

# Virtual Library Extension

---

## Configuring the MVS Host Software for VLE

Version 1.1

E24872-06



Revision 06

## Configuring the MVS Host Software for VLE

E24872-06

Copyright © 2011, 2012 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.

---

---

# Table of Contents

<b>Preface</b>	7
Audience	7
<b>1 What is Virtual Library Extension?</b>	9
Single Node VLE Configuration	10
What's New in VLE 1.1?	11
VLE Hardware and Software	13
MVS Host Software Requirements for VLE	15
<b>2 Configuring the MVS Host Software</b>	17
Key Configuration Values	18
Subsystem Name	18
VLE Data Port and VSM5 IFF3 Card Target IP Addresses	18
IP Addresses of VLE Ports for Host (UII) Communication	18
VMVC Volsers	18
MVS Host Software Configuration Tasks	19
Updating the SMC OMVS RACF Security Entry	19
Modifying the SMC SCMDS file	20
Updating the VTCS CONFIG Deck to Define VLE	21
Defining the VLE VMVCs to the MVS Host Software and Including VMVCs in an MVC Pool	24
Creating VMVC Volume Pools (6.2)	25
Updating the MVS Host Software Policies	26
<b>3 Using the Concurrent Disaster Recovery Test Software with VLE</b>	29
Overview	29
How to Use CDRT	31
CDRT Restrictions	32
Restrictions that The CDRT Software Enforces	32
Optimizing Access to Test and Production Resources	33
Running a DR Test	34
To run a DR test:	34
Cleaning Up After a DR Test	39
To remove DR test data from the VTSS buffer:	39
To resume normal operations:	41
Operational Scenarios	41
Scenario 1: Production and Test Sites, ACS and VLE at Each Site	42

Additional Operations for Scenario 1: .....	45
Scenario 2: Production and Test Sites, VLE Only at Each Site .....	46
Additional Operations for Scenario 2: .....	49
Scenario 3: Clustered VTSSs with Production and DR Test Sites .....	50
Preparing Your VTSS Cluster for a DR Test .....	51
<b>4 Command Reference .....</b>	<b>55</b>
ACTMVCGN - 7.0 and Above .....	56
Syntax .....	56
Parameters .....	57
Additional JCL Requirements .....	57
COMMtest - 6.2 and Above .....	58
Syntax .....	58
Parameters .....	59
CONFIG RECLAIM - 6.2 and Above .....	60
Syntax .....	60
Parameters .....	60
CONFIG RTD - 6.2 and Above .....	62
Syntax .....	62
Parameters .....	63
CONFIG STORMNGR - 7.0 and Above .....	65
Syntax - 7.0 .....	65
Syntax - 7.0 and 7.1 .....	65
Parameters .....	66
CONFIG TAPEPLEX - 6.2 and Above .....	67
Syntax - 7.0 and 7.1 .....	67
Syntax - 6.2 .....	67
Display STORMNGR .....	68
Interfaces: .....	68
Subsystem Requirements: .....	68
Description .....	68
Syntax .....	68
Parameters .....	68
Output .....	69
DRTEST .....	71
Syntax .....	71
Parameters .....	72
Usage .....	74
JCL Requirements .....	75
Route - 6.2 and Above .....	78
Syntax .....	78
Parameters .....	79
SERVER - 6.2 and Above .....	80
Syntax .....	80
Parameters .....	81
STORclas Control Statement - 6.2 and Above .....	84
Syntax .....	84
Parameters .....	85
STORMNGR - 6.2 and Above .....	88

Syntax .....	88
Parameters .....	89
DISPLAY Command and Report Outputs .....	90
Display CONFIG Output .....	90
Display MIGrate DEtail Output .....	91
Display MVCPool Output .....	92
MVC RPT/MVC PL RPT/DISPLAY MVC Output .....	94
New/Updated Messages .....	96
HSC/VTCS Messages .....	96
SMC Messages .....	98



---

# Preface

## Audience

This guide is for StorageTek or customer personnel who are responsible for configuring the MVS host software for Oracle's StorageTek™ Virtual Library Extension (VLE).





---

## What is Virtual Library Extension?

Oracle's StorageTek™ Virtual Library Extension (VLE) is back-end disk storage for VTSS. VLE provides:

- An additional storage tier in the VSM solution. VTVs can now migrate from VTSS to VLE to provide fast access to recent data. Additionally, VTVs can transition from VLE storage to tape media (MVCs) for long term archive. You can control how VTVs are migrated and archived via the existing HSC Management and Storage Classes, providing full backward compatibility with previous configurations.
- Back-end disk storage shared between multiple VTSS systems ensuring high-availability access to data.

**Note –** For VLE 1.1, a “VLE” is a collection of nodes interconnected with a private network, where single-node through seven-node VLEs are supported.

To VTCS, a VLE looks like a tape library except that the VTVs are stored in Virtual Multi-Volume Cartridges (VMVCs) on disk. With VLE, you can configure either a VLE and tape or a VLE only (for example, with Tapeless VSM configurations) back-end VTV storage solution. A VTSS can migrate VTVs to and recall them from a VLE, just as is done with a real tape library.

### Caution –

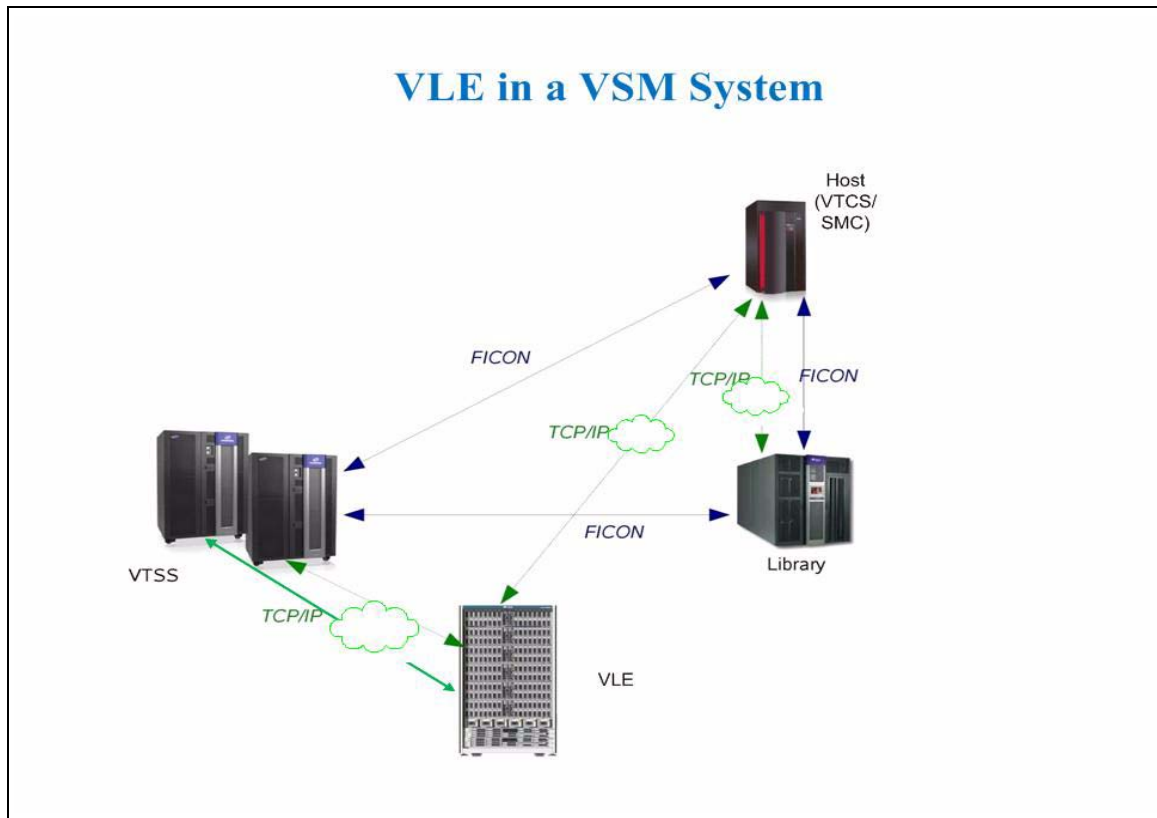
- **Note that** stopping SMC stops VTCS from sending messages to the VLE, which effectively stops data transfer. Therefore, you should ensure that VTCS activity is quiesced or VTCS is terminated before stopping SMC.
- You cannot use AT-TLS with the SMC HTTP server if you are using VLE.
- Note that in Tapeless VSM configurations, if you have only a single-node VLE attached to a specific VTSS and that VLE goes offline, you lose access to any VTVs migrated to the VLE that are not resident in the VTSS until the VLE comes back online.

The VLE solution consists of:

- Virtual Tape Storage Subsystem (VTSS) hardware and microcode (VSM5s with IFF3 cards).
- Virtual Tape Control Subsystem (VTCS) software and Storage Management Component (SMC).
- VLE hardware and software.

# Single Node VLE Configuration

FIGURE 1-1 shows a single node VLE configuration.



**FIGURE 1-1** Single VLE in a VSM System

As [FIGURE 1-1](#) shows:

- Multiple TCP/IP connections (between the VTSS's IFF3 card IP ports and the VLE's IP ports) are supported as follows:
  - A single VLE can connect up to 8 VTSSs, so VTSSs can share VLEs.
  - A single VTSS can connect to up to 4 VLEs to increase buffer space for heavy workloads.
- A single VTSS can be attached to:
  - Only RTDs
  - Only other VTSSs (clustered)
  - Only VLEs
  - Any combination of the above.
- TCP/IP is the only supported protocol for connections between the VLE and the VTSS and for connections between the VLE and hosts running SMC and VTCS.

# What's New in VLE 1.1?

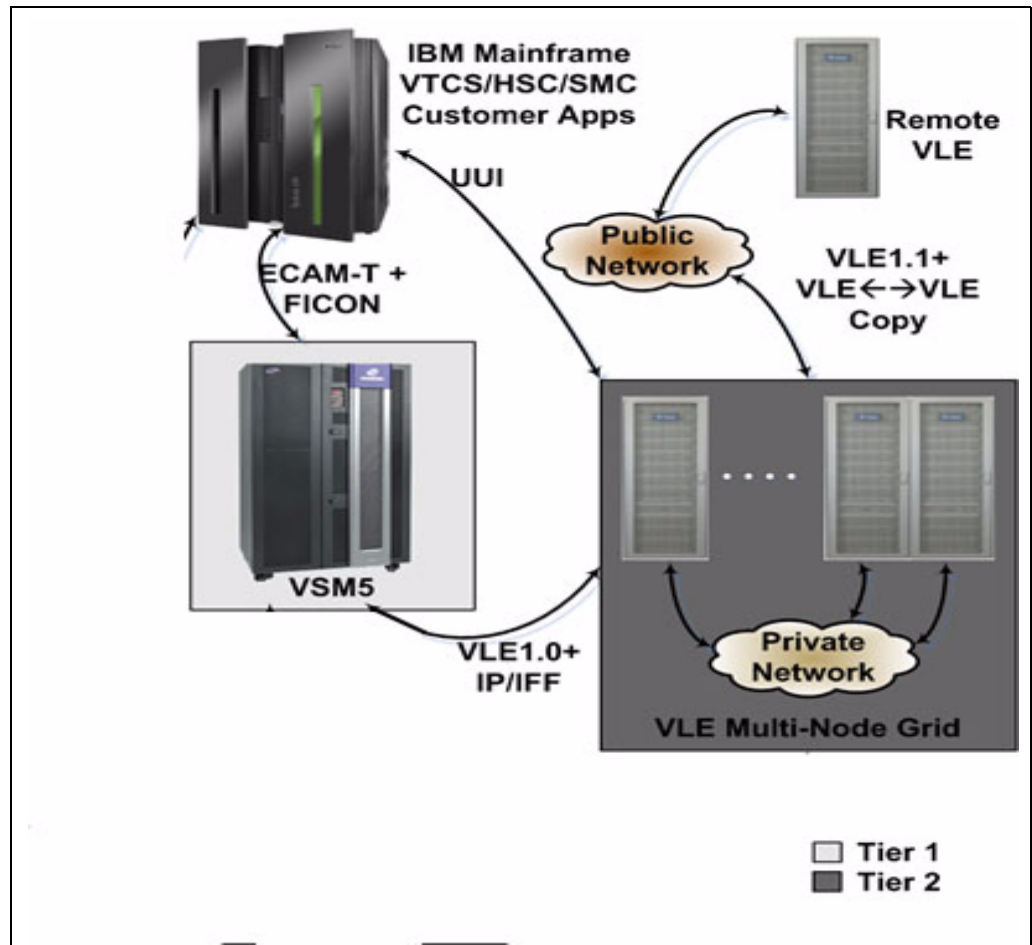
VLE 1.1 provides the following enhancements:

**Enables massive scaling of the VLE storage system.** VLE 1.0 is *single node system*. With VLE 1.1, you can construct *multi-node systems* that can consist of one to seven nodes, with multiple nodes interconnected by a private network. A multi-node VLE appears to SMC/VTCS as a single VLE system. These interconnected nodes allow a VLE 1.1 complex to scale between 330TB (for a single VLE with two JBODs) and 9.2 PB (for a fully populated seven node system).

**Note –** These are effective capacities, assuming 4:1 compression.

- **The VLE storage system** can now manage data transfers independently of the VTSS, which frees VTSS resources for front-end (host) workload, which improves the overall VTSS through-put. For example, in a VSM system with VLE 1.1:
  - If you want to migrate two VTV copies to separate, interconnected VLEs, you can use Management and Storage Classes to migrate one copy using VTSS to VLE connections and the second copy using VLE to VLE connections. The second migration, therefore, uses no VTSS resources.
  - Housekeeping tasks like a reclaim or audit of a VMVC no longer consume VTSS resources.
- **Encryption**, which enables encryption of VMVCs written to the VLE system.
- VTSS to VLE bandwidth enhancements; see [“VLE 1.1” on page 15](#).

[FIGURE 1-2 on page 12](#) shows a VLE 1.1 complex, where the nodes are cross connected into a dedicated 10GE switch so that each node can access any other node in the complex.



