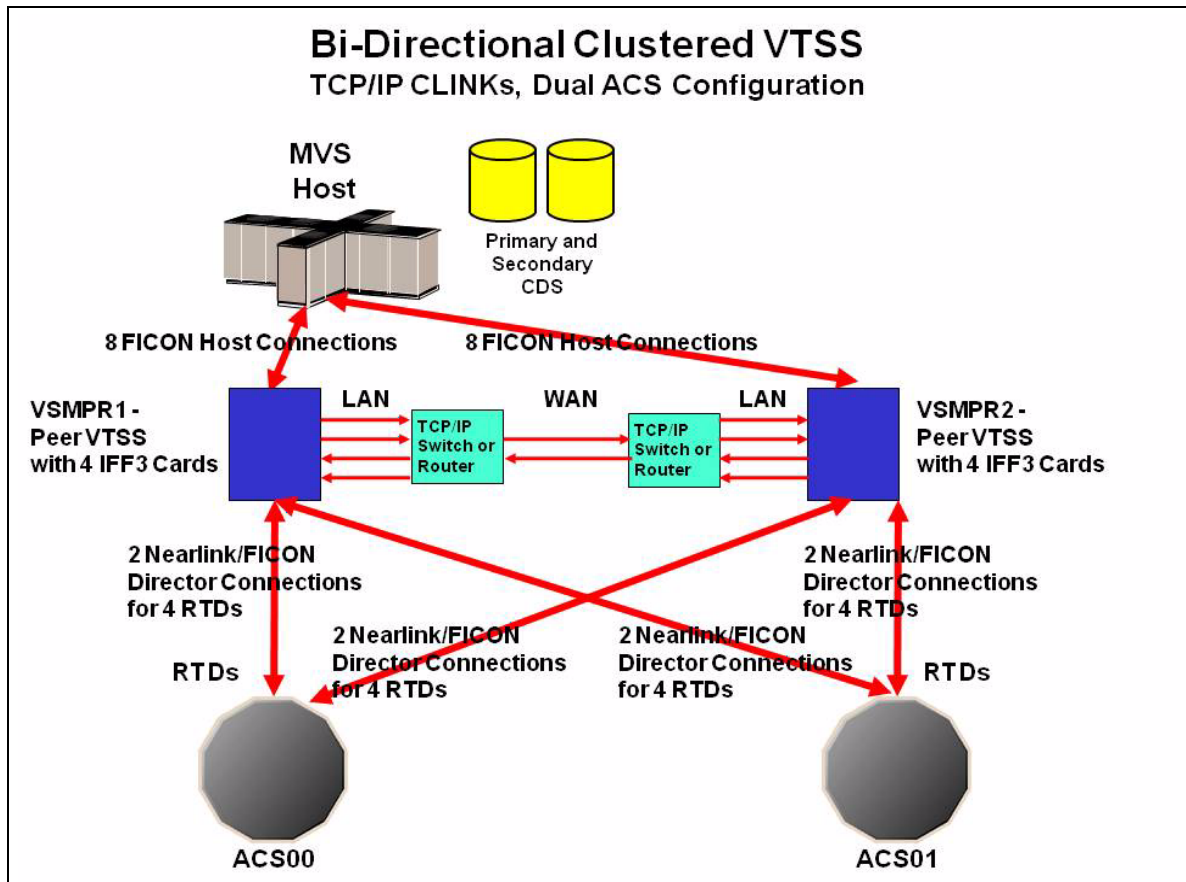


FIGURE 3-1 The TCP/IP Environment with IFF3 Cards for TCP/IP CLINKs

Replacing IFF2 Cards with IFF3 Cards

Note – IF the VSM5 came from factory with IFF3 cards installed skip to step 8

To replace IFF2 Cards with IFF3 cards:

1. Power off the VTSS, disconnect the Fibre Channel cables, and remove the IFF2 cards.
2. Install the new IFF3 cards.

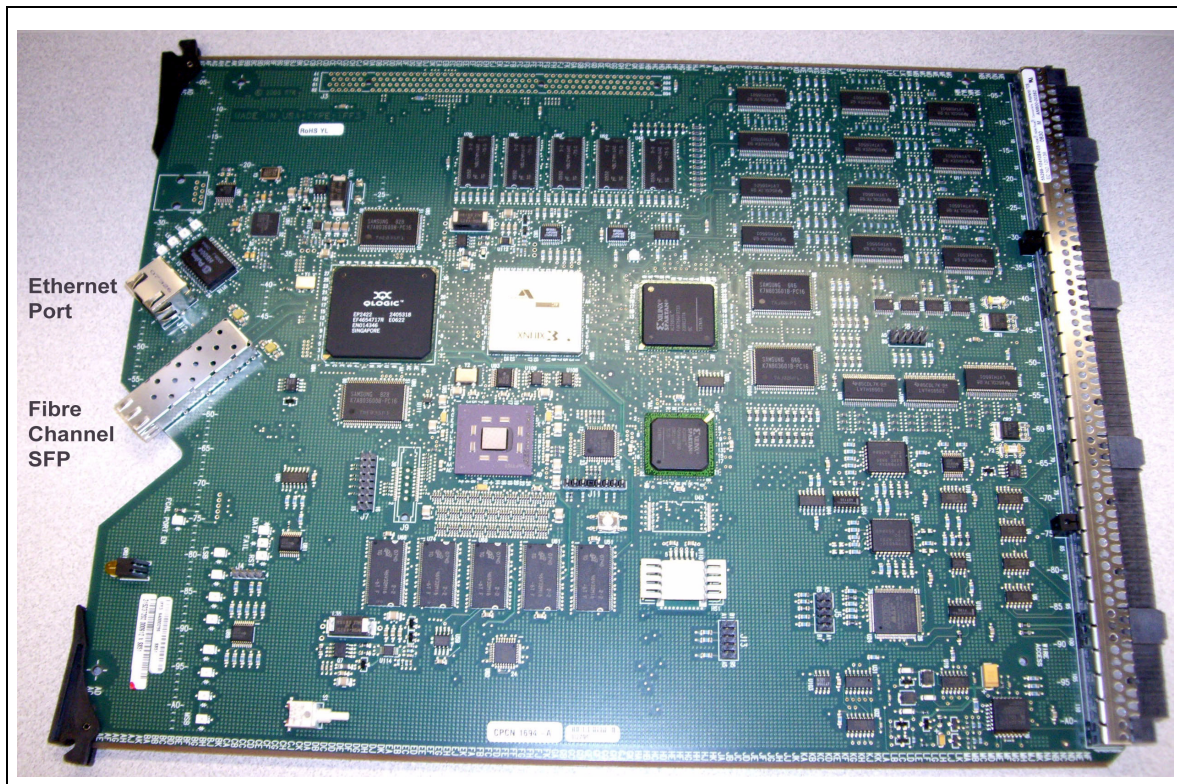


FIGURE 3-2 IFF3 Card

As shown in [FIGURE 3-2](#), each IFF3 card has one Ethernet port and one Fibre Channel SFP. **Also note the LED Status Indicators**, which are the small white squares above the Ethernet port. The **LEDs** have the following labels in white letters:

- ? **INT** - Interrupt to PPC440GX (Int 3), which is the LED furthest from the card edge. It comes on when the interrupt asserts, however in normal operation you cannot see it come on, because it is not on long enough.
- ? **LS2** - Link State 2.
- ? **LS1** - Link State.
- ? **ATN** - Activity LED, which is the LED closest to the card edge. It blinks when frames are coming or going.

TABLE 3-1 shows the meaning of the combination of the LS1 and LS2 LEDs.

TABLE 3-1 LS1 and LS2 LED Combinations

LS1 State	LS2 State	Meaning
OFF	OFF	No Link
OFF	ON	100Base-Tx
ON	OFF	10Base-T
ON	ON	1000Base-T

3. Connect the Ethernet cables and Fibre Channel Cables as shown in FIGURE 3-3.

Note that in FIGURE 3-3, the white cables from the wiring harness connect to the Ethernet port on each card, while the black cables connect to the Fibre Channel SFP on each card.

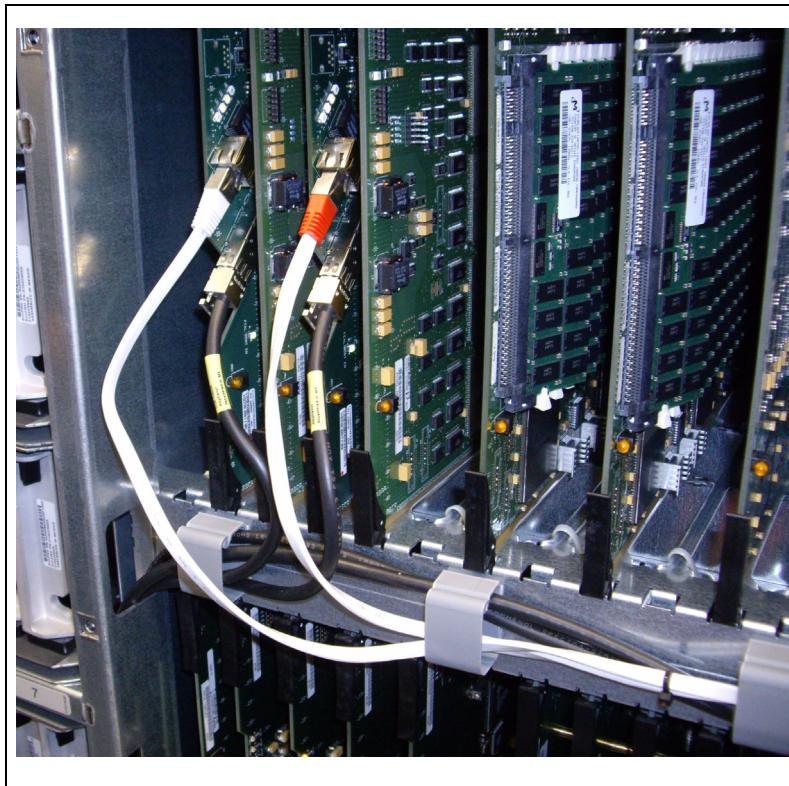


FIGURE 3-3 IFF3 Card Connections

4. Ensure that the VTSS is connected with Ethernet cable to the TCP/IP network via the IFF3 port at the bottom of the frame as shown in [FIGURE 3-4](#).