**FIGURE 1-2** VLE Multi-Node Complex

# VLE Hardware and Software

The VLE, which is a factory-assembled unit in a Sun Rack II Model 1242, consists of the following hardware:

- A server built on a 4470 platform.
- Four 1GigE ports for a combination of SMC UII connections and service connections.
- A service (ILOM) port.
- Four Quad-port 1GigE cards, which provide 16 ethernet ports for data transfer.
- J4410 JBODs in a ZFS RAID array, available in effective capacities of 330TB, 660TB, 990TB, or 1320TB (assuming a 4 to 1 compression ratio when the data is migrated to the VLE).
- Two dual-port 10GigE NIC cards per server, which are required for the internal network connections for VLEs with 3 or more nodes.

The VLE software consists of:

- Oracle Solaris 11 Operating System.
- ZFS file system and MySQL database.
- The VLE application software.

FIGURE 1-3 shows the VLE subsystem architecture.

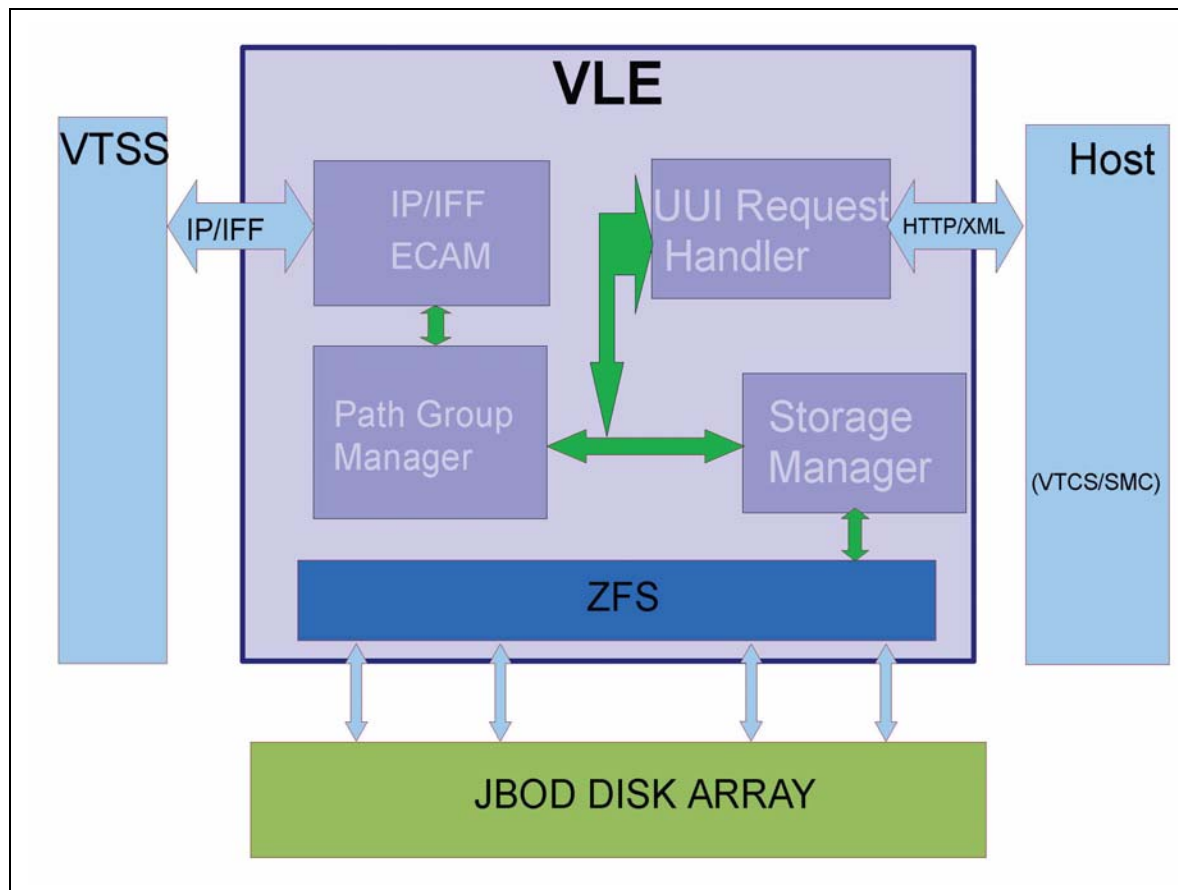


FIGURE 1-3 VLE Subsystem Architecture

As [FIGURE 1-3 on page 13](#) shows, the VLE application software is comprised of:

- HTTP/XML is the data protocol for host to VLE communications.
- The Universal User Interface (UII) Request Handler, which processes UII requests from and produces responses to Storage Management Component (SMC) and Virtual Tape Control Software (VTCS). The UII Request Handler determines which VLE components are used to service a request.

UII Request Handler calls:

- The PathGroup Manager to schedule VTV migrates and recalls. The PathGroup Manager manages all Path Groups, where each Path Group manages a single VTV data transfer between the VTSS and the VLE.
- The Storage Manager to schedule all report generation.
- The VLE Storage Manager component manages the VMVC/VTV data and meta data on the VLE. The VLE Storage Manager stores VTV data on and retrieves it from the ZFS on the JBOD array.
- TCP/IP/IFF is the data protocol for host to VLE communications, where the IP/IFF/ECAM component handles communications between the VTSS and the VLE.

# MVS Host Software Requirements for VLE

## VLE 1.0

Support for VLE 1.0 is provided by any of the following (plus software updates):

- HSC/VTCS/SMC 6.2
- ELS 7.0
- ELS 7.1

See My Oracle Support for the VTCS and SMC software updates that are additionally required for VLE support.

## VLE 1.1

VLE 1.1 is supported by the MVS host software in three different modes:

- **Toleration mode.** The VLE 1.1 system functions just like a 1.0 system. The VSM system can exploit the additional *capacity* of a 1.1 system but cannot exploit the VLE 1.1 backend *data management* capabilities.
- **Exploitation mode,** where the additional capacity and the backend data management capabilities are fully exploited. The number of backend data operations is, however, restricted by the number of paths to the VLE. For example, a VTV migration that simply requires the VLE to perform a copy operation will wait for a free path from the VTSS to the VLE before starting the copy. Exploitation mode is provided by:

- ELS 7.0 plus PTF 11h16C1
- ELS 7.1 plus PTF 11h16e9

**Note –** VLEs are configured via the `CONFIG STORMNGR` statement, **not** the `CONFIG TAPEPLEX STORMNGR` parameter.

- **Acceleration mode.** The VLEs are defined via a `CONFIG STORMNGR` statement, **not** the `CONFIG TAPEPLEX STORMNGR` parameter. The `CONFIG STORMNGR` statement specifies the VLEs that VTCS will connect to. Additionally, for each VLE, the `CONFIG STORMNGR VLEDEV` parameter defines the number and the names of the RTD devices that the VLE will emulate. The more devices defined, the greater the greater level of concurrent activities VTCS schedules on the VLEs. It is possible and desirable to define **more** devices on a VLE than there are paths from the VTSS to the VLE, which increases the amount of concurrent work scheduled to the VLE.

Acceleration mode is provided by:

- ELS 7.1 plus PTF 11h16e9 and PTF 11h16j6





---

## Configuring the MVS Host Software

This chapter provides the MVS host software configuration for VLE as described in the following sections:

- [“Key Configuration Values” on page 18](#)
- [“MVS Host Software Configuration Tasks” on page 19](#)

# Key Configuration Values

The following sections describes values required for software configuration that must match values that are typically already set in the hardware configuration and recorded in the `IP_and_VMVC_Configuration.xls` worksheet.

## Subsystem Name

The subsystem name of the VLE, which is set by the VLE installation scripts, is specified in the following:

- Either the VTCS `CONFIG TAPEPLEX STORMNGR` parameter or the `CONFIG STORMNGR NAME` parameter.
- The VTCS `CONFIG RTD STORMNGR` parameter.
- The SMC `STORMNGR NAME` parameter.
- The SMC `SERVER STORMNGR` parameter.
- The HSC `STORCLAS STORMNGR` parameter.

## VLE Data Port and VSM5 IFF3 Card Target IP Addresses

These IP addresses are initially set on the VSM5 DOP `IFF IP Configuration Status` panel and at the VLE GUI and the values must match. On the DOP panel, they are set as IP addresses with their equivalent `c:ip` addresses shown, which are required for the `CONFIG RTD IPIF` parameter.

## IP Addresses of VLE Ports for Host (UI) Communication

These addresses are required for the SMC `SERVER IP` parameter.

## VMVC Volsers

Required to define VMVCs to SMC/VTCS, method of definition depends on the software version, see [“Defining the VLE VMVCs to the MVS Host Software and Including VMVCs in an MVC Pool”](#) on page 24.

# MVS Host Software Configuration Tasks

Adding VLE to a VSM system requires the tasks described in the following sections:

- [“Updating the SMC OMVS RACF Security Entry” on page 19](#)
- [“Modifying the SMC SCMDs file” on page 20](#)
- [“Updating the VTCS CONFIG Deck to Define VLE” on page 21](#)
- [“Defining the VLE VMVCs to the MVS Host Software and Including VMVCs in an MVC Pool” on page 24](#)
- [“Updating the MVS Host Software Policies” on page 26](#)

## Updating the SMC OMVS RACF Security Entry

The VLE requires SMC to have an OMVS RACF security entry in order to have a TCP/IP connection to the host.

OMVS is a segment associated with the RACF userid. The SMC started task must have a userid associated with OMVS, either in the RACF `STARTED` class definition or the `ICHRIN03 LNKLIST` module. The userid associated with the SMC task needs to have an OMVS segment defined to it within RACF as follows:

```
ADDUSER userid  
DFLTGRP (groupname) OWNER (owner) OMVS (UID (uidnumber) )
```

Or, if the userid already exists but does not have an OMVS segment:

```
ALTUSER userid OMVS (UID (uidnumber) )
```

## Modifying the SMC SCMDS file

SMC manages all communication between VTCS and VLE, so SMC must know how to connect to the VLE server. You do so by adding an SMC STORMNGR statement for each VLE system plus one or more SMC SERVER statements that define the TCP/IP control paths for the VLE. For 7.0 and above, you may want to do this in your SMC CMDS file as shown in [CODE EXAMPLE 2-1](#).

```
TAPEPLEX NAME (TMVSA) LOCSUB (SLS0)
SERVER NAME (ALTSERV) TAPEPLEX (TMVSA) +
HOSTNAME (MVSX) PORT (8888)
STORMNGR NAME (VLE1)
SERVER NAME (VLESERV1) +
STORMNGR (VLE1) IP (192.168.1.10) PORT (60000)
```

### CODE EXAMPLE 2-1 SMC Commands for VLE

[CODE EXAMPLE 2-1](#) contains:

- A TAPEPLEX statement, which defines a single TapePlex, TMVSA, with an HSC/VTCS running on the same MVS host (SLS0).
- A SERVER statement, which defines a backup HSC/VTCS subsystem (ALTSERV) running on another host.
- A STORMNGR command that defines a VLE (VLE1) .
- A second SERVER command that defines a UII communication path to the VLE, where:
  - The server name is VLESERV1.
  - The STORMNGR parameter value is VLE1 .
  - The IP parameter value is the VLE port IP address of 192.168.1.10 for UII communications.
  - The PORT parameter value is 60000; this value is always used for the SERVER PORT parameter for SMC communication with a VLE.

## Updating the VTCS CONFIG Deck to Define VLE

You must update VTCS CONFIG deck to define the VLE and the connectivity from the VTSS systems to the VLE. As described in “[VLE 1.1](#)” on [page 15](#), VTCS can drive VLE 1.1 in three different modes:

- **Toleration mode.** The VLE 1.1 system functions just like a 1.0 system. VLE 1.0 provides a single method of defining the VLE system to VTCS, which is via a CONFIG TAPEPLEX statement. This CONFIG TAPEPLEX statement defines the TapePlex that VTCS is running under and provides the list of defined VLEs on the STORMNGR parameter as shown in [CODE EXAMPLE 2-2](#).

```
TAPEPLEX THISPLEX=TMVSA STORMNGR=VLE1
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD  NAME=VL1RTD1 STORMNGR=VLE1 IPIF=0A:0
RTD  NAME=VL1RTD2 STORMNGR=VLE1 IPIF=0A:1
RTD  NAME=VL1RTD3 STORMNGR=VLE1 IPIF=0I:0
RTD  NAME=VL1RTD4 STORMNGR=VLE1 IPIF=0I:1
RTD  NAME=VL1RTD5 STORMNGR=VLE1 IPIF=1A:0
RTD  NAME=VL1RTD6 STORMNGR=VLE1 IPIF=1A:1
RTD  NAME=VL1RTD7 STORMNGR=VLE1 IPIF=1I:0
RTD  NAME=VL1RTD8 STORMNGR=VLE1 IPIF=1I:1
VTD  LOW=6900 HIGH=69FF
```

### CODE EXAMPLE 2-2 CONFIG Example for Toleration (1.0) Mode

In [CODE EXAMPLE 2-2](#), note:

- The CONFIG TAPEPLEX statement, which defines TMVSA as the TapePlex that VTCS is running under and the connection to VLE1.
- The CONFIG RTD statements for VTSS1, which specify:
  - The connections to VLE1.
  - The IPIF value and an RTD name for each IFF target to VLE port connection.

- **Exploitation mode**, where the additional capacity and the backend data management capabilities are fully exploited. The VLEs are defined via a CONFIG STORMNGR statement, **not** the CONFIG TAPEPLEX STORMNGR parameter. Instead, the CONFIG STORMNGR statements specify the VLEs that VTCS will connect to. The number of backend data operations is, however, restricted because there is a single VRTD on each VLE.

```
TAPEPLEX THISPLEX=TMVSC
STORMNGR NAME=VLE1
STORMNGR NAME=VLE2
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=VL1RTD1 STORMNGR=VLE1 IPIF=0A:0
RTD NAME=VL1RTD2 STORMNGR=VLE1 IPIF=0A:1
RTD NAME=VL1RTD3 STORMNGR=VLE1 IPIF=0I:0
RTD NAME=VL1RTD4 STORMNGR=VLE1 IPIF=0I:1
RTD NAME=VL1RTD5 STORMNGR=VLE2 IPIF=1A:0
RTD NAME=VL1RTD6 STORMNGR=VLE2 IPIF=1A:1
RTD NAME=VL1RTD7 STORMNGR=VLE2 IPIF=1I:0
RTD NAME=VL1RTD8 STORMNGR=VLE2 IPIF=1I:1
VTD LOW=6900 HIGH=69FF
```

### CODE EXAMPLE 2-3 Exploitation Mode CONFIG Example

In [CODE EXAMPLE 2-3](#), note:

- The CONFIG TAPEPLEX statement, which defines TMVSC as the TapePlex that VTCS is running under.
- The CONFIG STORMNGR statements, which define TMVSC's connection to VLE1 and VLE2.
- The CONFIG RTD statements for VTSS1, which specify:
  - The connections to the two VLEs.
  - The IPIF value and an RTD name for each IFF target to VLE connection.

- **Acceleration mode.** The VLEs are defined via a CONFIG STORMNGR statement, **not** the CONFIG TAPEPLEX STORMNGR parameter. The CONFIG STORMNGR statement specifies the VLEs that VTCS will connect to. Additionally, for each VLE, the CONFIG STORMNGR VLEDEV parameter defines the number and the names of the RTD devices that the VLE will emulate. The more devices defined, the greater the greater level of concurrent activities VTCS schedules on the VLEs. It is possible and desirable to define **more** devices on a VLE than there are paths from the VTSS to the VLE, which increases the amount of concurrent work scheduled to the VLE.

```
TAPEPLEX THISPLEX=TMVSC
STORMNGR NAME=VLE1 VLEDEV(S100-S11F)
STORMNGR NAME=VLE2 VLEDEV(S200-S21F)
VTSS NAME=VTSS1 LOW=70 HIGH=80 MAXMIG=8 MINMIG=4 RETAIN=5
RTD NAME=VL1RTD1 STORMNGR=VLE1 IPIF=0A:0
RTD NAME=VL1RTD2 STORMNGR=VLE1 IPIF=0A:1
RTD NAME=VL1RTD3 STORMNGR=VLE1 IPIF=0I:0
RTD NAME=VL1RTD4 STORMNGR=VLE1 IPIF=0I:1
RTD NAME=VL1RTD5 STORMNGR=VLE2 IPIF=1A:0
RTD NAME=VL1RTD6 STORMNGR=VLE2 IPIF=1A:1
RTD NAME=VL1RTD7 STORMNGR=VLE2 IPIF=1I:0
RTD NAME=VL1RTD8 STORMNGR=VLE2 IPIF=1I:1
VTD LOW=6900 HIGH=69FF
```

#### CODE EXAMPLE 2-4 Acceleration Mode CONFIG Example

In [CODE EXAMPLE 2-4](#), note:

- The CONFIG TAPEPLEX statement, which defines defines TMVSC as the TapePlex that VTCS is running under.
- The CONFIG STORMNGR statements, which define TMVSC's connection to VLE1 and VLE2 and the number of VLE devices via the VLEDEV parameter. In this example, each VLE has 32 emulated devices, which allows VTCS to schedule up to 32 processes on each VLE. Note that VLEDEV supports up to a maximum of 96 emulated devices.
- The CONFIG RTD statements for VTSS1, which specify:
  - The connections to the two VLEs.
  - The IPIF value and an RTD name for each IFF target to VLE connection.

### Specifying the Reclamation Policy for VMVCS

VLE MVC media (VMVCs) is subject to fragmentation and must be reclaimed just like real MVCs. The VMVC reclaim process, however, uses far fewer resources than a standard reclaim. The reclaim threshold for a VMVC is specified via the CONFIG RECLAIM VLTHRES parameter. The lower that you set VLTHRES, the more frequent VTCS will run reclaim on the VMVCs and the greater the effective capacity of the VMVS (less fragmentation).

# Defining the VLE VMVCs to the MVS Host Software and Including VMVCs in an MVC Pool

VMVC volsters must be defined both to the MVS host software and to the VLE. The VMVCs are defined to the VLE as part of the VLE configuration. The following sections tell how to define the VMVCs to the MVS host software.

## ▼ Creating VMVC Volume Pools (7.0 and Above)

### 1. Code HSC POOLPARM/VOLPARM statements to define the VMVC pools.

For example, to define two separate pools for VLE1 and VLE2:

```
POOLPARM NAME (LEPOOL1) TYPE (MVC)
VOLPARM VOLSER (VL0000-VL880)
```

```
POOLPARM NAME (LEPOOL2) TYPE (MVC)
VOLPARM VOLSER (VL2000-VL2880)
```

### 2. Run SET VOLPARM to validate the POOLPARM/VOLPARM statements.

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

APPLY (NO) validates the statements without loading them. If you like the results, go to [Step 3](#). Otherwise, rework your volume definitions, rerun this step, and if the definitions are valid, then go to [Step 3](#).

### 3. Run SET VOLPARM to load the POOLPARM/VOLPARM statements.

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



## ▼ Creating VMVC Volume Pools (6.2)

### 1. Code HSC VOLATTR statements to define the VMVCs to HSC.

For example, to define two separate VMVC volser ranges for VLE1 and VLE2:

```
VOLATTR SERIAL (VL0000-VL880)  
VOLATTR SERIAL (VL2000-VL2880)
```

### 2. In your VTCS CONFIG JCL, code MVCVOL statements to define the VMVCs to VTCS.

For example:

```
MVCVOL LOW=VL0000 HIGH=VL880  
MVCVOL LOW=VL2000 HIGH=VL2880
```

### 3. Code HSC MVCPOOL statements to define the VMVC pools.

For example:

```
MVCPOOL VOLSER (VL0000-VL880)  
MVCPOOL VOLSER (VL2000-VL2880)
```

### 4. Run the VT MVCDEF and VOLDEF commands to activate the updated data set, for example:

```
.VT MVCDEF DSN (VSM.VMVCPOOL  
VOLDEF DSN (HSC.VOLATTR)
```

## Updating the MVS Host Software Policies

The following sections tell how to update the MVS host software policies to direct data to the VLE system.

### Creating Storage and Management Classes for VLE

*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 *Storage Classes*, which specify where migrated VTVs reside. The HSC STORclas control statement defines a Storage Class and its attributes.

You specify the VLE system as the destination for migrated VTVs via the STORCLAS STORMNGR keyword. For example:

```
STOR NAME (VLOCAL) STORMNGR (VLESERV1)
STOR NAME (VREMOTE) STORMNGR (VLESERV2)
```

The preceding statements define a “local” Storage Class (VLOCAL) on the VLSERV1 and a “remote” Storage Class (VREMOTE) on the on the VLSERV2. As these STORCLAS statements specify, all migrations to storage class VLOCAL or VREMOTE **must** go to the specified VLEs. You can be less restrictive than this if desired. For example, if you define an MVCPOOL that contains both VMVCs and MVCs you can set up the migration policies to migrate to a VLE but if the VLE becomes full or not available, to continue to migrate to real tape media (MVCs). For example the MVC pool DR is defined as follows:

```
POOLPARM NAME (DR) TYPE (MVC)
VOLPARM VOLSER (VL0000-VL0100)
VOLPARM VOLSER (ACS000-ACS099)
```

Pool DR, therefore, contains both MVCs and VMVCs. A Storage Class that specifies pool DR will migrate first to VMVCs and only use MVCs if VMVCs are not available. For example:

```
STOR NAME (DRCLASS) MVCPOOL (DR)
```

This method is valuable if you have a configuration where both an ACS and a VLE are connected to the VTSS systems.

Next, to specify migration to VLE, you specify the VLE Storage Classes you defined via the MGMTCLAS MIGPOL parameter. For example:

```
MGMT NAME (M1) MIGPOL (VLOCAL, VREMOTE)
MGMT NAME (M2) MIGPOL (DRCLASS)
```

Management Class M1 migrates one VTV copy to the “remote” VLE, one copy to the “local” VLE. Management Class M2 migrates a single VTV copy to the Storage Class that points to the “mixed” MVC pool that contains both MVCs and VMVCs.

**Note –** In addition to directing migration to a VLE also consider:

1. 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. You can use Archive Policies to archive (move) VTVs from VMVCs to MVCs as the VTVs age. For more information, see *Managing HSC and VTCS*.
2. You can use STORSEL statements to cause VTCS to preference recalls from VLE media. For more information, see *Managing HSC and VTCS*.
3. If you are running at ELS 7.0 or above, you can use HSC MIGRSEL and MIGRVTV to fine tune migration to VLE. Using these statements, you can cause migration of data in a Management Class to start to one Storage Class before another. This method is typically used to ensure that a critical DR copy is made as soon as possible. For more information, see *Configuring HSC and VTCS*.
4. On a VLE 1.1 system, if multiple VLEs are connected to each other and to the VTSS, **by default**, VTCS preferences VLE to VLE connections to make multiple VTV copies. In a DR scenario, for example, where you have a "local" and a "remote" VLE, you typically want the remote copy made as quickly as possible (using VTSS to VLE connections) and the local copy using VLE to VLE connections. The local copy, by default, uses VLE to VLE TCP/IP connections. You use the STORCLAS FROMLST parameter to specify VTSS to VLE connections for the "remote" copy. For example:

```
STOR NAME (VREMOTE) STORMNGR (VLE2) FROMLST (VTSSA1,VTSSA2)
```

In this example, for Storage Class VREMOTE, migrations to VLE2 come from either VTSSA1 or VTSSA2, thus using VTSS to VLE connections for the "remote" copy. For more information, see ["FROMLST" on page 87](#).

## Routing Data to VLE

To route data to VLE, first create an SMC POLICY command that specifies a VLE Management Class. Next, create SMC TAPEREQ statements that route the desired workload to the SMC VLE policy. For example

```
POLICY NAME (VLEWORK) MEDIA (VIRTUAL) MGMT (VLECLASS)  
TAPEREQ DSN (VLETEST.***) POLICY (VLEMIGR)
```

The preceding example assigns the VLEWORK policy to all tape datasets with an HLQ of VLETEST.



---

# Using the Concurrent Disaster Recovery Test Software with VLE

## Overview

Customers that use or maintain a Disaster Recovery (DR) site as part of a business continuance plan may want to periodically validate their ability to continue normal production processing before an actual disaster occurs; other customers don't have a choice and must periodically demonstrate the readiness of their business continuance model to satisfy insurance requirements and/or their auditors.

Using the Concurrent Disaster Recovery Test (CDRT) facility, now an integrated feature of StorageTek ELS software, those businesses currently using StorageTek Streamline and/or Nearline (real hardware) tape libraries, VSM (virtual hardware), and associated software (HSC, VTCS) can validate their real and virtual tape business continuance capability without the need to purchase additional hardware and software.

CDRT supports a parallel test of production hosts and applications with simultaneous access to production data by both the production and the DR test systems.

Key CDRT concepts include the following:

- Using CDRT, a DR test can execute with real hardware, virtual hardware, or both.
- CDRT, HSC, and VTCS programmatically enforce certain functional restrictions during the CDS preparation and actual DR test in an attempt to ensure system integrity.
- CDRT logically separates a portion of existing production real and virtual hardware and tape volume pools for the period of the DR test. This allows testing of your DR configuration while concurrently running production work, ensures the integrity of the production data, and minimizes conflicts for tape volumes and hardware resources.
- CDRT creates a test copy of the production CDS. The production ELS subsystem and the DR test ELS subsystems therefore do not communicate with each other. Changes that occur in the DR test CDS are not reflected in the production CDS copy and vice versa. The DR test hosts exercise the logically separated hardware only. The production hosts continue to use all hardware with one exception: the DR hosts have exclusive use of any logically separated VTSSs during the DR test. Other resources, like RTDs, Multiple Virtual Cartridges (MVCs) and real scratch tapes, must be controlled by defining separate pools to each set of hosts.

- A DR test can be conducted using local resources only or with a combination of local and remote resources; configurations consisting of a remote site with real and virtual hardware only, or real and virtual hardware with a mainframe processor, are also supported.
- The DR test real hardware is a minimum of one ACS. One or more ACSs can be a dummy ACS in VSM environments running without RTDs attached to a VTSS. Optionally, one or more VTSSs can be employed as DR test virtual hardware.
- At the end of a DR test, the test copy of the CDS and all data created from the DR test are typically discarded and the logically separated hardware is redeployed back to the normal production environment.

## Metadata Considerations

Fundamental to the execution of a successful DR test using CDRT is a consistent copy of the state of all tape volumes managed by ELS software and the real and virtual hardware. The consistency in the state of tape volumes between the production hosts and the DR hosts at the start of the DR test is what allows the parallel processing of customer applications. Since the CDS reflects the state of all tape volumes and resources in the real and virtual hardware, CDRT partially meets this consistency requirement for you when it makes its test copy of the CDS.

In a tape volume environment, however, quite often some of this tape volume state data (metadata) is retained and managed outside of the ELS subsystem and the real and virtual hardware. Typically, tape volume metadata (i.e. VOLSER, DSN, Expiration date, scratch status, real or virtual designation, etc.) is stored in one or more Tape Management Catalogs (TMCs), one or more z/OS catalogs and the CDS.

You must co-ordinate the creation of copies of metadata retained and managed outside of ELS (and the real and virtual hardware) with the creation of the test copy of the CDS by CDRT.

# How to Use CDRT

You have a production CDS at your production site. You want to run a DR test at your DR site. The DR test will use hardware (VTSSs and ACSs) that exist at the DR site but are normally managed by the production site. During the DR test you will have two different CDSs each with access to the hardware at the DR site. Now, what are the steps?

## At the production site:

- Set up your POOLPARM/VOLPARM definitions on your primary CDS to define a set of MVC pools and scratch subpools for exclusive use by the DR test.
- “Prime” the production CDS to define the resources that the DR test will use, by using the SLUADMIN utility command:

```
DRTEST PRIMEPRD
```

(and include the DR host, VTSS and ACS parameters). Your production CDS is now ready for the DR test.

## At the DR test site:

Assuming that you have a copy of the production CDS at the DR site, you now use it as input to create the DR test CDS by using the SLUADMIN utility command:

```
DRTEST CREATE NOUPDPRD
```

(and include the same DR host, VTSS and ACS parameters you used above)

## At the production site:

Run the SLUADMIN utility to start the DR test using the command:

```
DRTEST START
```

## At the DR test site:

Bring up the DR test systems (HSC/SMC) using the DR test CDS. Run your batch tests at the DR site.

At the conclusion of your batch tests at the DR site, perform cleanup steps as described on [page 39](#).

For more information on the DRTEST command, see *ELS Command, Control Statement, and Utility Reference*.

## CDRT Restrictions

### Warning –

- Because two versions of the CDS exist during the DR test, you must explicitly follow the restrictions in this section. Even though the CDRT software programmatically automates the preparation of the test CDS and enforces certain functional restrictions during the CDS preparation and actual DR test to attempt to ensure system integrity, you still need to adhere to the following restrictions. Otherwise, unpredictable and undesirable results can occur, such as but not limited to corrupting your production CDS and losing data!
- VTVs to be accessed by the DR test must be resident in the DR test VTSS or on an MVC in the DR ACS. You should ensure that VTVs that will be accessed by the DR test are migrated to the DR ACS.