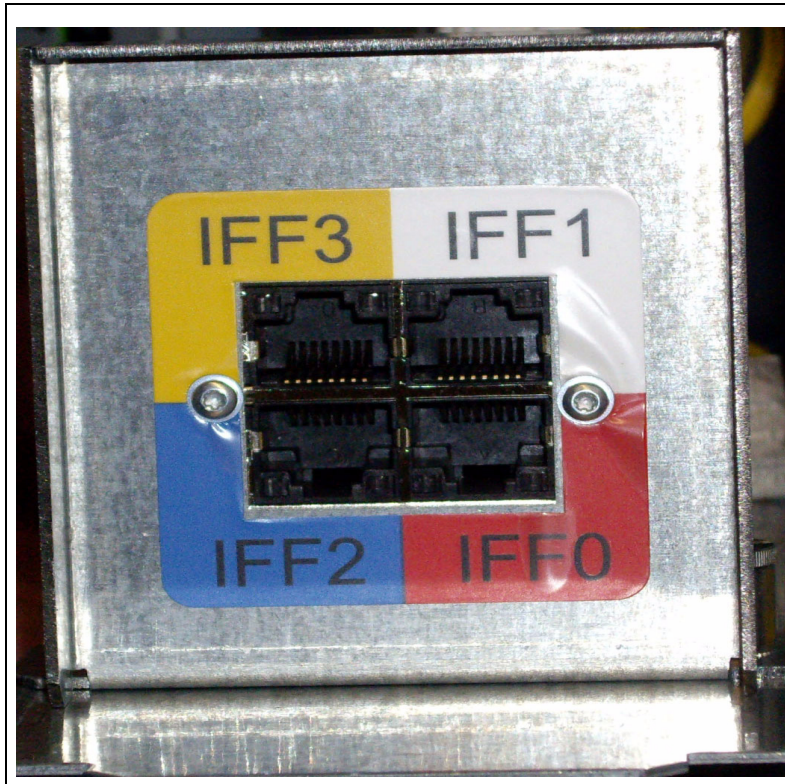


FIGURE 3-4 IFF3 Ethernet Port for Connection to TCP/IP Network

5. Power on the VTSS and IML VIP.
6. Upgrade the VTSS microcode to D02.08.00.00.
7. Set the '6 MAC feature' in the ACMB Frudata:


The '6 MAC Feature' must be enabled before proceeding. This can be checked or enabled using [Step 7](#).

- a. Select 'Subsystem Debug, Fru ID, ACMB.
If MOD FLAGS=0x04, skip to [Step 8](#). Otherwise, continue with [Step b](#).
- b. Set MOD FLAGS to 0x04.
- c. Set new MAC address provided with Conversion Bill or from NPDC.

d. Press the 'Continue' button and verify that the new data is recorded.

http://zenobia - StorageTek: VSM Op-Panel - Web Browser


STORAGETEK™ VSM - Virtual Storage Manager


VIP Version 4.2.6.1	SIML Version 02000007	IP dvtx3	S/N 00200124	Master ISP 0	
-------------------------------	---------------------------------	--------------------	------------------------	------------------------	---


ACMB FRU Data


Machine Family ID:	0567
Card Type:	ACMB
Part Number:	000311881303
Serial Number:	013366
EC Level:	133609
Mod Flags:	0x 04
Software Compat Level:	0x 00
Hardware Compat Level:	0x 01
Features:	0x 00
Primary Error Code:	0x 0000
Site Code:	77
MAC Address:	0x 000000104F08A9BE


Exit
Exit



Configuration & Status



Maintenance & Debug


Diagnostics


State Save


Software Release Level


Directory Display


IML

Done

8. IML the D02.08.00.00 level VTSS Microcode.

9. Ensure that the ClusterVTSS StorageKey is installed.

For more information, see *VSM5 Installation and Service Guide*, Chapter 1.

10. Install the IPCluster StorageKey.

For more information, see *VSM5 Installation and Service Guide*, Chapter 1.

11. Obtain from the customer/network administrator:

- ? IP addresses for Primary and Secondary VTSS - IFF0-3.
- ? Gateway addresses (if required) for Primary and Secondary VTSS - IFF0-3.
- ? Subnet Mask for Primary and Secondary VTSS - IFF0-3.
- ? IFF card Ethernet Port Target Addresses for Primary and Secondary VTSS - IFF0-3.

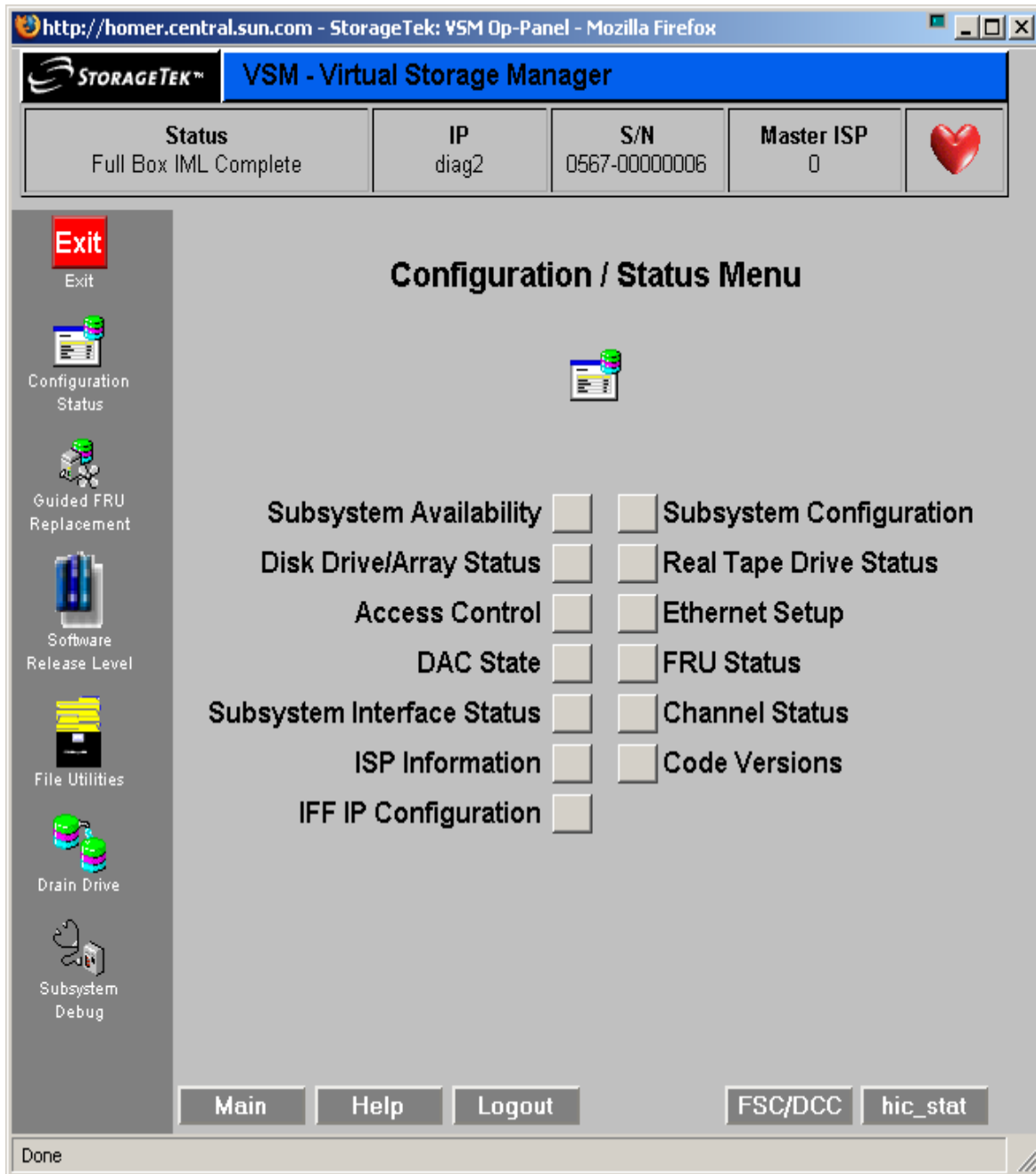
Note – For optimum throughput and redundancy, StorageTek recommends the following for Target Addresses:

- ? Allocate IP addresses to all Targets on all IFF cards. Even if you are not currently using all IFF cards for Native IP replication, this ensures that you have these addresses available for future expansion.
 - ? If each card is on a different physical network, allocate different IP addresses to the same target on different IFF cards. If each IFF card is on the same physical network, you can use the same IP address for the same target on different IFF cards.
-