## Restrictions that The CDRT Software Enforces

During a DR test, the CDRT software programmatically enforces the following restrictions **on the production hosts**:

- All CAPs in the DR test ACS are in manual mode.  
You must issue the HSC CAPPREF command to set the CAPs to manual prior to running the utility. The software insures that they remain in that state as long as the test is in effect.
- Any DR test VTSS(s) are offline. You must vary the DR VTSS(s) offline to the production system prior to starting the DR test. The software ensures that they remain offline as long as the DR test is in effect.  
  
**Note –** If the VTSS is offline and not used by production, the VTSS defined must have `CONFIG VTD NOVERIFY` specified with the VTD addresses.
- `FLOAT (OFF)` and `EJCTAUTO (OFF)` are set and enforced in the DR test ACS by the software.
- No ejects, moves, HSC library audits, or scratch redistributions are allowed in the DR test ACS.

During a DR test, the CDRT software programmatically enforces the following restrictions **on the DR Test hosts**:

- The non-DR test ACS(s) remain disconnected.
- All CAPs in the DR test ACS are in manual mode.
- Any non-DR test VTSS(s) remain offline.
- `FLOAT (OFF)` and `EJCTAUTO (OFF)` are enforced everywhere.
- No moves, HSC library audits, scratch updates, or scratch redistributions are allowed anywhere.
- If you DO NOT use the POOLPARM/VOLPARM feature, no scratches may be done on the DR test host.
- The non-DR VLEs are inaccessible.



## Optimizing Access to Test and Production Resources

During a DR test, it is recommended that you enforce procedures to optimize access to resources in both test and production environments. Specifically:

- Before starting the DR test, define production management classes that specify immediate migration to both production and DR test ACSs so that VTVs will be available on MVCs accessible to the DR test system as well as the production system.
- Define DR test management classes that specify a single migration copy, since only one ACS is normally available to the DR test site.
- Use the POOLPARM/VOLPARM facility to segregate both scratch subpools and MVC and/or VMVC pools between production and DR test.
- If possible, ensure that your DR test processing does not update any pre-existing VTVs (DISP=MOD or overwrite with DISP=OLD).
- Minimize contention between production jobs migrating to the DR test ACS and DR test jobs accessing VTVs on MVCs in the DR test ACS by running the ACTMVCN utility to mark active MVCs and/or VMVCs as read-only in the production environment.
- Disable MVC space reclamation (via CONFIG HOST NORECLAM) on production MVC and/or VMVCs during DR test, to preserve the contents of volumes being used by the DR test system.

## Running a DR Test

**Note** – For more information about the commands and utilities used in this procedure, see *ELS Command, Control Statement, and Utility Reference*.

### ▼ To run a DR test:

1. Define volume pools in the production CDS using the **SET VOLPARM** command and the following example **POOLPARM/VOLPARM** statements in the **SLSPARM DD**.

```
POOLPARM NAME(MVCP1) TYPE(MVC) MVCFREE(40) MAXMVC(4) THRESH(60) START(70)
VOLPARM VOLSER(T14000-T14999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARM NAME(MVCP1) TYPE(MVC) MVCFREE(40) MAXMVC(4) THRESH(60)
START(70) DRTEST
VOLPARM VOLSER(T13000-T13999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARM NAME(SCRCP1) TYPE(SCRATCH)
VOLPARM VOLSER(T11000-T11999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARM NAME(SCRCP1) TYPE(SCRATCH) DRTEST
VOLPARM VOLSER(T12000-T12999) MEDIA(T10000T1) RECTECH(T1AE)
POOLPARM NAME(SCRVTV1) TYPE(SCRATCH)
VOLPARM VOLSER(V1000-V1999) MEDIA(VIRTUAL)
POOLPARM NAME(SCRVTV1) TYPE(SCRATCH) DRTEST
VOLPARM VOLSER(V2000-V2999) MEDIA(VIRTUAL)
```

#### CODE EXAMPLE 3-5 POOLPARM/VOLPARM Statements for Production and DR Test

**Note** – If you use SET VOLPARM as described, you can then scratch DR test volumes from a DR test system. If you use any other method, you cannot scratch DR test volumes from a DR test system.

2. Create MGMTCLAS/STORCLAS statements for the DRTEST environment...

...using the same Management Class names as your production environment. Any Management Classes used for the DR Test will simplex data for the DR test systems. We don't want to duplex while there's only ACS01 available, so the MGMTCLAS statement looks like [CODE EXAMPLE 3-6](#).

```
MGMT NAME(CRITICAL) MIGPOL(LOCAL) IMMWAIT(0) DELSCR(YES)
```

#### CODE EXAMPLE 3-6 Create Management Class for DR Test

The MIGPOL Management Class now specifies a LOCAL Storage Class. To complete simplex of the MIGPOL Management Class, create any Storage Class referenced by MIGPOL in [CODE EXAMPLE 3-6](#) to point to ACS01 and MVC Pool DRTEST...

...as shown in [CODE EXAMPLE 3-7](#):

```
STOR NAME (LOCAL) ACS (01) MVCPOOL (MVCP1)
```

### CODE EXAMPLE 3-7 Create Storage Class LOCAL to Point to ACS01

#### 3. Set FLOAT OFF to all Production HSC hosts at the Primary Site.

This freezes the environment at the Secondary Site. If you do not do so, the CDRT CDS can be out of synchronization with the production CDS at the Primary Site.

#### 4. Ensure the following:

- Don't scratch any volumes that will be required by the DRTEST.
- Stop all CAP activity and scratch processing in the DR test ACS, and all production jobstreams using Nearline or VSM resources. This ensures that all VTVs are been migrated to MVCs and freezes the VTV states until the DR test CDS copy is created.

It is also recommended that you freeze the MVC and/or VMVC states and reduce contention for the MVC volumes by running the SLUADMIN ACTMVCN utility to create read-only control statements for all active production MVCs and/or VMVCs in the DRTEST ACS as shown in [CODE EXAMPLE 3-8](#).

```
//ACTMVCN JOB (ACCT) , 'ACTMVCN' , NOTIFY=&SYSUID
// *
//ACTMVCN1 EXEC PGM=SLUADMIN, PARM= 'MIXED '
//STEPLIB DD DSN=HLQ.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
// * NOTE: MVCMAINT READONLY (ON) STATEMENTS
//SLUSMVON DD DSN=HLQ.SLUSMVON, DISP= (MOD, CATLG, DELETE) ,
//          SPACE= (CYL, 1) , UNIT=SYSALLDA
// * NOTE: MVCMAINT READONLY (OFF) STATEMENTS
//SLUSMVOF DD DSN=HLQ.SLUSMVOF, DISP= (MOD, CATLG, DELETE) ,
//          SPACE= (CYL, 1) , UNIT=SYSALLDA
// * NOTE: THE FOLLOWING STEP SELECTS ALL "ACTIVE" MVCS
// * IN ACS 01 and STORMNGR VLE1
//SLSIN DD *
ACTMVCN ACS (01)
ACTMVCN STORMNGR (VLE1)
// *
//ACTMVCN2 EXEC PGM=SLUADMIN, PARM= 'MIXED '
//STEPLIB DD DSN=HLQ.SEALINK, DISP=SHR
//SLSPRINT DD SYSOUT=*
// * NOTE: EXEC MVCMAINT TO SET READONLY (ON)
//SLSIN DD DSN=HLQ.SLUSMVON, DISP=SHR
```

### CODE EXAMPLE 3-8 ACTMVCN Example JCL

**Note** – While completely stopping all Nearline and VSM activity is the safest way to ensure that the DR test environment is set up correctly, some customers may be unable or unwilling to completely quiesce these operations. Because the DR Test CDS is a point-in-time copy of the production CDS, the DRTEST environment has access only to VTVs created before the creation of the DR Test CDS. If volumes used in the DRTEST may have changed (because of a DISP=MOD use, or a scratch and re-use), then the DR test will not produce the expected results.

5. **Copy the MVS Catalog for the DR test site if required.**
6. **Optionally, copy the TMS database (if a TMS is used) for the DR test site.**
7. **On the production system, use the HSC CAPPREF command to set all CAPs in the DR test ACS to manual mode.**
8. **At the Production Site, run the DRTEST utility (with the PRIMEprd keyword) to prepare the production CDS for the DR test.**

For example:

```
/PRIME EXEC PGM=SLUADMIN, PARM= 'MIXED '
//STEPLIB DD DSN=hlq.SEALINK, DISP=SHR
//SLSCNTL DD DSN=hlq.DBASEPRM, DISP=SHR
//SLSCNTL2 DD DSN=hlq.DBASESEC, DISP=SHR
//SLSSTBY DD DSN=hlq.DBASETBY, DISP=SHR
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRTESTPRIME -
HOSTID (MVS1, MVS2) -
DRVTSS (VTSS01, VTSS02) -
STORMNGR (VLE1) -
DRACS (01)
```

You only need to run PRIMEprd once in your environment, no matter how many DRTEST iterations you run, unless your configuration changes. If your DR test configuration changes in any way, then you need to rerun PRIMEprd.

- 9. At the DR Test Site, run the DRTEST utility (with the CREATE keyword) against the mirrored or backup copy of the production CDS to prepare the DR Test CDS for the DR test.**

For example:

```
/CREATE EXEC PGM=SLUADMIN, PARM= 'MIXED '
//STEPLIB DD DSN=hlq.SEALINK, DISP=SHR
//SLSCNTL DD DSN=hlq.DBASEPRM, DISP=SHR
//SLSNEW1 DD DSN=hlq.DBASNEW1, DISP= (NEW, CATLG, DELETE) ,
//          UNIT=SYSDA, SPACE= (CYL, (cdssize) , , CONTIG)
//SLSNEW2 DD DSN=hlq.DBASNEW2, DISP= (NEW, CATLG, DELETE) ,
//          UNIT=SYSDA, SPACE= (CYL, (cdssize) , , CONTIG)
//SLSNEW3 DD DSN=hlq.DBASNEW1, DISP= (NEW, CATLG, DELETE) ,
//          UNIT=SYSDA, SPACE= (CYL, (cdssize) , , CONTIG)
//SLSPRINT DD SYSOUT=*
//SLSIN DD *
DRTESTCREATE -
HOSTID (MVS1, MVS2) -
DRVTSS (VTSS01, VTSS02) -
STORMNGR (VLE1) -
DRACS (01)
NOUPD
```

The CDSs and journals (if used) must be allocated via the DD statements on the utility. Note that when NOUPD is used, only the SLSCNTL DD statement is required, and it can be either the actual primary CDS, a backup, or a mirrored copy.

- 10. Start the DR test at the production site, pointing to the DRTEST MGMTCLAS/STORCLAS definitions you created in [Step 2](#).**

For example:

```
/PRIME EXEC PGM=SLUADMIN, PARM= 'MIXED '
//STEPLIB DD DSN=hlq.SEALINK, DISP=SHR
//SLSIN DD *
DRTEST START
```

- 11. Start the SMC system on the DRTEST client host(s).**
- 12. Start the SMC/HSC/VTCS system(s) on the DR test systems.**
- 13. Vary online the DR VTSSs to the DR system.**
- 14. Vary online the DR RTDs to the DR system.**

### 15. Run the tests at the DR test site.

During the DR test, the following conditions are programmatically enforced:

- The production site ACS(s) are disconnected from the DR test host(s).
- The production site VTSSs are offline to the DR test host(s).
- No floating dismounts, ejects, moves, scratch updates, audits, or scratch redistributions can occur at the DR test site.
- No floating dismounts, enters/ejects, moves, audits, or scratch redistributions on the DR test ACS can occur at the production site.
- All CAPs in the DR test ACS are in manual mode.

**Note –** You can enter volumes into the DR Test AC, but after the test is complete, you must either eject the volumes or audit the cells to synchronize the production CDS with the actual library volumes.

# Cleaning Up After a DR Test

**Note** – For information about the DRTEST command and DRTEST utility, see *ELS Command, Control Statement, and Utility Reference*. For information about CDRT messages, see *ELS Messages and Codes*.

## ▼ To remove DR test data from the VTSS buffer:

### 1. Run a VTV report.

Examine the report for any VTVs on VTSS1 that were created or modified since you began the DR test.

### 2. If your DRTEST has updated VTVs created by the production system, ensure that these VTVs are migrated and deleted from the buffer.

### 3. Run the SLUADMIN SCRATCH utility to scratch all possible VTVs in your DRTEST subpool(s).

Because you set DELSCR (YES) in your Management Class, the VTVs will be automatically deleted from the buffer when you scratch them at the conclusion of the test. If you did not use SET VOLPARM, continue with [Step 4](#). Otherwise, go to [Step 6](#).

#### **Warning –**

- If you do not use SET VOLPARM and you do not set up separate scratch pools, you are risking data loss!
- If you do not use SET VOLPARM and are using a VLE at the DR test site that is shared with a production system, then the DR VMVCs created on the VLE during the DR test must be drained from the production system to recover the space used by the DR test.

### 4. Demand migrate (and delete from the VTSS) the new VTVs you identified in [Step 1](#):

```
MIGRATE VTV (volser1, volser2, . . . volsern) DELETE (YES)
```

### 5. Optionally, migrate VTSS1 to zero to ensure that VTVs that were created and/or changed during the DR test are not reconciled to VTSS0.

**Warning – You must do this migrate to zero, otherwise production data in VTSS0 could be replaced by the test VTVs in VTSS1, which causes loss of production data!**

### 6. Stop VTCS/HSC/SMC on the DR TEST MVS system.

7. Run SLUADMIN utility with ACTMVCGN MVCMAINT READONLY(OFF) statements to reset active MVCs so that they can be used for migration as shown in **CODE EXAMPLE 3-9**.

```
//ACTMVCGN JOB (ACCT), 'ACTMVCGN', NOTIFY=&SYSUID  
//ACTMVCG1 EXEC PGM=SLUADMIN, PARM= 'MIXED '  
//STEPLIB DD DSN=hlq.SEALINK, DISP=SHR  
//SLSPRINT DD SYSOUT=*  
// * NOTE: EXEC MVCMAINT TO SET READONLY (OFF)  
//SLSIN DD DSN=hlq.SLUSMVOF, DISP=SHR
```

**CODE EXAMPLE 3-9** ACTMVCGN Example JCL (Test Cleanup)



## ▼ To resume normal operations:

1. **Stop the DR test on the PRODUCTION MVS system and reset all DR test settings in the production CDS.**

For example:

```
/STOP EXEC PGM=SLUADMIN, PARM= 'MIXED '  
//STEPLIB DD DSN=hlq.SEALINK, DISP=SHR  
//SLSPRINT DD SYSOUT=*  
//SLSIN DD *  
DRTEST STOP  
DRTEST RESET
```

2. **To resume normal production operations, do the following:**

- If desired, place CAPs in the DR test ACS in automatic mode.
- Reset FLOAT, EJECT, etc, status to desired production state.

## Operational Scenarios

This section tells how to use the DR Test software to set up the environment for, start, and stop DR testing. This section consists of the following information:

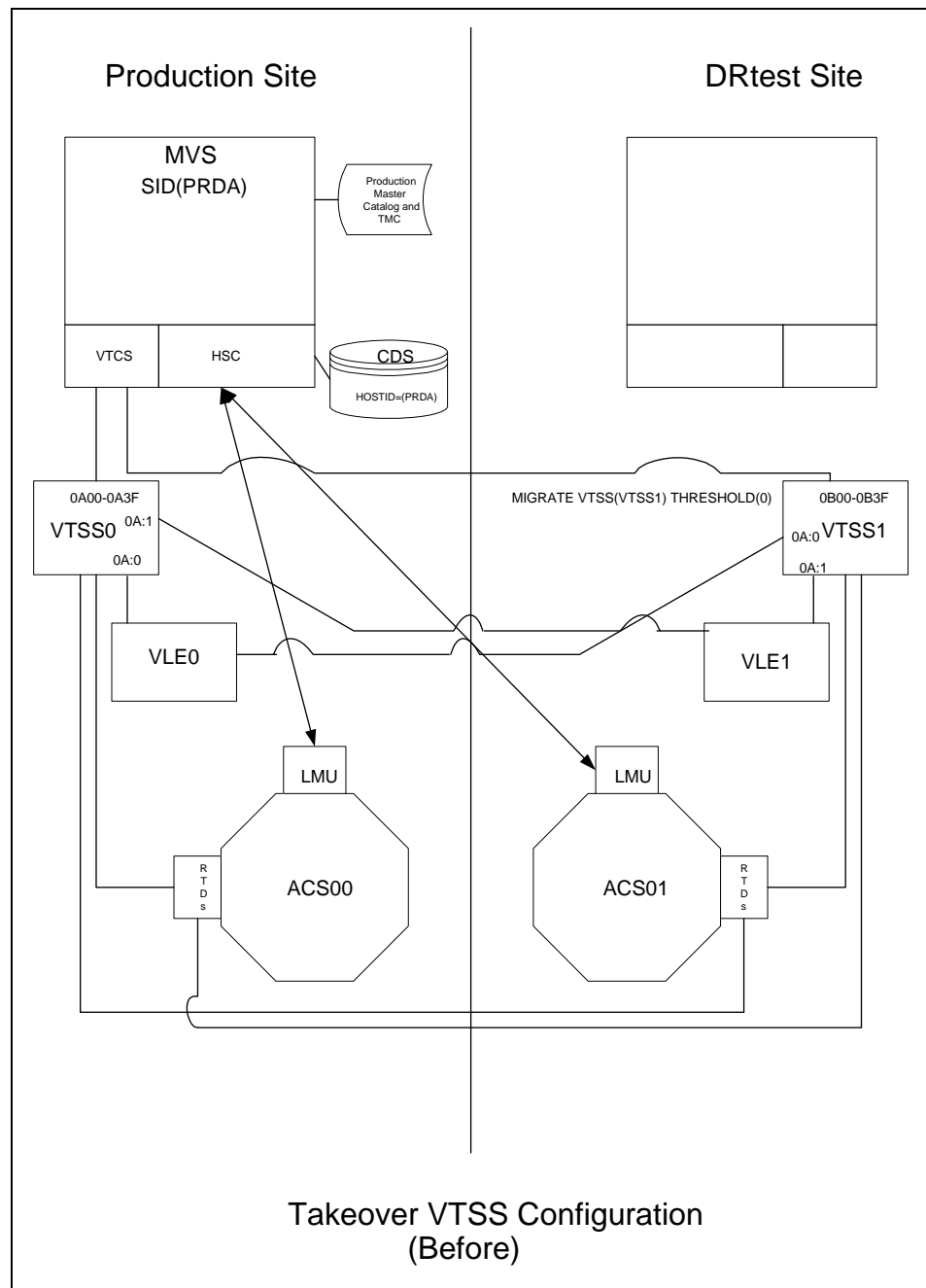
- [“Scenario 1: Production and Test Sites, ACS and VLE at Each Site” on page 42](#)
- [“Scenario 2: Production and Test Sites, VLE Only at Each Site” on page 46](#)
- [“Scenario 3: Clustered VTSSs with Production and DR Test Sites” on page 50](#)

For information about the DRTEST command and DRTEST utility, see [“DRTEST” on page 71](#). For information about CDRT messages, see [“New/Updated Messages” on page 96](#).

## Scenario 1: Production and Test Sites, ACS and VLE at Each Site

In Scenario 1, there is a single ACS at both the production and test sites, but no “spare” VTSS(s) at the test site used for testing. In normal operations, the production site writes and accesses VTVs on VTSSs at both sites, and output VTVs are always migrated immediately and duplexed, one copy to an MVC in the ACS, one copy to a VMVC in the VLE. In this configuration, you must demand migrate to zero one or more VTSSs at the test site and vary these VTSSs offline to the production system, so testing can take over the required VTSS resources. In addition, one or more LPARs at the test site function as displaced production systems, running in parallel with the actual production systems. Both ACSs and VLEs are online to the production system.

[FIGURE 3-4 on page 43](#) shows the system for Scenario 1 before running the DRTEST utility.



**FIGURE 3-4** VLE and ACS Configuration - Before Running the DRTEST Utility

FIGURE 3-5 shows the system for Scenario 1 after running the DRTEST utility.

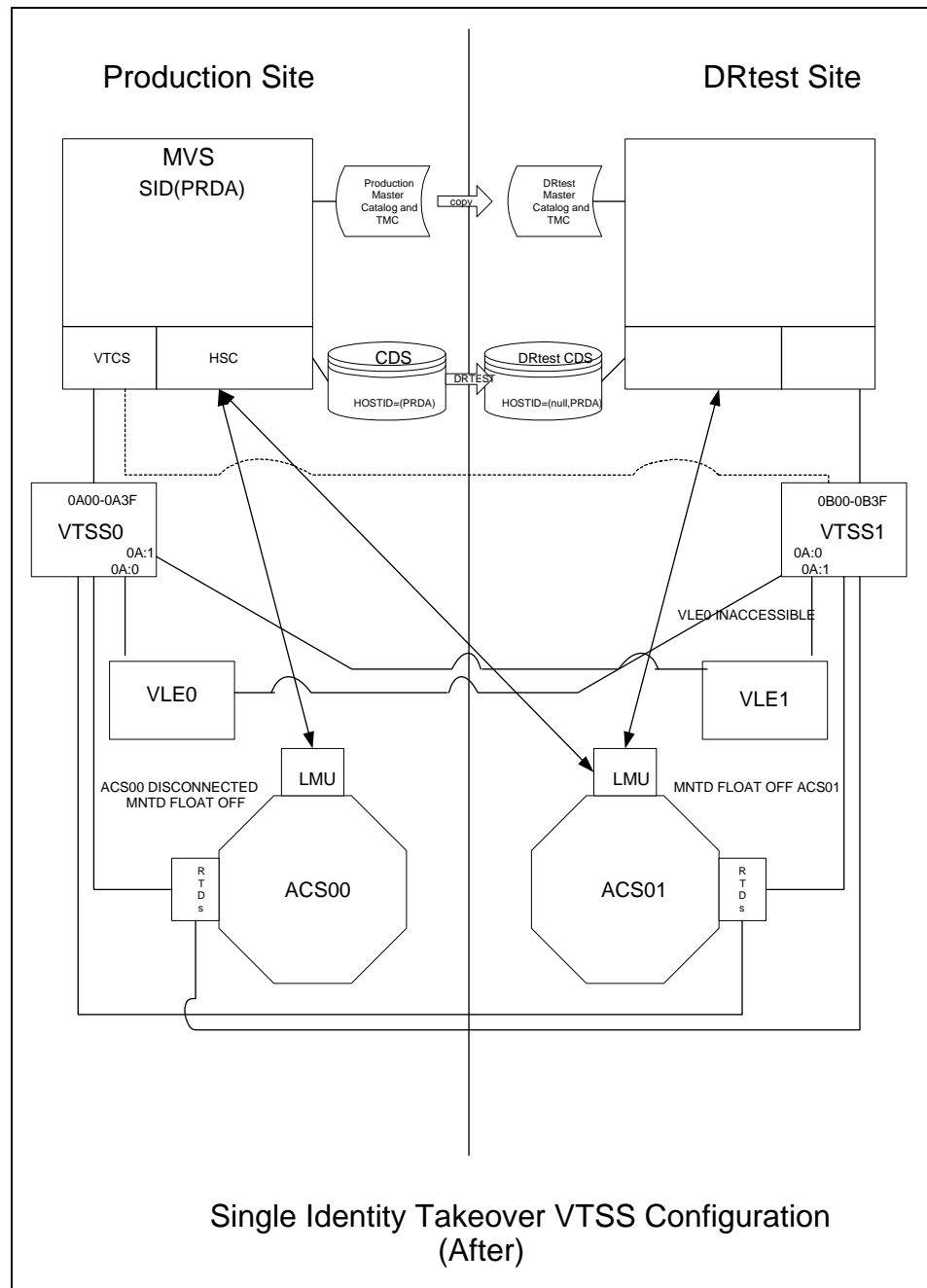


FIGURE 3-5 VLE and ACS Configuration - After Running the DRTEST Utility

## Additional Operations for Scenario 1:

- Before the test, optionally, migrate the spare VTSS(s) to zero or have your StorageTek CSE “clean” the VTSS.

Migrating to zero synchronizes the CDS with the “cleaned” state of the spare VTSS to suppress SLS6680E messages for VTV mounts. You can run VTCS VTVRPT OPTION (UNAVAIL) to ensure that all VTVs are migrated and are available to other VTSSs.

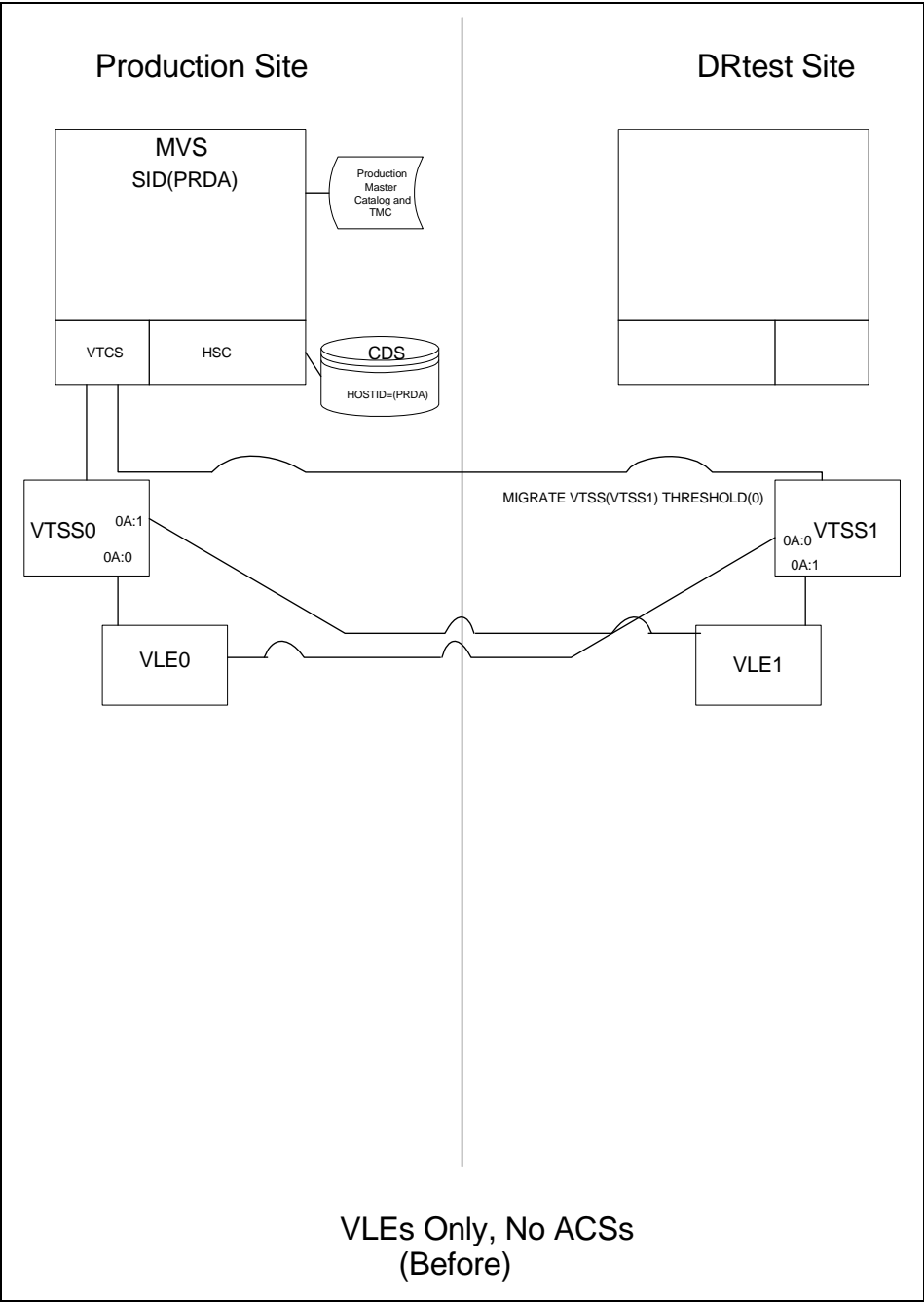
- You must add the STORMNGR parameter to the DRTEST PRIME and DRTEST CREATE SLUADMIN jobs in [“To run a DR test:” on page 34](#). The VLE that you specify on the STORMNGR parameter is the VLE at the DR site (VLE1, in this scenario).
- You do not need to do anything to the VLE; it and the ACS are shared with the production environment.
- After the test, run the procedure in [“Cleaning Up After a DR Test” on page 39](#).