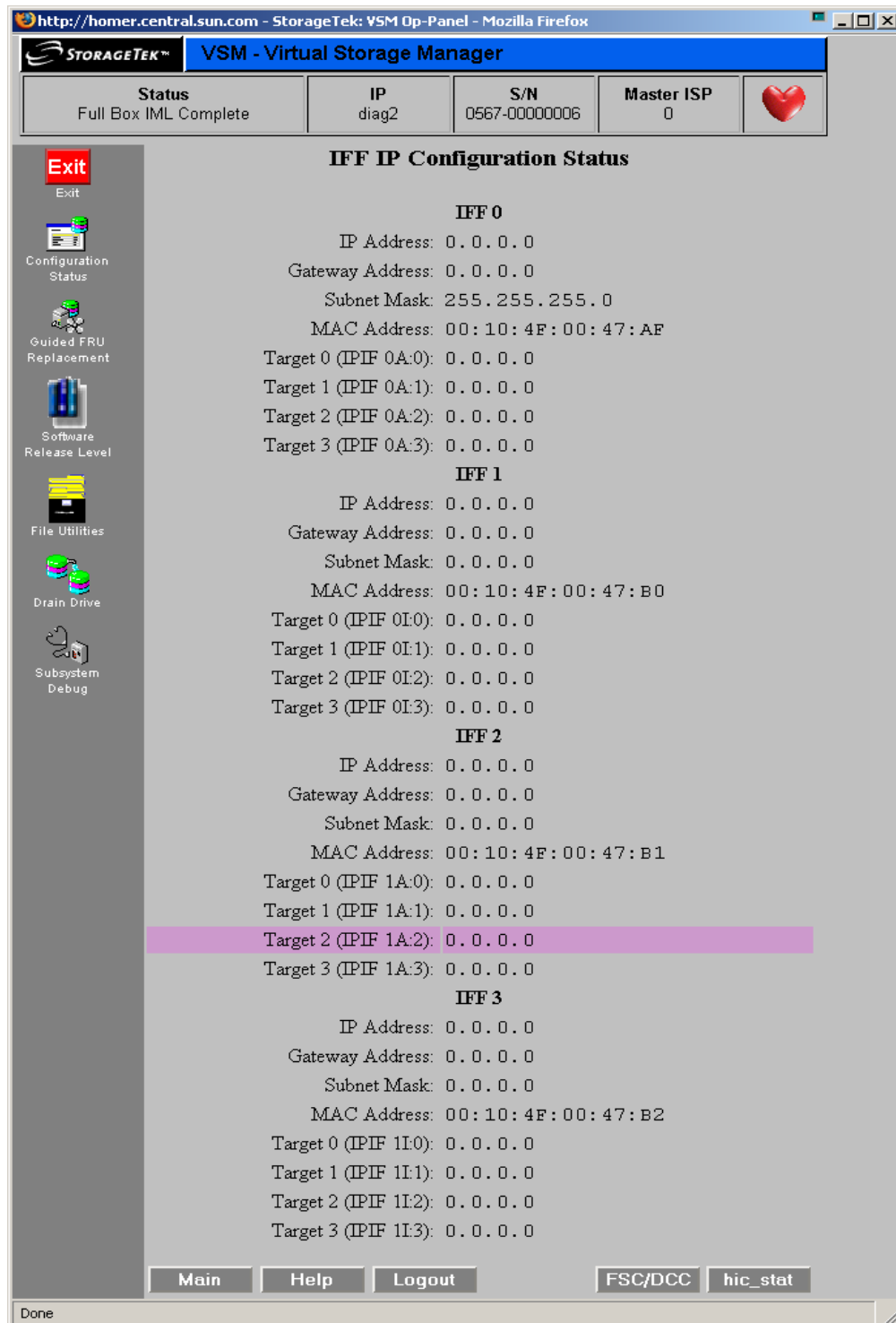
Configuring the IFF3 IP Addresses

To configure the VTSSs:

1. From DOP 'Configuration/Status' screen, select 'IFF IP Configuration'.



- From IFF IP Configuration screen you will configure each IFF card: IP address, Gateway Address, Subnet Mask, and Secondary Target IPs. You can also view the MAC Address assigned to each IFF ethernet port.



3. Hover the cursor over the IFF0 IP Address field to modify (field will highlight) and left click mouse.

This will bring up the data entry screen for IP Address. Fill in the IP address using the customer supplied IP then hit the Continue button. You will get a screen showing Success or Failure. Hit Continue button to take you back to the IFF IP Configuration screen for next entry.

http://homer.central.sun.com - StorageTek: VSM Op-Panel - Mozilla Firefox

STORAGETEK™ VSM - Virtual Storage Manager

Status Full Box IML Complete	IP diag2	S/N 0567-00000006	Master ISP 0	
--	--------------------	-----------------------------	------------------------	--

Exit
Exit

Configuration Status

Guided FRU Replacement

Software Release Level

File Utilities

Drain Drive

Subsystem Debug

Configure IFF0 IP Address Address

0 . 0 . 0 . 0

Continue

Cancel

Main Help Logout FSC/DCC hic_stat

Done



OR



4. **Hover the cursor over the Gateway Address field to modify (field will highlight) and left click mouse.**

This will bring up the data entry screen for Gateway Address. Fill in the Gateway address using the customer supplied IP then hit the Continue button. You will get a screen showing Success or Failure. Hit Continue button to take you back to the IFF IP Configuration screen for next entry.

http://homer.central.sun.com - StorageTek: VSM Op-Panel - Mozilla Firefox

STORAGETEK™ VSM - Virtual Storage Manager

Status Full Box IML Complete	IP diag2	S/N 0567-00000006	Master ISP 0	
--	--------------------	-----------------------------	------------------------	---

Exit
Exit


Configuration Status


Guided FRU Replacement


Software Release Level


File Utilities


Drain Drive


Subsystem Debug

Configure IFF0 Gateway Address Address

0

.

0

.

0

.

0

Continue

Cancel

Main

Help

Logout

FSC/DCC

hic_stat


Done

5. Hover the cursor over the Subnet Address field to modify (field will highlight) and left click mouse.

This will bring up the data entry screen for Subnet Address. Fill in the Subnet address using the customer supplied IP then hit the Continue button. You will get a screen showing Success or Failure. Hit Continue button to take you back to the IFF IP Configuration screen for next entry.

http://homer.central.sun.com - StorageTek: VSM Op-Panel - Mozilla Firefox

STORAGETEK™ VSM - Virtual Storage Manager

Status Full Box IML Complete	IP diag2	S/N 0567-00000006	Master ISP 0	
--	--------------------	-----------------------------	------------------------	---

Exit
Exit


Configuration Status


Guided FRU Replacement


Software Release Level


File Utilities


Drain Drive


Subsystem Debug

Configure IFF0 Subnet Mask Address

. . .

Done

6. Hover the cursor over the Subnet Address field to modify (field will highlight) and left click mouse.

This will bring up the data entry screen for Subnet Address. Fill in the Subnet address using the customer supplied IP then hit the Continue button. You will get a screen showing Success or Failure. Hit Continue button to take you back to the IFF IP Configuration screen for next entry.

7. Hover the cursor over the Target Address field to modify (field will highlight) and left click mouse.

This will bring up the data entry screen for Target Address. Fill in the Target address using the customer supplied IP then hit the `Continue` button. You will get a screen showing Success or Failure.

8. Hover the cursor over the Target Address field to modify (field will highlight) and left click mouse.

This will bring up the data entry screen for Target Address. Fill in the Target address using the customer supplied IP then hit the `Continue` button. You will get a screen showing Success or Failure.

If the Source and Target are fully cabled you can use the “Validate” button to send a test to the Target. If the validate is successful you will get a screen display showing the Target's Family ID and Serial Number. If the Validate is not successful you will get a message indicating to check `Hic_Stat` for failing FSC.

Hit `Continue` button to take you back to the IFF IP Configuration screen for next entry.
Repeat for Target 1, 2, and 3.



9. Repeat [Step 3](#) through [Step 8](#) for IFF1, IFF2, IFF3.

10. Repeat [Step 1](#) through [Step 9](#) for Secondary VTSS unit.

Configuring VTCS

CONFIG Utility Changes

The CONFIG CLINK statement now provides for two types of VTSS-to-VTSS connections via the following parameters:

CLINK CHANIF=*nn* or *nn:n*

existing parameter; allows connection of two FICON (or ESCON) ports.

CLINK IPIF=*nn:n*

new parameter; allows Ethernet connection of two Native IP ports, where the *nn:n* values are shown in parentheses *before* each Target IP Address on the IFF IP Configuration Status screen for each IFF ethernet port (see [Step 2](#) on [page 57](#)).

For example, IPIF=0A:0 corresponds to the IP address for Target 0 on IFF0.

Note – CLINK statement must contain either the CHANIF or the IPIF parameter, but not both.

Example: Clustered VTSS with TCP/IP CLINKs

FIGURE 3-5 shows an example of a Clustered VTSS configuration with TCP/IP CLINKs.