## Scenario 2: Production and Test Sites, VLE Only at Each Site

In Scenario 2, there is a single VTSS with a VLE attached at each site. The VTSS at the test site is not a spare and is used by the production site during normal operations. The output VTVs are always migrated immediately and duplexed to separate VMVCs, one in each VLE.

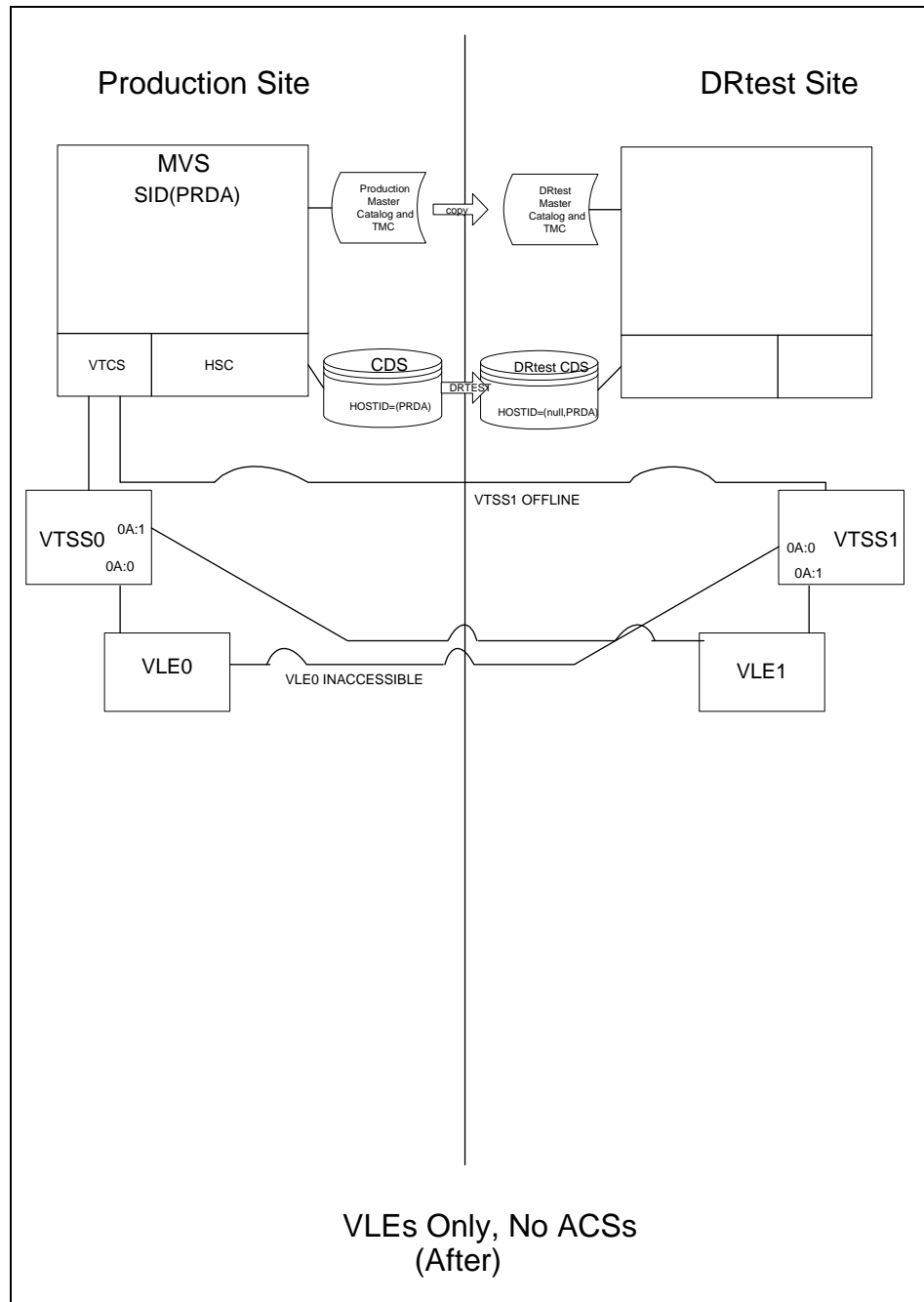
In this configuration, you must demand migrate to zero one or more VTSSs at the test site and vary these VTSSs offline to the production system, so testing can take over the required VTSS resources. In addition, one or more LPARs at the test site function as displaced production systems, running in parallel with the actual production systems. Both VLEs are online to the production system.

[FIGURE 3-6 on page 47](#) shows the system for Scenario 2 before running the DRTEST utility.



**FIGURE 3-6** VLE Only Configuration - Before Running the DRTEST Utility

FIGURE 3-7 shows the system for Scenario 2 after running the DRTEST utility.



**FIGURE 3-7** VLE Only Scenario- After Running the DRTEST Utility



## Additional Operations for Scenario 2:

- Before the test, optionally, migrate the spare VTSS(s) to zero or have your StorageTek CSE “clean” the VTSS.

Migrating to zero synchronizes the CDS with the “cleaned” state of the spare VTSS to suppress SLS6680E messages for VTV mounts. You can run VTCS VTVRPT OPTION (UNAVAIL) to ensure that all VTVs are migrated and are available to other VTSSs.

- You must add the STORMNGR parameter to the DRTEST PRIME and DRTEST CREATE SLUADMIN jobs in [“To run a DR test:” on page 34](#). The VLE that you specify on the STORMNGR parameter is the VLE at the DR site (VLE1, in this scenario).
- You do not need to do anything to the VLE; it is shared with the production environment.
- After the test, run the procedure in [“Cleaning Up After a DR Test” on page 39](#).

## Scenario 3: Clustered VTSSs with Production and DR Test Sites

As shown in [FIGURE 3-8 on page 52](#), in normal operations, Scenario 3 is a Clustered VTSS and VLE configuration used for DR, with Production and DR Test sites cross-connected to the Production and DR Test ACSs and VLEs. At the Production Site, VTSS0 is the Primary, and VTSS1 is the Secondary at the DR Test Site.

Using the secondary VTSS of a cluster as the DR test VTSS requires careful consideration in order to ensure that there are no unintended consequences during and after the DR test. Unintended consequences can include recall errors during the test and/or production data loss after the DR test is complete.

If the contents of VTSS1 change between the time that the DRTEST utility is executed to create the DRTEST CDS and the actual start of the test using the DRTEST CDS, error messages will be issued for VTVs that the DRTEST CDS indicates are resident in VTSS1 but no longer are resident. These messages are expected in this situation and do not indicate a failure. These messages can be prevented by migrating VTSS1 to zero before executing the DRTEST utility. This will synchronize the status of VTVs resident in VTSS1 with the CDS prior to the start of your DR test.

## Preparing Your VTSS Cluster for a DR Test

In this section, you're making sure any in process replications from VTSS0 to VTSS1 are complete.

### 1. Vary VTSS1 to quiesced state:

```
VARY VTSS1 QUIESCED
```

The objective here is to (gracefully) shut down replication to VTSS1 so we can use it exclusively for the DR test.

### 2. Monitor replication until it completes.

...with Display REPLICat. Here, replication is still active:

```
VTSS    HOST    QDEPTH
```

```
VTSS0   PRODUCTION    1
```

You know replication is complete when you see this:

```
VTSS    HOST    QDEPTH
```

```
VTSS0   PRODUCTION    0
```

### 3. Cross check that replication is complete by checking the CLINK status...

...with Display CLINK. Here, the CLINK is still active:

```
VTSS    CLINK    STATUS USAGE          HOST
```

```
VTSS0    7      ONLINE REPLICATING    PRODUCTION
```

```
VTSS0    8      ONLINE REPLICATING    PRODUCTION
```

You know the CLINK is no longer active when you see this:

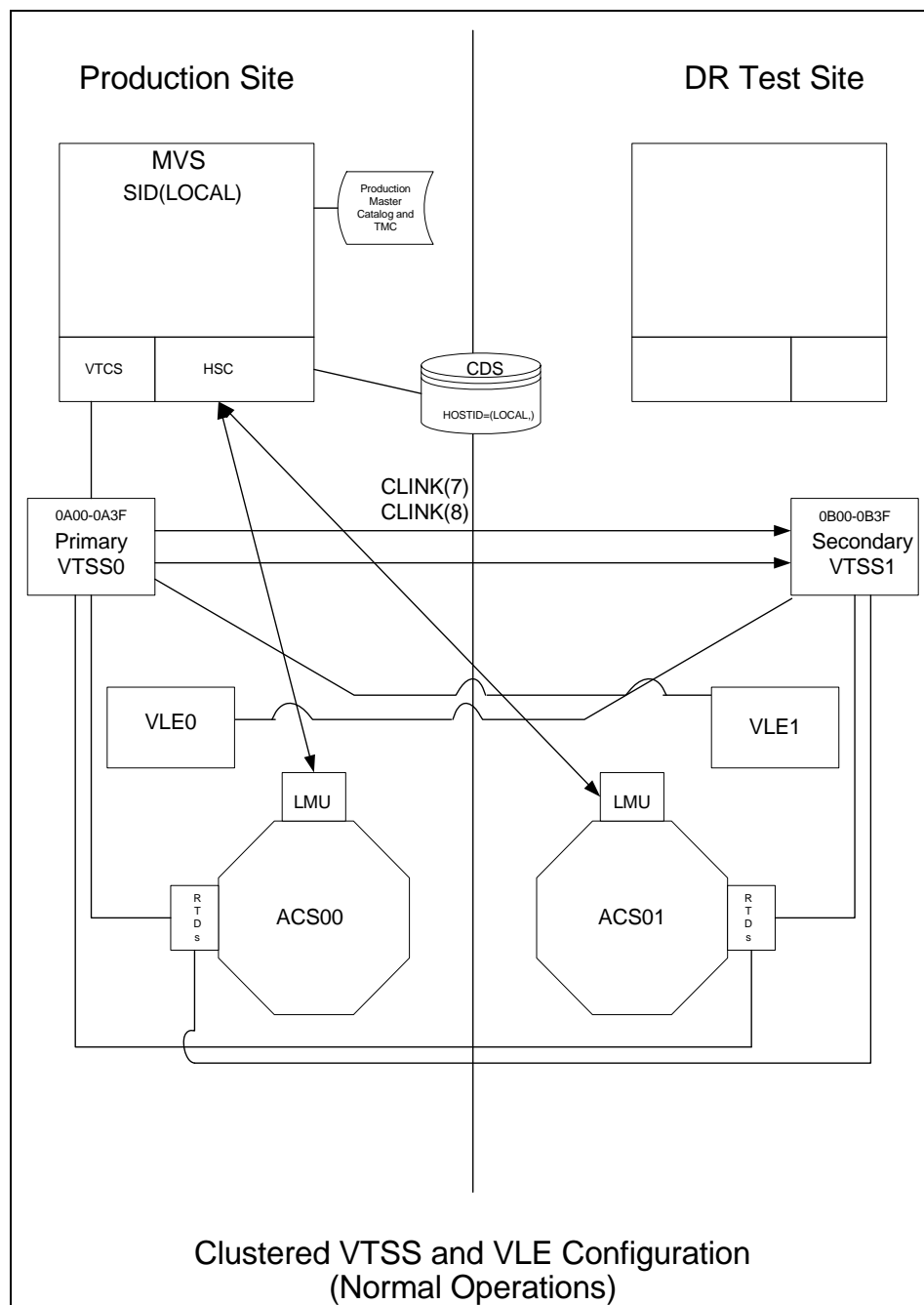
```
VTSS    CLINK    STATUS USAGE          HOST
```

```
VTSS0    7      ONLINE FREE
```

```
VTSS0    7      ONLINE FREE
```

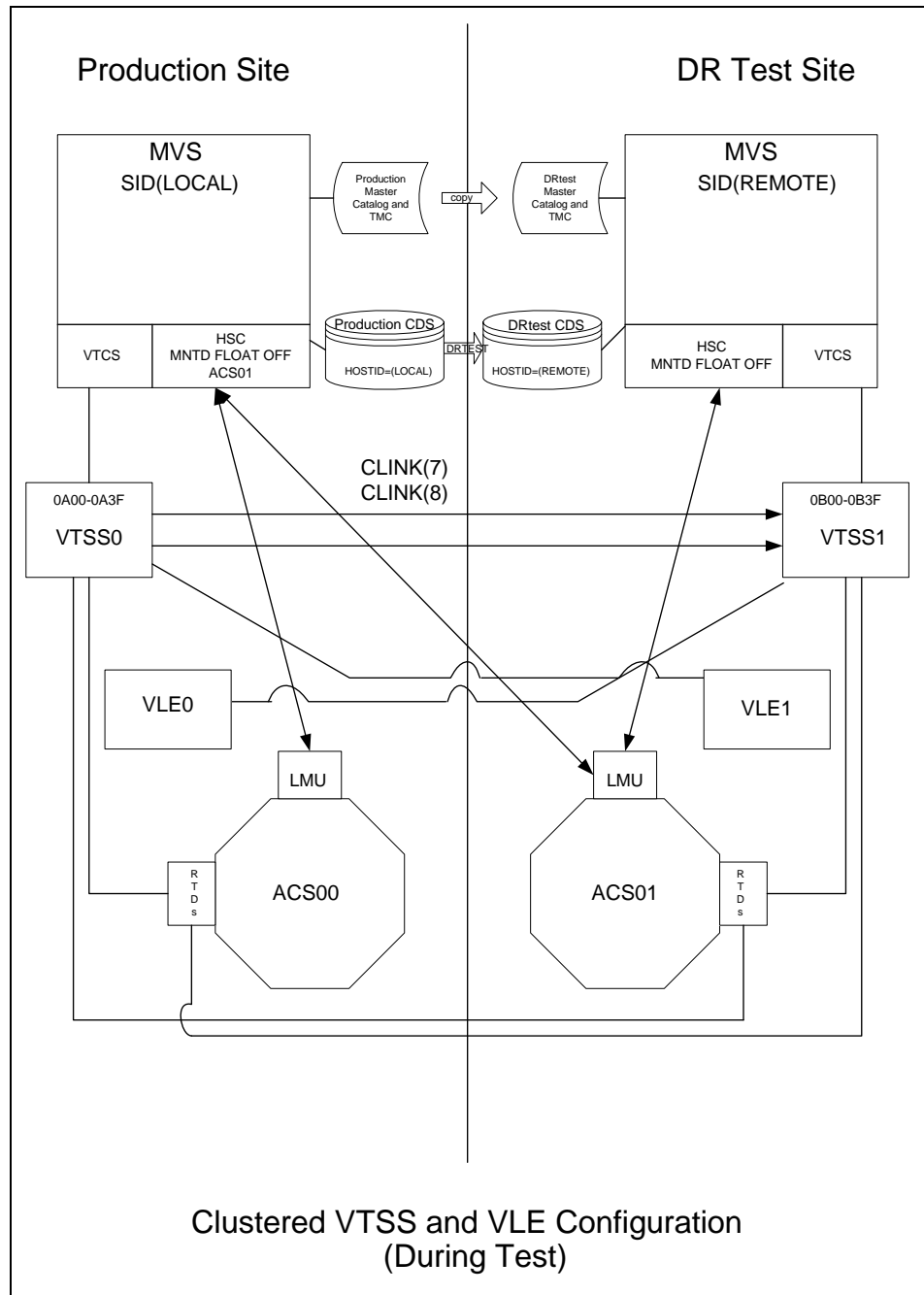
### 4. Vary VTSS1 offline:

```
VARY VTSS1 OFFLINE
```



**FIGURE 3-8** Primary/Secondary Clustered VTSS and VLE Configuration - Normal Operations

What if you wanted to use the DR Test Site for testing? [FIGURE 3-9](#) shows the system for Scenario 3 during the DR test.