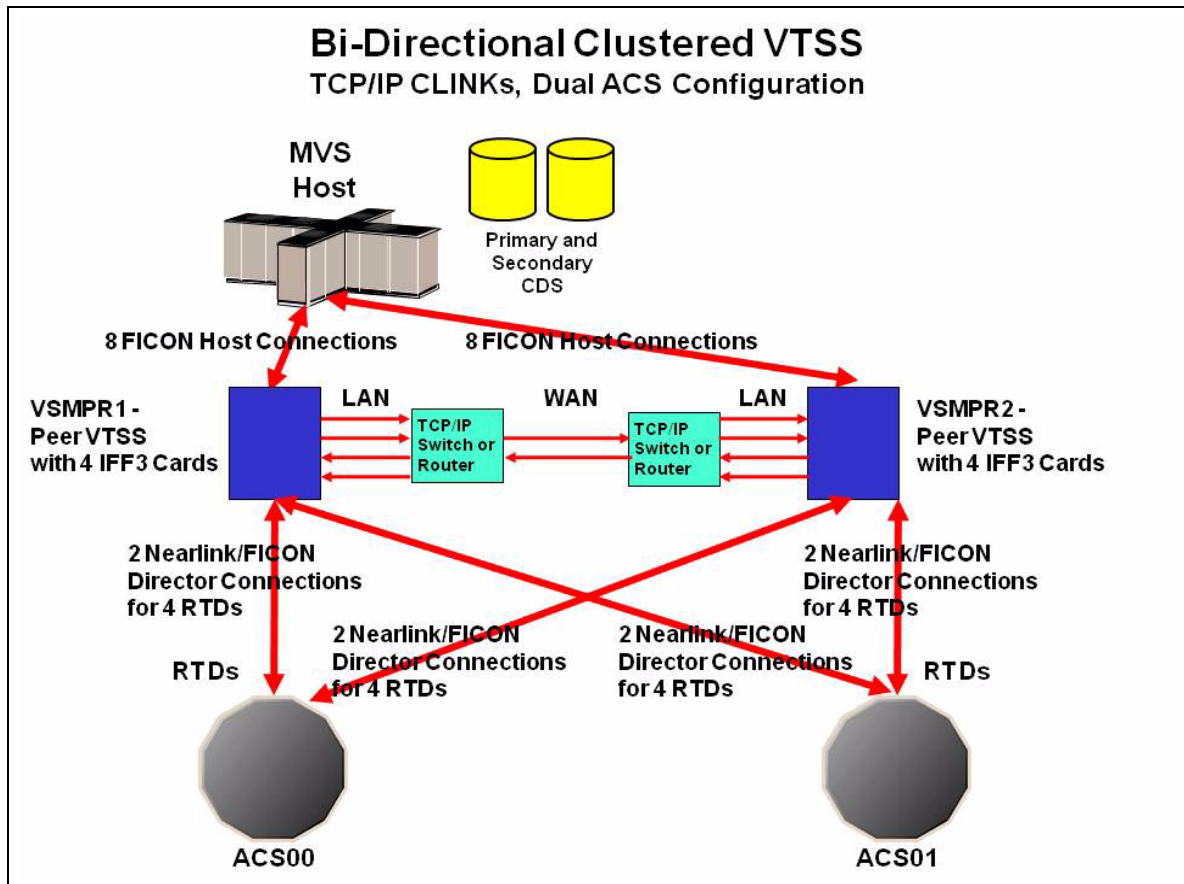


FIGURE 3-5 Clustered VTSS with TCP/IP IP CLINKs

In [FIGURE 3-5 on page 66](#), assume that initially you will *only* use IFF0 on each VTSS for Native IP replication. In this situation, you would allocate all IP addresses for all Targets on all IFF cards, but would only code IPIF parameters for the four Targets on IFF0, as shown in [TABLE 3-2](#) and [TABLE 3-3](#).

TABLE 3-2 CLINK IPIF Values for IFF0, VSMRP1

IFF0 Target Number	Example IP	Corresponding CLINK IPIF
Target 0	128.0.1.1	0A:0
Target 1	128.0.2.1	0A:1
Target 2	128.0.3.1	0A:2
Target 3	128.0.4.1	0A:3

TABLE 3-3 CLINK IPIF Values for IFF0, VSMRP2

IFF0 Target Number	Example IP	Corresponding CLINK IPIF
Target 0	128.0.1.2	0A:0
Target 1	128.0.2.2	0A:1
Target 2	128.0.3.2	0A:2
Target 3	128.0.4.2	0A:3

[FIGURE 3-6](#) shows example CONFIG JCL to define the configuration shown in [FIGURE 3-5 on page 66](#) with the values shown in [TABLE 3-2](#) and [TABLE 3-3](#).

```

//CREATECF EXEC PGM=SLUADMIN,PARM='MIXED'
//STEPLIB DD DSN=h1q.SEALINK,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 CDSLEVEL(V61ABOVE)
GLOBAL MAXVTV=32000 MVCFREE=40 VTVATTR=SCRATCH RECALWER=YES
LOCKSTR=VTCS_LOCKS REPLICAT=ALWAYS VTVPAGE=LARGE INITMVC=YES
SYNCHREP=YES MAXRTDS=16 FASTMIGR=YES
RECLAIM THRESHLD=70 MAXMVC=40 START=35
VTSS NAME=VSMR1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=VPR12A00 DEVNO=2A00 CHANIF=0C:0
RTD NAME=VPR12A01 DEVNO=2A01 CHANIF=0C:1
RTD NAME=VPR12A02 DEVNO=2A02 CHANIF=0C:2
RTD NAME=VPR12A03 DEVNO=2A03 CHANIF=0C:3
RTD NAME=VPR12A04 DEVNO=2A04 CHANIF=0G:0
RTD NAME=VPR12A05 DEVNO=2A05 CHANIF=0G:1
RTD NAME=VPR12A06 DEVNO=2A06 CHANIF=0G:2
RTD NAME=VPR12A07 DEVNO=2A07 CHANIF=0G:3
VTD LOW=9900 HIGH=99FF
VTSS NAME=VSMR2 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=VPR22B00 DEVNO=2B00 CHANIF=0C:0
RTD NAME=VPR22B01 DEVNO=2B01 CHANIF=0C:1
RTD NAME=VPR22B02 DEVNO=2B02 CHANIF=0C:2
RTD NAME=VPR22B03 DEVNO=2B03 CHANIF=0C:3
RTD NAME=VPR22B04 DEVNO=2B04 CHANIF=0G:0
RTD NAME=VPR22B05 DEVNO=2B05 CHANIF=0G:1
RTD NAME=VPR22B06 DEVNO=2B06 CHANIF=0G:2
RTD NAME=VPR22B07 DEVNO=2B07 CHANIF=0G:3
VTD LOW=9900 HIGH=99FF
CLUSTER NAME=CLUSTER1 VTSSs(VSMR1,VSMR2)
CLINK VTSS=VSMR1 IPIF=0A:0
CLINK VTSS=VSMR1 IPIF=0A:1
CLINK VTSS=VSMR1 IPIF=0A:2
CLINK VTSS=VSMR1 IPIF=0A:3
CLINK VTSS=VSMR2 IPIF=0A:0
CLINK VTSS=VSMR2 IPIF=0A:1
CLINK VTSS=VSMR2 IPIF=0A:2
CLINK VTSS=VSMR2 IPIF=0A:3

```

FIGURE 3-6 CONFIG example: Clustered VTSS with TCP/IP IP CLINKs

Other VTCS Changes

DECOM

The DECOM utility now outputs the IPIF parameter of the CLINK statement if the CLINK is on a Native IP interface.

DISPLAY CLINK

The DISPLAY CLINK command now displays the CLINK interface address with a prefix to show whether the CLINK is on an IP or FICON interface. For example:

```
SLS6603I Clink information: 141

VTSS      Id  IF      Status      Usage      Host

VTSS0001  00  COM:0   Online     Free
           01  C1M:0   Online     Free
VTSS0002  00  COM:1   Online     Free
           01  COM:1   Online     Free
           02  IOA:3   Online     Free
           03  I1I:2   Online     Free

SLS5013I Command completed (0)
```

DISPLAY VTSS DETAIL

DISPLAY VTSS DETAIL displays if the VTSS supports IP replication over IP.

XML Processing

A new tag <ipif_id> display the interface address of a CLINK attached to an IP interface.

Messages

The following messages are updated.

SLS6751I

CLINK CLINKID ON VTSS XXXXXXXX iftype ifaddr RETURNED ECAM ERROR CC=CCC RC=RRRRRRRR

SLS6754I

CLINK CLINKID iftype ifaddr XXXXXXXX FAILED TO DISMOUNT VTV VVVVVV

SLS6755I

CONFIGURING CLINK CLINKID iftype Ifaddr VTSS XXXXXXXX

SLS6756E

CLINK CLINKID iftype ifaddr VTSS XXXXXXXX CONFIGURATION MISMATCH
ifid1:ifid2 ifaddr1:ifaddr2

SLS6757I

CLINK CLINKID iftype ifaddr VTSS XXXXXXXX FAILED INITIAL CONFIGURATION
WITH CC=CCC RC=RRRRRR

SLS6758I

CLINK CLINKID iftype ifaddr VTSS XXXXXXXX FAILED TO REPLICATE VTV
VVVVVV

SLS6759I

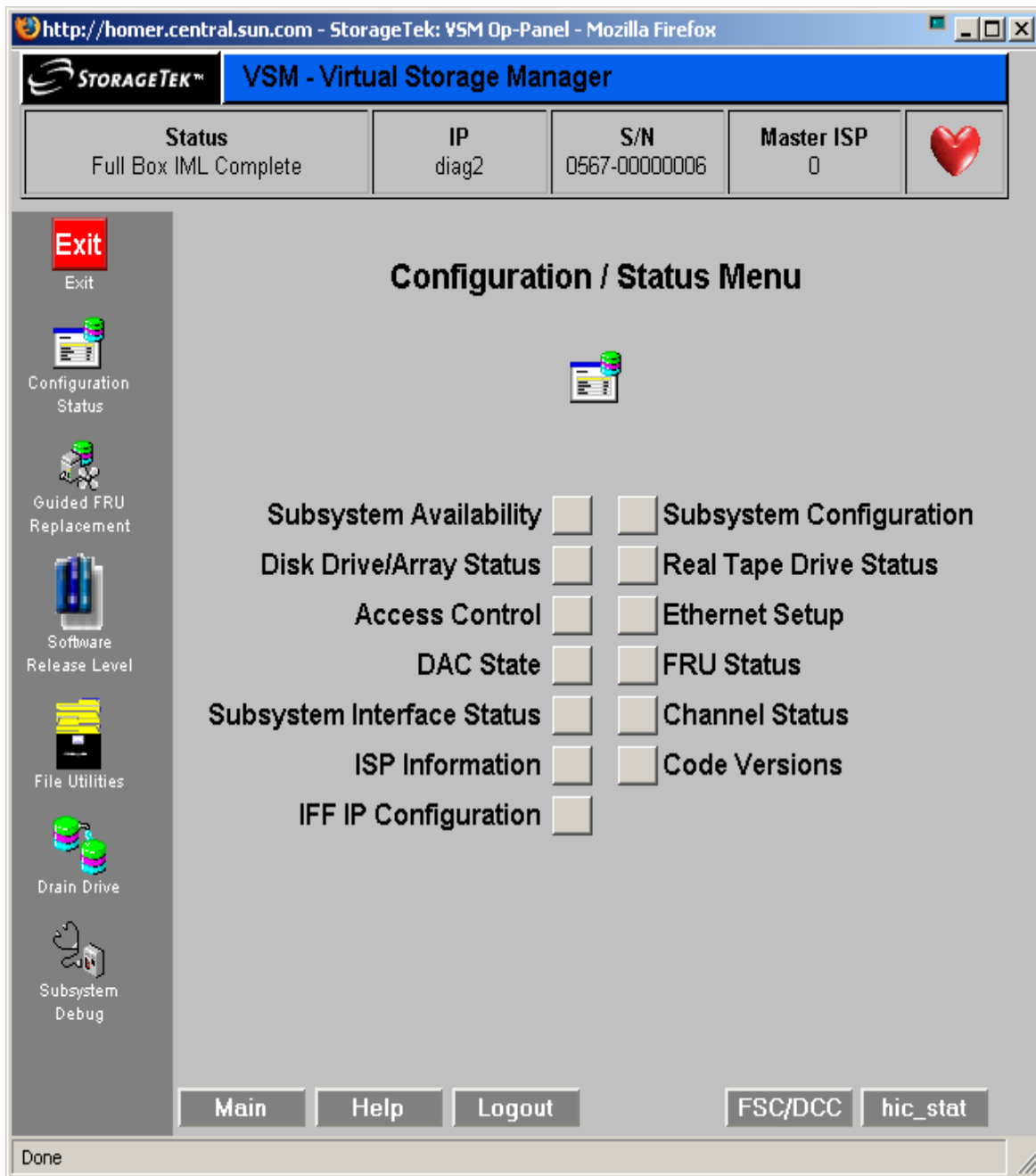
CLINK CLINKID iftype ifaddr VTSS XXXXXXXX NOW ONLINE:

Changing the IP Address of an IFF3 Card

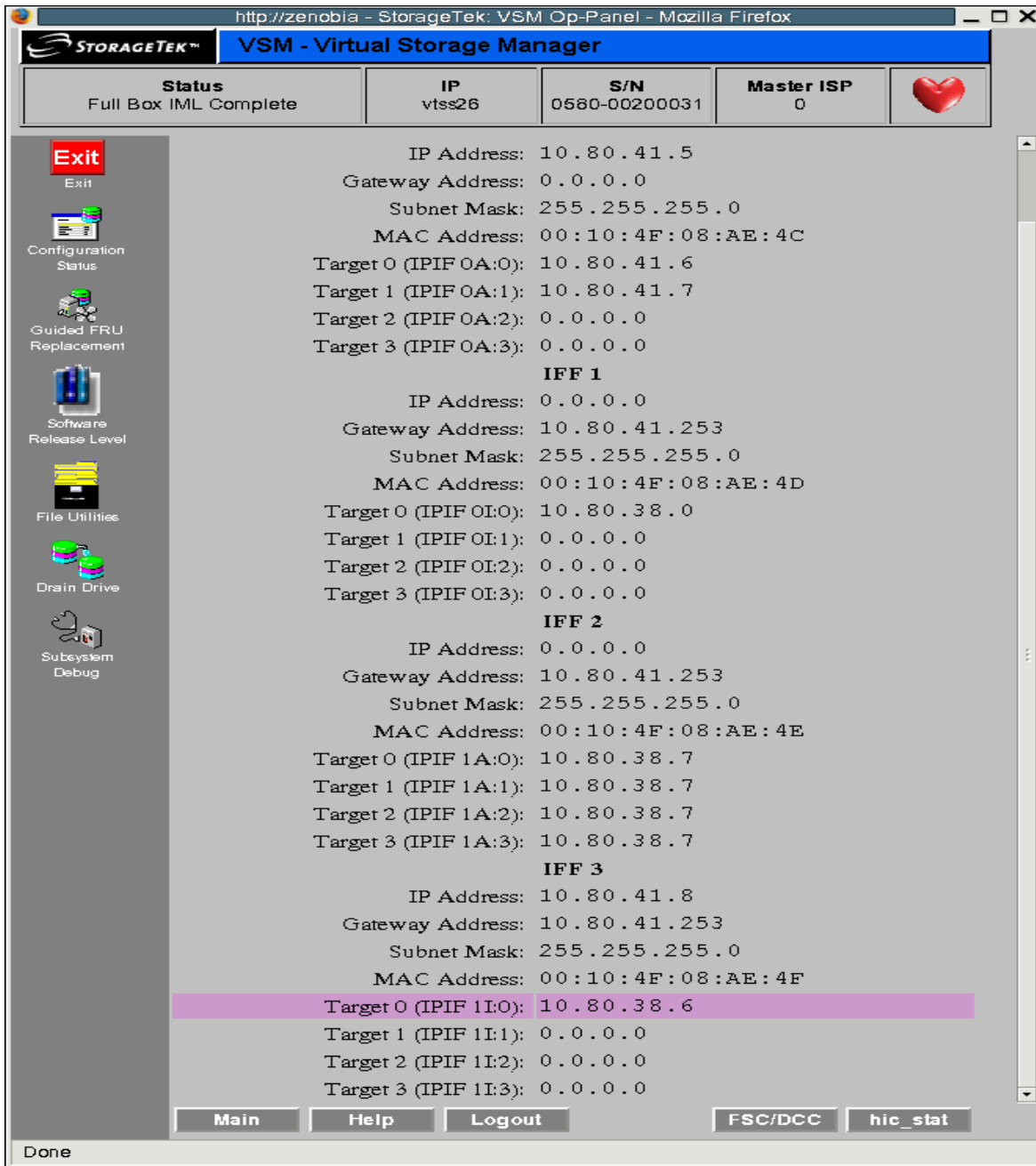
As an example, the following procedure changes the IP address of Target 0 on IFF 3 from 10.80.38.6 to 10.80.38.5. **Note that** this changes **only** the IP address; it is not necessary to update the `CONFIG CLINK` statement that corresponds to this IP address.

To change the IP address of an IFF3 card:

1. From DOP 'Configuration/Status' screen, select 'IFF IP Configuration'.



- The following shows the IFF IP Configuration screen with Target 0 on IFF 3 initially set to 10.80.38.6.