**FIGURE 3-9** Primary/Secondary Clustered VTSS Configuration - During Test



---

## Command Reference

This chapter contains information about the new and updated commands and control statements used to configure VLE.

# ACTMVCGN - 7.0 and Above

The ACTMVCGN command is an optional command used in a VSM environment with the CDRT facility. ACTMVCGN produces two sets of MVCMAINT statements which are output to two files specified by the SLUSMVON and SLUSMVOF DD statements.

After ACTMVCGN executes:

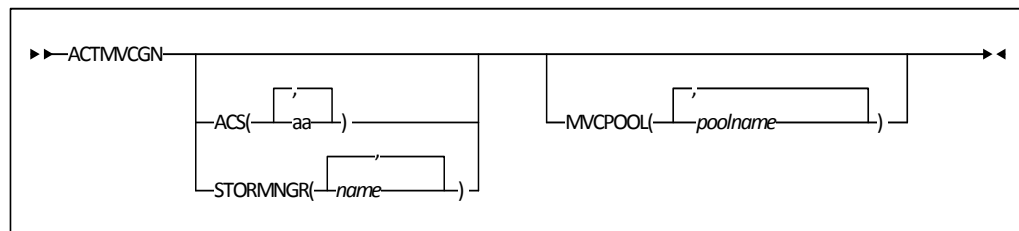
- SLUSMVON contains a set of MVCMAINT statements with the READONLY(ON) keyword.
- SLUSMVOF contains a set of MVCMAINT statements with the READONLY(OFF) keyword.

Successful execution of ACTMVCGN results in an equal number of MVCMAINT statements in both SLUSMVON and SLUSMVOF data sets.

**Note** – Refer to the *ELS Disaster Recovery and Offsite Data Management Guide* for examples of use of this utility function.

<b>Interfaces:</b>	SLUADMIN utility only
<b>Subsystem Requirements:</b>	Active HSC required only when specifying the MVCPOOL parameter

## Syntax



**FIGURE 4-10** ACTMVCGN syntax



## Parameters

### ACS

optionally, specifies an ACS that the CSV-based output is filtered against.

#### *aa*

the two-character ACS id. Blanks are not valid. To specify multiple ACSs, separate each ACS id with a comma. If ACS is not specified, the default is all ACSs.

### MVCPOOL

optionally, specifies an MVCPOOL that the CSV-based output is filtered against.

#### *poolname*

the MVCPOOL name. This name can include a maximum of 13 characters. Blanks are not valid. To specify multiple MVCPOOLS, separate each MVCPOOL name with a comma.

**Note –** If you specify the MVCPOOL parameter and the HSC subsystem is not active, the utility cannot complete and a return code of 8 is issued.

### STORMNGR

optionally, specifies a VLE that the CSV-based output is filtered against.

#### *name*

is the Subsystem Name of a VLE. This name can include a maximum of 8 characters. Blanks are not valid. To specify multiple STORMNGR names, separate each name with a comma.

## Additional JCL Requirements

In addition to the required JCL definition statements, the following definition statements apply to the ACTMVCN JCL:

### SLUSMVON

ACTMVCN output in the form of MVCMAINT utility control statements with the READONLY(ON) keyword

### SLUSMVOF

ACTMVCN output in the form of MVCMAINT utility control statements with the READONLY(OFF) keyword

## COMMtest - 6.2 and Above

The SMC COMMtest command is used to perform an end-to-end communication test between SMC and defined TapePlexes or VLEs. This command can generate a request for all paths to all TapePlexes or Storage Managers, regardless of their status. The COMMtest command does not update any error statistics or path status, but does display messages indicating the communication result.

**Note – Only** HSC TapePlexes or VLEs are eligible for the COMMtest command.

<b>Interfaces:</b>	UI: All (no XML/CSV output)
<b>Subsystem Requirements:</b>	Active SMC required, or may be input to the SMCUSIM utility

### Syntax

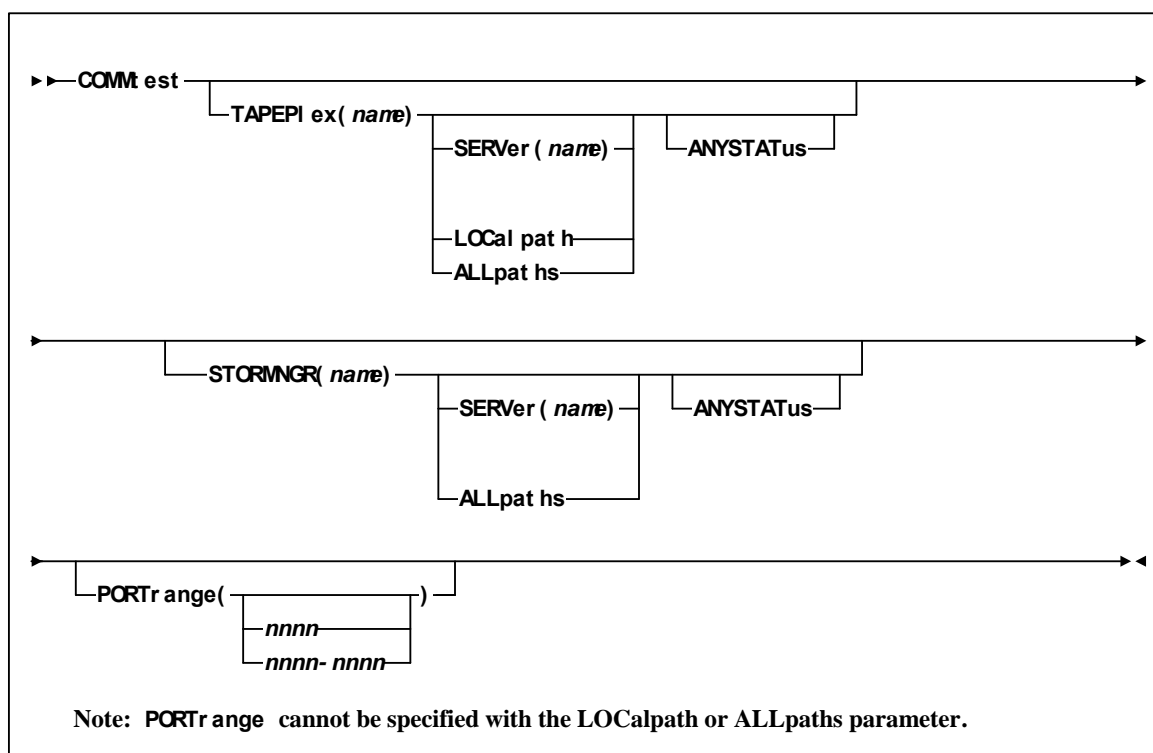


FIGURE 4-11 COMMtest syntax

## Parameters

### TAPEPlex

optionally, restricts communication to the specified TapePlex. By default, communication with all defined TapePlexes is attempted.

*name*

a TapePlex name as defined by the SMC TAPEPlex command.

### STORMNGR

optionally, restricts communication to the specified VLE. By default, communication with all defined VLEs is attempted.

*name*

a VLE name as defined by the SMC STORMNGR command.

**Note –** TAPEPlex and STORMNGR are mutually exclusive.

### SERVER

optionally, restricts communication to the specified server path. When specifying this parameter, TAPEPlex must also be specified. By default, communication with all server paths is attempted.

*name*

a server path name as defined by the SMC SERVER command.

### LOCALpath

The local server path, if any, is tested. By default, only defined remote server paths are tested.

### ALLpaths

All server paths, both local and remote, are tested.

### ANYSTATUS

Communication is attempted on all communication paths, including paths that were disabled by an operator command or by the SMC. By default, the communications test is performed on any communication path that is active, inactive, or never active. Specification of ANYSTATUS also includes disabled communication paths.

### PORTrange

optionally, specifies that communication for a remote server path be tested from the specified port or range. The specified PORTrange may be different from the TCPip PORTrange specification to allow testing of a firewall setup.

PORTrange is mutually exclusive with the LOCALpath and ALLpaths keywords.

*nnnnn or nnnn-nnnn*

the port number or port number range to be used for communication

If omitted, a port in the defined TCPip PORTrange is used. If no such port is defined, any ephemeral port is used. If a port range is specified, then communication is attempted on each port number.

# CONFIG RECLAIM - 6.2 and Above

The VTCS CONFIG RECLAIM statement controls demand and automatic MVC and VMVC space reclamation.

## Syntax

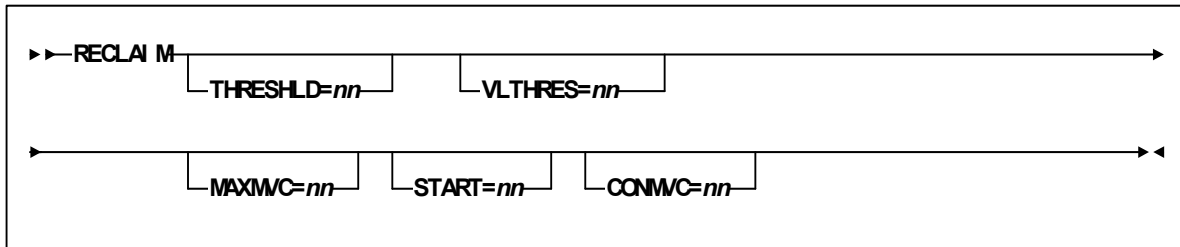


FIGURE 4-12 CONFIG RECLAIM syntax

## Parameters

THRESHLD=*m*

optionally, specifies the percentage of fragmented space that makes an MVC eligible for demand or automatic reclamation. Valid values are 4 to 98. The default is 75.

VLTHRES=*nn*

specifies the percentage of fragmented space that makes a Virtual MVC eligible for demand or automatic reclamation. Valid values are 4 to 98. The default is 30.

**Note –** Reclaim on a VMVC consists of simply deleting the expired VTV images from the VMVC. That is, no recall and re-migrate of the VTV is required. VMVC reclaim is therefore much faster than MVC reclaim, and you can set VLTHRES lower (more aggressive) than THRESHLD.

MAXMVC=*nn*

optionally, specifies the maximum number of MVCs that will be processed by a single space reclamation task. Valid values are 1 to 98. The default is 40.

For automatic space reclamation to start, the number of eligible MVCs (determined by the THRESHLD parameter) must also exceed the MAXMVC value.

START=*nn*

optionally, specifies the level at which automatic space reclamation starts for each ACS (not globally for all ACSs). Specify a percentage value, which is equal to:

$(\text{Reclaim Candidates} / \text{Reclaim Candidates} + \text{Free MVCs}) * 100$

Where:

*Reclaim Candidates*

is the number of Reclaim Candidates determined by the CONFIG RECLAIM THRESHLD parameter.

*Reclaim Candidates + Free MVCs*

equals the number of Reclaim Candidates plus the number of free MVCs. Valid values are 1 to 98. The default is 35.

CONMVC=*nn*

optionally, specifies the maximum number of MVCs that VTCS concurrently processes for both drain and reclaim.

Valid values are 1 to 99. The default is 1.

## CONFIG RTD - 6.2 and Above

The VTCS CONFIG RTD statement defines the devices (RTDs or VLEs) connected to the VTSS. This statement is required and must follow the VTSS statement that defines the VTSS to which the devices are connected. The maximum number of each device type you can connect are as follows:

- For a VSM2 or VSM3, 8 RTDs.
- For a VSM4, 16 RTDs.
- For a VSM5, 32 RTDs.
- For a VSM5, 4 VLEs.

Note that these are maximums for a single device type. If you intermix devices on a VTSS or, additionally, define CLINKs, the actual maximum for each device type is reduced.

**Note –** You must specify the RESET parameter to change device definitions if VTCS is running with a CDS level lower than V61ABOVE. For an initial RTD definition, if the RTD name displayed at the VTSS LOP or DOP is anything other than all blanks, you must also specify RESET.

## Syntax

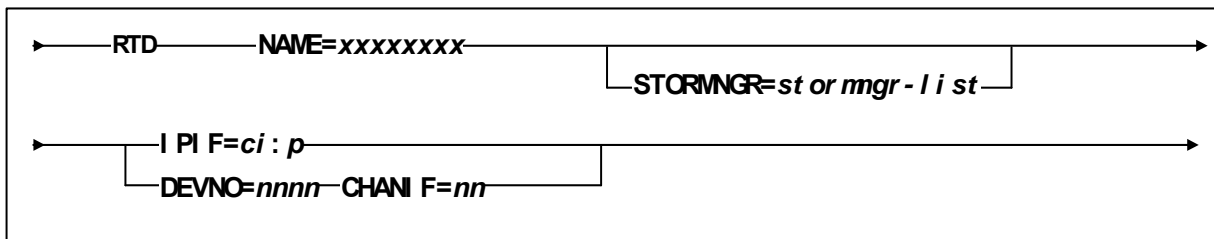


FIGURE 4-13 CONFIG RTD Syntax

## Parameters

NAME=xxxxxxx

specifies the 1 to 8 character identifier of the RTD or VLEs. This parameter is required; there is no default value

For RTDs, you set or change the RTD identifier only via the RTD NAME parameter; to do so, the RTD identifier must be all blanks as displayed at the VTSS LOP or DOP.