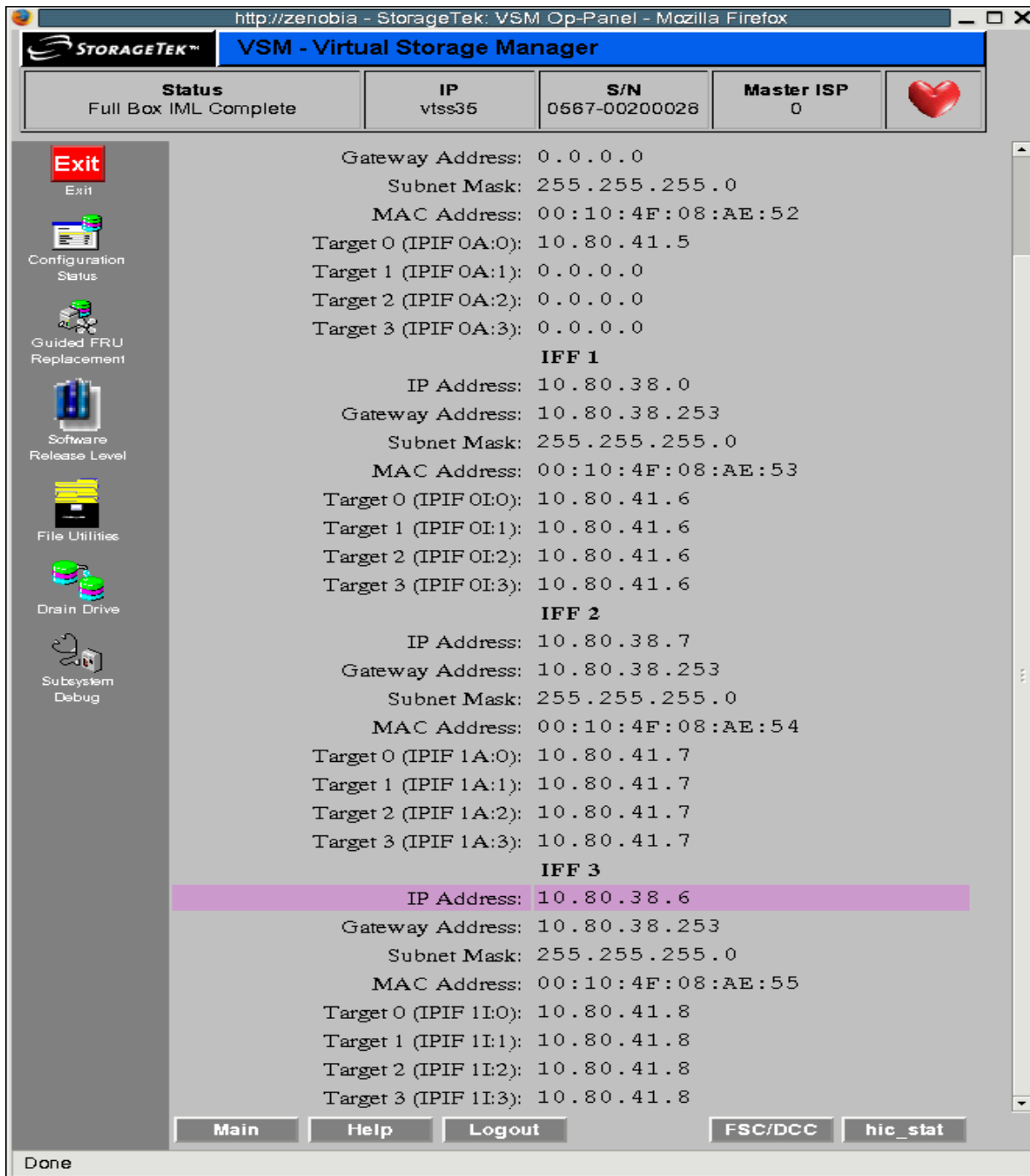
3. Change Target 0 on IFF 3 to 0.0.0.0 as shown below.

The screenshot displays the StorageTek VSM - Virtual Storage Manager web interface. The browser address bar shows 'http://zenobia - StorageTek: VSM Op-Panel - Mozilla Firefox'. The interface has a blue header with the StorageTek logo and the title 'VSM - Virtual Storage Manager'. Below the header is a status bar with the following information:

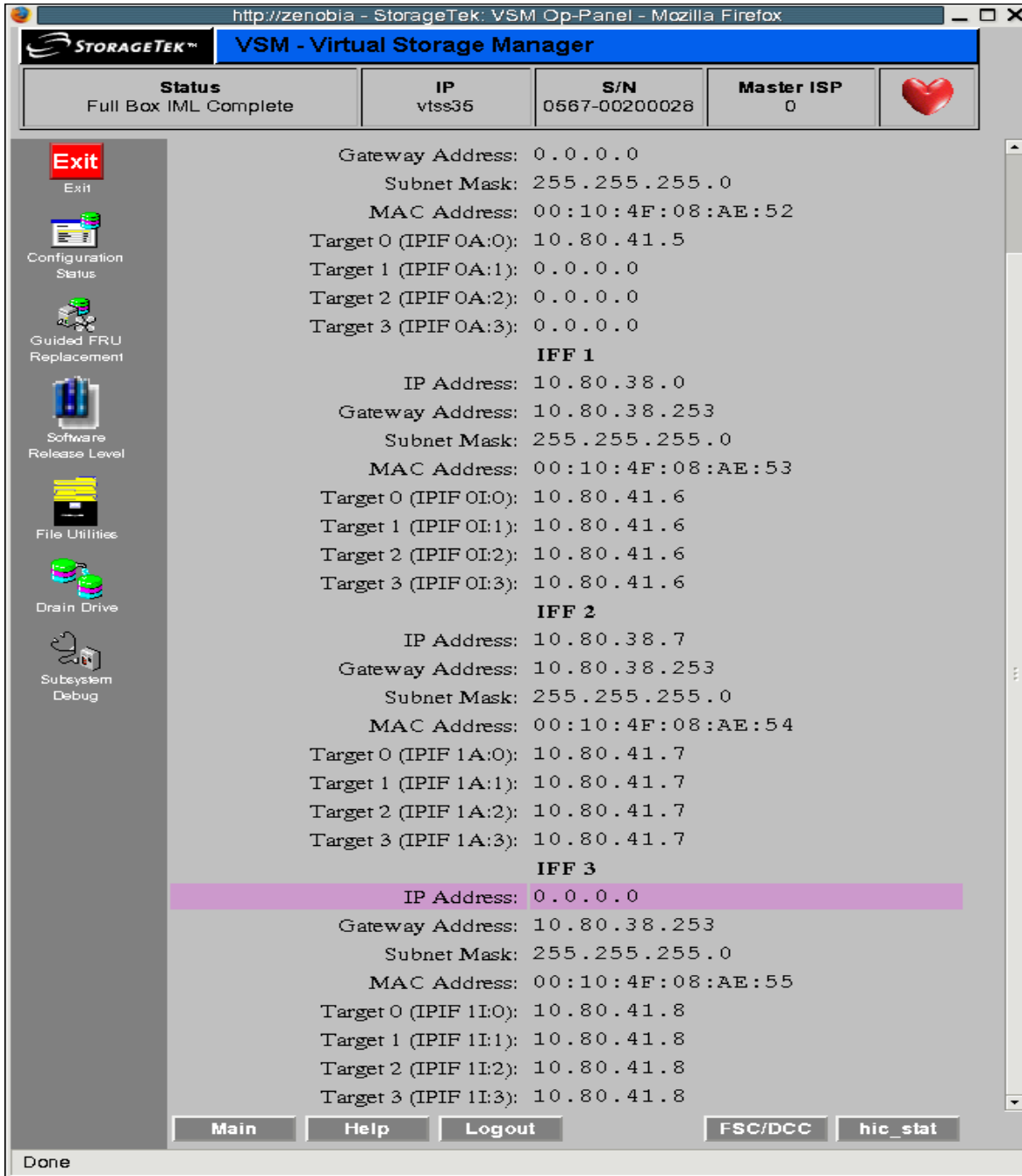
Status Full Box IML Complete	IP vtss26	S/N 0580-00200031	Master ISP 0	
--	---------------------	-----------------------------	------------------------	--

On the left side, there is a vertical menu with icons and labels for various functions: Exit, Configuration Status, Guided FRU Replacement, Software Release Level, File Utilities, Drain Drive, and Subsystem Debug. The main content area displays configuration details for three IFFs (IFF 1, IFF 2, and IFF 3). Each IFF section lists the IP Address, Gateway Address, Subnet Mask, MAC Address, and four targets (Target 0 to Target 3). The 'Target 0 (IPIF 1I:0)' field under IFF 3 is highlighted in pink and set to 0.0.0.0. At the bottom of the interface, there are buttons for 'Main', 'Help', 'Logout', 'FSC/DCC', and 'hic_stat'. The status bar at the very bottom shows 'Done'.

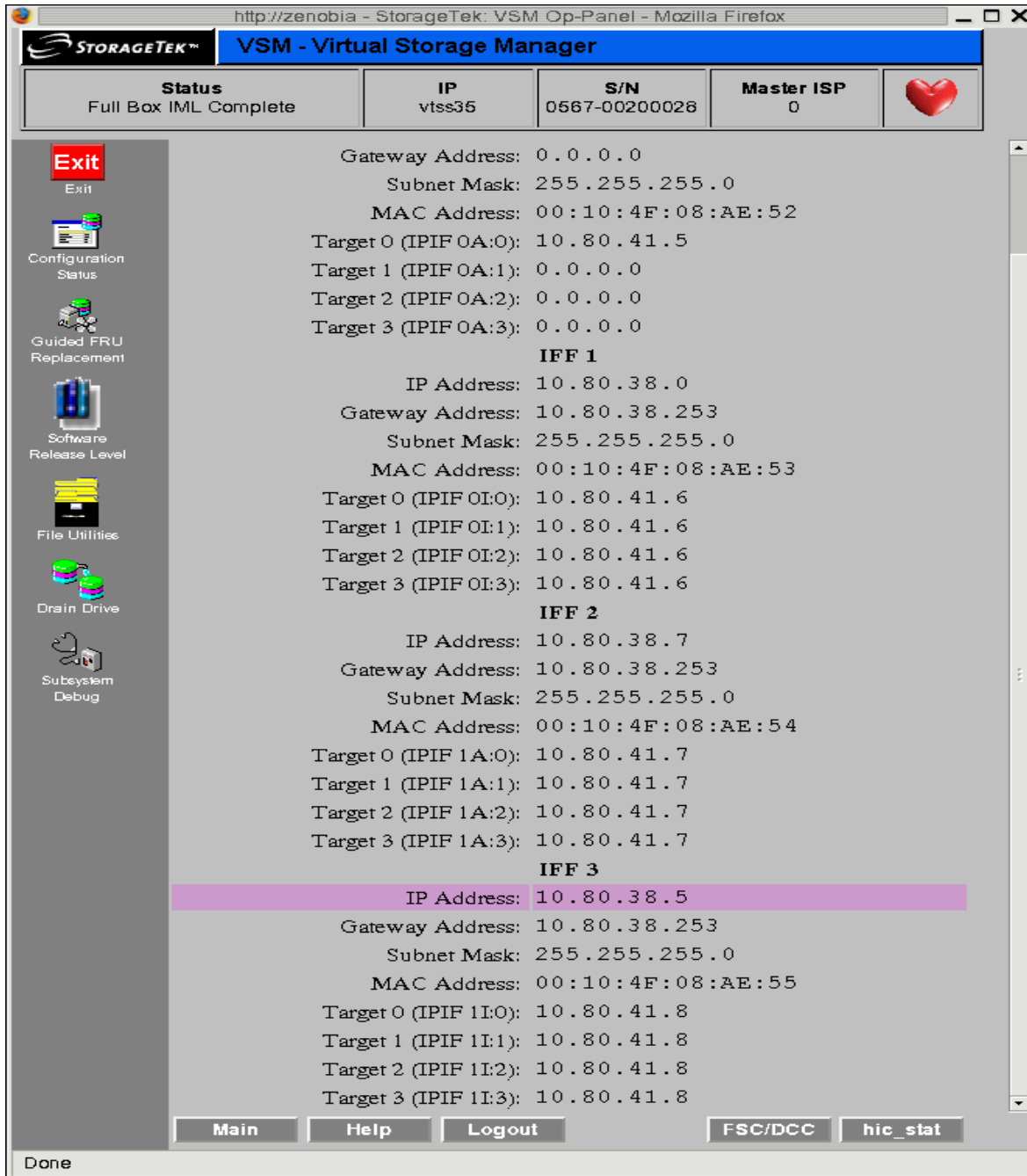
4. The IP Address of IFF 3 is initially set to 10.80.38.6 as shown below.



5. Change the IP Address of IFF 3 to 0.0.0.0 as shown below.



6. Change the IP address of the IFF3 card from 0.0.0.0 to 10.80.38.5. as shown below.



7. Change Target 0 on IFF 3 from 0.0.0.0 to 10.80.38.5 as shown below.

StorageTek™ VSM - Virtual Storage Manager

Status	IP	S/N	Master ISP
Full Box IML Complete	vtss26	0560-00200031	0

Exit
Exit

Configuration Status

Guided FRU Replacement

Software Release Level

File Utilities

Drain Drive

Subsystem Debug

IFF 1

IP Address: 0.0.0.0
Gateway Address: 10.80.41.253
Subnet Mask: 255.255.255.0
MAC Address: 00:10:4F:08:AE:4D
Target 0 (IPIF 0I:0): 10.80.38.0
Target 1 (IPIF 0I:1): 0.0.0.0
Target 2 (IPIF 0I:2): 0.0.0.0
Target 3 (IPIF 0I:3): 0.0.0.0

IFF 2

IP Address: 0.0.0.0
Gateway Address: 10.80.41.253
Subnet Mask: 255.255.255.0
MAC Address: 00:10:4F:08:AE:4E
Target 0 (IPIF 1A:0): 10.80.38.7
Target 1 (IPIF 1A:1): 10.80.38.7
Target 2 (IPIF 1A:2): 10.80.38.7
Target 3 (IPIF 1A:3): 10.80.38.7

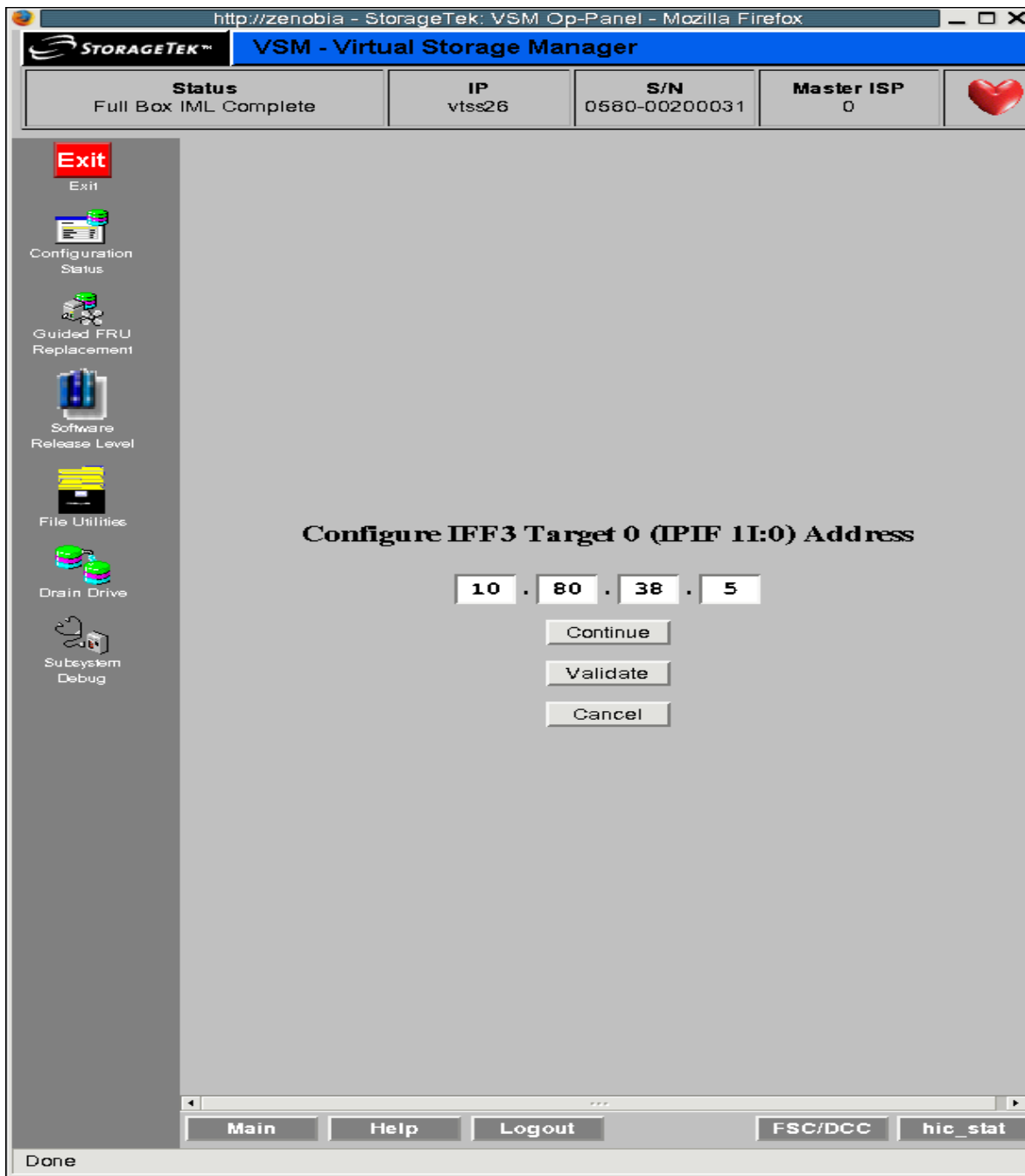
IFF 3

IP Address: 10.80.41.8
Gateway Address: 10.80.41.253
Subnet Mask: 255.255.255.0
MAC Address: 00:10:4F:08:AE:4F
Target 0 (IPIF 1I:0): 10.80.38.5
Target 1 (IPIF 1I:1): 0.0.0.0
Target 2 (IPIF 1I:2): 0.0.0.0
Target 3 (IPIF 1I:3): 0.0.0.0

Main **Help** **Logout** **FSC/DCC** **hic_stat**

Done

8. For the target just changed, press the “Validate” button.



9. You should get a confirmation message as shown below.



Doing Physical Exports and Imports

The EXPORT and IMPORT facilities give you the tools to create physically portable MVCs. At the source site, you use EXPORT to consolidate VTVs on MVCs (if necessary), and produce a Manifest File that describes the MVC contents (VTVs on the MVC). You then eject the MVCs from the source site, physically transport them to the target site, then IMPORT them, using the Manifest File to update the CDS with information on the imported MVCs and VTVs. **Note that** you can import VTVs into a CDS without VTCS being active. You then enter the MVCs into the target site.

Note –

- ? If you decide to return exported MVCs back to the source system, no special VTCS processing is required; just enter the MVCs into an LSM at the source system.
 - ? For each VTV imported, the only MVC copies that will be created will be for MVCs that have been exported and imported via the same statements. This has a particular significance when importing Duplexed VTVs. Such a VTV will only have copies on both MVCs after the import if both MVCs are present in the same Manifest file and imported as a result of the same IMPORT statement.
-

You export by either of the following general methods:

- ? **Export by VTV or Management class**, which consolidates the selected VTVs onto a new set of MVCs. As Consolidation takes time and requires VTSS resources the preferred option is to perform Export by MVC or Storage Class. For more information, see [“Exporting and Importing by Management Class” on page 82](#).
 - ? **Export by MVC or Storage Class**. An export by storage class or MVC does not require any Consolidation post-processing of VTVs, and there is no data movement required. The export simply creates a Manifest File that describes the contents of the selected MVCs. For more information, see [“Exporting and Importing by Storage Class” on page 86](#).
-

Note – If you export by:

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

Exporting and Importing by Management Class

? Example: Exporting by Management Class from the Source VSM System

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

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

1. **Identify the Management Classes used for export.**
2. **Export by Management Class:**

```
//EXPORT EXEC PGM=SLUADMIN, PARM='MIXED' REGION=6M
//STEPLIB DD DSN=hlq.SEALINK, DISP=SHR
//MOVE1 DD DSN=hlq.REMOTE2, DISP=(, CATLG, DELETE),
// UNIT=SYSDA, SPACE=(CYL,(1,1),RLSE),
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=27920)
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
EXPORT MGMT (PAY, ACCOUNT) MANIFEST (MOVE1)
```

FIGURE 4-1 EXPORT by Management Class

In [FIGURE 4-1](#), the output manifest file is MOVE1, which is required for the import. Because we exported by Management Class, EXPORT consolidates (makes copies of) the selected VTVs on *export MVCs*. The export MVCs are marked as read-only and as exported in the CDS, and are now available for ejection from a source system LSM.

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

Caution – Schedule the export for a time when the exported data is not being updated!

3. **Remove the MVCs for export from the MVC pool.**
For more information, see *Managing HSC and VTCS*.
4. **Eject the MVCs for export from a source VSM system LSM.**
For more information, see *Managing HSC and VTCS*.

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

After the export, the source system retains the CDS records of the exported VTVs and MVCs. The export MVCs are marked as exported and readonly in the source system CDS. At this point, you have two choices, depending on why you exported the VTVs:

- ? **If you exported the VTVs to provide a backup copy at a second site**, leave the VTVs as readonly in the source system CDS so they cannot be updated.
- ? **If you are permanently moving the exported VTVs to a second site**, then scratch or otherwise make them unavailable in the source system CDS. Use the HSC scratch utilities or the LCM SYNCVTV function to scratch exported VTVs.

? Example: Importing by Management Class into the Target VSM System

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

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

1. If the VTVs and MVCs you are importing are not in the target system CDS, redo your POOLPARM/VOLPARM definitions to add these volsters...

...as described in *Configuring HSC and VTCS*. If you're actually merging or consolidating data centers, adding volsters is usually a gimme. If necessary, also increase the CDS size on the target VSM system. For more information, see *Configuring HSC and VTCS* or *Managing HSC and VTCS*.

What if there were duplicate VTV volsters across the source and target systems? In general, do the following:

- ? If the source system has more current VTVs with the same volsters as those on the target system, specify `REPLACE (ALL)`.
- ? If you are moving the VTVs from the source to the target system (first time export/import), specify `REPLACE (NONE)`. In this case, you have to decide what to do with the duplicate VTVs on a case by case basis.

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

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

3. Optionally, do a "validate" run of `IMPORT`:

```
//IMPORT EXEC PGM=SLUADMIN, PARM='MIXED' REGION=6M
//STEPLIB DD DSN=hlq.SEALINK, DISP=SHR
//REMOTE1 DD DSN=hlq.REMOTE1, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
IMPORT MANIFEST(MOVE1) NOUPDATE
```

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

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

- ? The Manifest File is the export Manifest specified in [Step 2](#) on [on page 82](#).
- ? `REPLACE(NONE)` (the default) specifies that VTCS does not overwrite duplicate VTVs.
- ? `IMMDRAIN(NO)` (the default) specifies that VTCS does not drain all imported VTVs to VTSS space.
- ? `NOUPDATE` specifies that the CDS is not updated (validate run only).
- ? `INACTCDS` is not specified, so HSC is active.

Doing a validate run optional but **highly recommended**, because you really want to see what's going to happen before you push the button for real. Study carefully the Import Report. Like what you see? Continue with [Step 4](#) on [on page 85](#).

Note –

- ? `IMPORT` is valid only if `FEATures VSM(ADVMGMT)` is specified.

- 7 Ensure that the "to" CDS has the same features (enabled by CDS level) as the "from" CDS. For example, if the "from" CDS has Large VTV page sizes enabled and 2/4 Gb VTVs have been created, then the "to CDS" must have the same capabilities, otherwise the import fails.
-

4. Do an actual run of IMPORT:

```
//IMPORT EXEC PGM=SLUADMIN, PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
IMPORT MANIFEST(MOVE1) REPLACE(ALL)
```

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

[FIGURE 4-3](#) shows example JCL to run the IMPORT utility where, as in the "validate" run, REPLACE(NONE) (the default) specifies that VTCS does not overwrite duplicate VTVs.

Note – What if you want to return the MVCs to the source system? If so, you can specify IMMDRAIN(YES) to drain the import MVCs.

5. Adjust your VTV definitions as needed.

For example, you need to define any new VTVs to the target system's TMS.

6. Do one of the following:

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

Next, add the imported MVCs to the MVC pool as described in *Managing HSC and VTCS*. At this point, the MVCs can be reclaimed, drained, migrated to, recalled from, and so forth.

- 7 If you specified IMMDRAIN(YES) in [Step 4](#), you can return the MVCs to the source system.

Exporting and Importing by Storage Class

? Example: Exporting by Storage Class from the Source VSM System

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

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

- 1. Identify the Storage Classes used for export.**
- 2. Export by Storage Class:**

```
//EXPORT EXEC PGM=SLUADMIN, PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//MOVE2 DD DSN=h1q.REMOTE2, DISP=(,CATLG,DELETE),
// UNIT=SYSDA, SPACE=(CYL,(1,1),RLSE),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
EXPORT STOR(OFF1,OFF2) MANIFEST(MOVE2)
```

FIGURE 4-4 EXPORT by Storage Class

In [FIGURE 4-1](#), the output manifest file is MOVE2, which are required for the import. Because we exported by Storage Class, a Manifest File is created by no VTV consolidation occurs. The export MVCs are marked as read-only and as exported in the CDS, and are now available for ejection from a source system LSM.

The VTVs were resident on the MVCs removed from the LSM can still be used provided they are resident on other MVCs.

Caution – Schedule the export for a time when the exported data is not being updated!

- 3. Remove the MVCs for export from the MVC pool.**
For more information, see *Managing HSC and VTCS*.
- 4. Eject the MVCs for export from a source VSM system LSM.**
For more information, see *Managing HSC and VTCS*.
- 5. If desired, at the source system, scratch or make unavailable the exported VTVs or reuse the data they contain.**

After the export, the source system retains the CDS records of the exported VTVs and MVCs. The export MVCs are marked as exported and readonly in the source system CDS. At this point, you have two choices, depending on why you exported the VTVs:

- ? **If you exported the VTVs to provide a backup copy at a second site**, leave the VTVs as readonly in the source system CDS so they cannot be updated.
- ? **If you are permanently moving the exported VTVs to a second site**, then scratch or otherwise make them unavailable in the source system CDS. Use the HSC scratch utilities or the LCM SYNCVTV function to scratch exported VTVs.

? Example: Importing by Storage Class into the Target VSM System

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

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

1. If the VTVs and MVCs you are importing are not in the target system CDS, redo your POOLPARM/VOLPARM definitions to add these volsers...

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

What if there were duplicate VTV volsers across the source and target systems? In general, do the following:

- ? If the source system has more current VTVs with the same volsers as those on the target system, specify REPLACE (ALL) .
- ? If you are moving the VTVs from the source to the target system (first time export/import), specify REPLACE (NONE) . In this case, you have to decide what to do with the duplicate VTVs on a case by case basis.

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

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

3. Optionally, do a "validate" run of IMPORT:

```
//IMPORT EXEC PGM=SLUADMIN, PARM='MIXED' REGION=6M
//STEPLIB DD DSN=hlq.SEALINK, DISP=SHR
//REMOTE1 DD DSN=hlq.REMOTE1, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
IMPORT MANIFEST(REMOTE1) NOUPDATE
```

FIGURE 4-5 IMPORT utility example: validation import run, HSC is active

[FIGURE 4-2](#) shows example JCL to run the IMPORT utility where:

- ? The Manifest File is the export Manifest specified in [Step 2](#) on [page 82](#).
- ? REPLACE(NONE) (the default) specifies that VTCS does not overwrite duplicate VTVs.
- ? IMMDRAIN(NO) (the default) specifies that VTCS does not drain all imported VTVs to VTSS space.
- ? NOUPDATE specifies that the CDS is not updated (validate run only).
- ? INACTCDS is not specified, so HSC is active.

Doing a validate run optional but **highly recommended**, because you really want to see what's going to happen before you push the button for real. Study carefully the Import Report. Like what you see? Continue with [Step 4](#) on [page 85](#).

Note –

- ? IMPORT is valid only if FEATURES VSM(ADVMGMT) is specified.

- 7 Ensure that the "to" CDS has the same features (enabled by CDS level) as the "from" CDS. For example, if the "from" CDS has Large VTV page sizes enabled and 2/4 Gb VTVs have been created, then the "to CDS" must have the same capabilities, otherwise the import fails.
-

4. Do an actual run of IMPORT:

```
//IMPORT EXEC PGM=SLUADMIN, PARM='MIXED' REGION=6M
//STEPLIB DD DSN=h1q.SEALINK, DISP=SHR
//REMOTE1 DD DSN=h1q.REMOTE1, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
IMPORT MANIFEST(REMOTE1)
```

FIGURE 4-6 IMPORT utility example: validation import run, HSC is active

[FIGURE 4-3](#) shows example JCL to run the IMPORT utility where, as in the "validate" run, REPLACE (NONE) (the default) specifies that VTCS does not overwrite duplicate VTVs.

Note – What if you want to return the MVCs to the source system? If so, you can specify IMMDRAIN (YES) to drain the import MVCs.

5. Adjust your VTV definitions as needed.

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

6. Do one of the following:

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

Next, add the imported MVCs to the MVC pool as described in *Managing HSC and VTCS*. At this point, the MVCs can be reclaimed, drained, migrated to, recalled from, and so forth.
- 7 If you specified IMMDRAIN (YES) in [Step 4](#), you can return the MVCs to the source system.