For VLEs, use any meaningful 1 to 8 character identifier. Unlike RTDs, VLEs, which do not use the DEVNO parameter, do not have actual MVS unit addresses. VTCS, however, generates a unit address for VLEs which you may want to use when specifying a NAME value for a VLE.

VTCS generates the unit address in the form Vxxi

where:

*xx* is derived from the VTSS config index number (order of VTSS statements in the config deck) - starting at 00.

*i* is derived from the IPIF parameter value as shown in [TABLE 4-1](#).

**TABLE 4-1** Derived i Values

IPIF Parameter Value	Derived i Value
0A:0	0
1A:0	1
0I:0	2
1I:0	3
0A:1	4
1A:1	5
0I:1	6
1I:1	7
0A:2	8
1A:2	9
0I:2	A
1I:2	B
0A:3	C
1A:3	D
0I:3	E
1I:3	F

## STORMNGR

Specifies the Subsystem Name of a VLE.

*stormngr*

a Subsystem name. This parameter does not apply to connections to RTDs (that is, when the DEVNO and CHANIF parameters are specified). This parameter only applies to connections to a VLE (that is, when the IPIF parameter is specified).

IPIF=*ci:p*

specifies the IP interface on the VTSS IFF3 card that communicates with the VLE. This value must match the value shown on the VTSS DOP IFF IP Configuration Status screen. For example, Target 0 on card IFF 0 has an interface value of 0A:0.

**Note –** The IPIF parameter is only valid for specifying a connection to a VLE, it is not valid for RTD connections. Similarly, the DEVNO and CHANIF parameters are only valid for specifying a connection to an RTD. They are not valid for specifying a connection to a VLE.

DEVNO=*nnnn*

specifies the unit address of the RTD.

This parameter is required with CHANIF; there is no default value.

CHANIF=*ci* or *ci:p*

specifies the channel interface on the VTSS that communicates with the RTD where:

- *c* is the VTSS Storage Cluster number (0 or 1).
- *i* is the interface number (A, C, E, G, I, K, M, or O)
- *p* is the device number on the interface (0, 1, 2, or 3).

This value must match the Nearlink channel interface defined at the VTSS by your StorageTek hardware representative at VTSS installation and configuration. This parameter is required with DEVNO; there is no default value. Note the following:

- Regardless of whether the Maximum 32 RTDs feature is enabled, if you do not have greater than 16 RTDs attached to a VTSS, you can use the “old” addressing scheme (CHANIF=*ci*).
- If, however, the Maximum 32 RTDs feature is enabled and you have greater than 16 RTDs attached to a VTSS, you must use the “new” addressing scheme (CHANIF=*ci:p*).

**Note –** The DEVNO and CHANIF parameters are only valid for specifying RTD connections, they are not valid for a connection to a VLE.

**Caution –** For RTDs, driving mount and dismount commands to the device is version dependant. If the remote HSC server is running V6.2, then it is dependant upon the local SMC trapping the MVS message and forwarding the mount/dismount across to the HSC server as if it was a job. If the remote HSC server is running V7 or above, then this is automatically detected and VTCS directs a mount/dismount request directly to the remote HSC server. It is important that the SMC parameters only direct commands to servers of one of these types.



# CONFIG STORMNGR - 7.0 and Above

The VTCS CONFIG STORMNGR statement defines a VLE to VTCS.

## Note –

- Do not specify this statement when down-level hosts are active. Once CONFIG STORMNGR is specified, down-level hosts no longer function.
- This statement requires CDS level V62ABOVE or higher.
- This statement is an alternative to [“CONFIG TAPEPLEX - 6.2 and Above” on page 67](#). That is, you can either specify a list of VLEs on CONFIG TAPEPLEX or each VLE individually via CONFIG STORMNGR. If you code CONFIG STORMNGR statements, they must immediately follow the CONFIG TAPEPLEX statement to which the CONFIG STORMNGR statements apply.

**Note, however,** that if you want to define a VLE multi-node system, you must use the CONFIG STORMNGR statement to do so.

## Syntax - 7.0

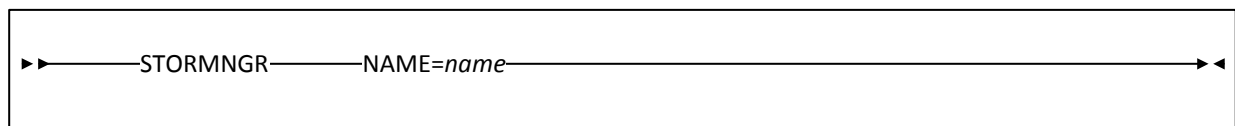


FIGURE 4-14 CONFIG STORMNGR Syntax - 7.0

## Syntax - 7.0 and 7.1

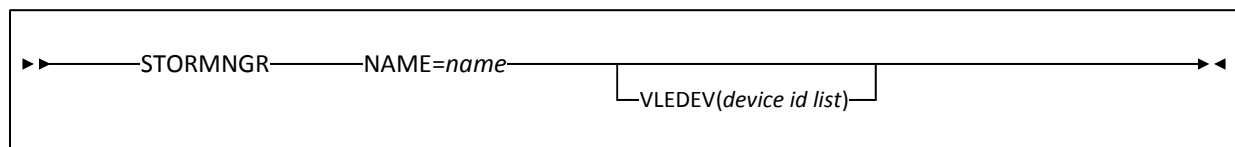


FIGURE 4-15 CONFIG STORMNGR Syntax - 7.1

## Parameters

STORMNGR

specifies Subsystem Name of the VLE attached to the preceding TAPEPLEX statement.

NAME=*name*

the Subsystem name.

VLEDEV

allows you to specify the logical device IDs of a VLE. This parameter is optional and requires CONFIG CDSLEVEL=V71ABOVE (CDS Level H).

*device id list*

specifies a list or range of device IDs whose format is an 'S' prefix followed by three hexadecimal characters. These IDs are similar to MVS device addresses but do not overlap with the MVS name space. You can specify up to 96 device IDs per VLE , which defines each VLE with 96 emulated devices, which allows VTCS to schedule up to 96 processes on each VLE.

# CONFIG TAPEPLEX - 6.2 and Above

The VTCS CONFIG TAPEPLEX statement defines values to VTCS for Cross-TapePlex Replication (CTR) (7.0 and 7.1 only) or for migration to a VLE (6.2 and above).

## Note –

- Do not specify this statement when down-level hosts are active. Once CONFIG TAPEPLEX is specified, down-level hosts no longer function.
- This statement requires CDS level V61ABOVE or higher.

## Syntax - 7.0 and 7.1

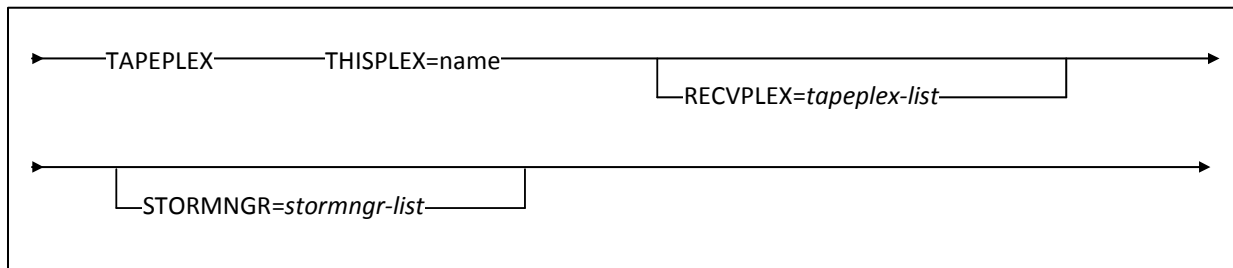


FIGURE 4-16 CONFIG TAPEPLEX Syntax - 7.0 and 7.1

## Syntax - 6.2

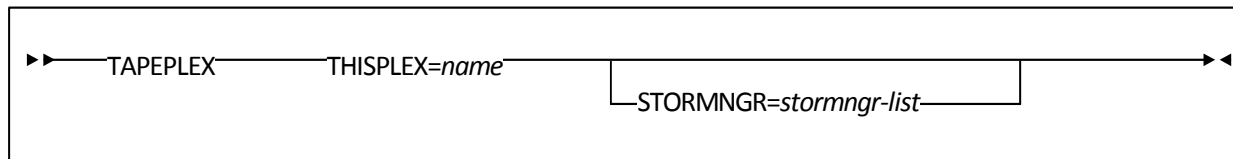


FIGURE 4-17 CONFIG TAPEPLEX Syntax - 6.2

# Display STORMNGR

## Interfaces:

Console or PARMLIB

UII: No

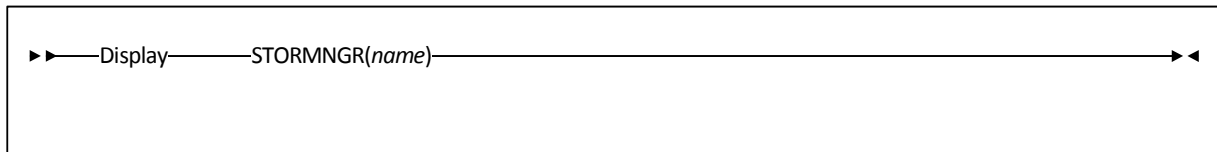
## Subsystem Requirements:

Active HSC/VTCS

## Description

Display STORMNGR displays the status of an external storage manager and the paths defined to it from the VTSSs.

## Syntax



**FIGURE 4-18** Display STORMNGR syntax

## Parameters

STORMNGR

specifies the external storage manager to be displayed

*name*

the storage manager name

## Output

Name	Type	Version	Status
VLELIB0	VLE	1.1	Online
Features: VLE to VLE			
Device	Name	VTSS	CHANIF
V000	SS16IP0	DVTSS16	08 I0A:0
V002	SS16IP1	DVTSS16	09 I0I:0
V001	SS16IP2	DVTSS16	0A I1A:0
V003	SS16IP3	DVTSS16	0B I1I:0

**CODE EXAMPLE 4-10** Example Display STORMNGR output

## Fields

### Name

the name of the external storage manager

### Type

the type of storage manager

- HSC - An HSC in another TapePlex
- VLIB - A VLE

### Version:

the software version currently executing on the storage manager

### Features:

a list of significant features installed or supported by the storage manager

### Status:

the current status of the storage manager

### Device

the destination device for the path. This column is blank in the connection to the storage manager is for autonomous device data transfers.

### Name

the name allocated to the path within the storage manager. This column is blank if the storage manager is just for servicing mounts and dismounts to the device. If this is for a CLINK, then this column displays the target VTSS name.

### VTSS

the name of a VTSS which has this connection to this storage manager. This column is blank if the device is driven directly by the storage manager.

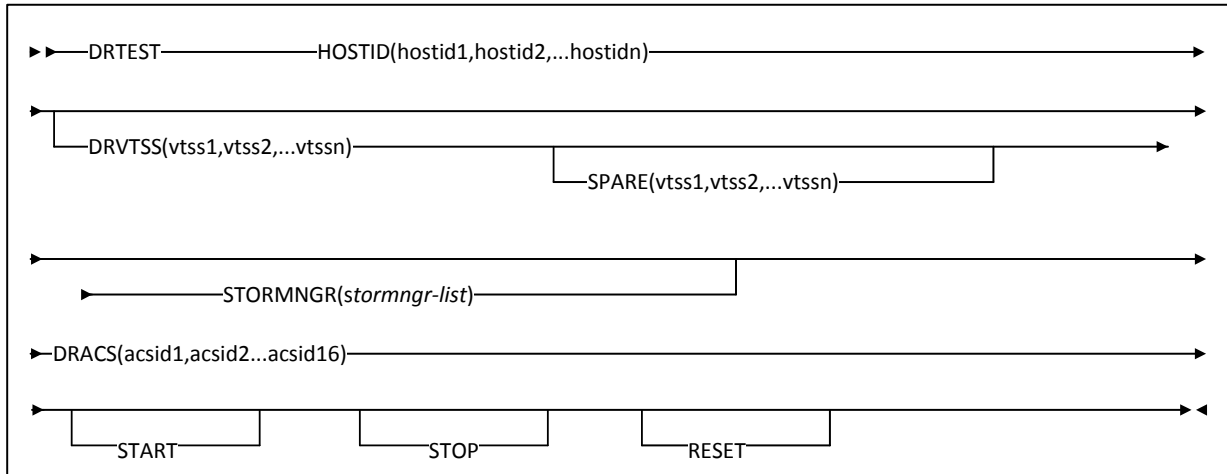
## CHANIF

the logical device Id assigned to the path and the CHANIF or IPIF value that was specified for the path. This column is blank if the path to the storage manager is not directly from a VTSS.

# DRTEST

The CDRT DRTEST utility (6.2) or command (7.0) sets up the environment for DR testing and, optionally, starts and stops the test.

## Syntax



**FIGURE 4-19** DRTEST utility syntax

## Parameters

HOSTID (*hostid1,hostid2,...hostidn*)

Specifies the HOSTIDs replicated in the test CDS in available host-id positions. The combination of existing production HOSTIDs and those identified in this parameter for replication cannot exceed the 16 host limit. If an additional host-id (not used in production) needs to be added for DR test purposes alone, this additional host-id will use 2 of the 16 host-ids in the limit.

See [“PRIMING the Production CDS” on page 74](#) for parameter use details.

DRVTSS (*vtss1,vtss2,...vtssn*)

Optionally, specifies the VTSS names available to the DR test site. Unless you specify these VTSSs as spares, the VTSSs are offline to the production site and online to the DR test site when the DR test starts.

SPARE (*vtss1,vtss2,...vtssn*)

Specifies that the DR test VTSS(s) are spares and that identically named VTSS(s) exist and are online at the production site when the DRTEST utility is run and during the DR test.

See [“PRIMING the Production CDS” on page 74](#) for use details.

DRACS (*acsid1...acsid16*)

Specifies one or more ACSs available to the DR test site. Multiple ACS IDs must be specified as a list, and not as a range of ACS IDs. The ACSs are online to the DR test site but can remain connected (and usually are) to the production site when the DR test starts. CDRT Version 6.2 and above supports a maximum of 16 DR ACSs.

See [“PRIMING the Production CDS” on page 74](#) for use details.

STORMNGR

Optionally, specifies the Subsystem Names of VLEs attached to the DR test site TapePlex.

*stormngr-list*

the list of Subsystem names.

START

Starts the DR test on the production site.

Example Control Cards:

DRTEST START

START Requirements:

- HSC/VTCS must be active on the executing system.
- DR test should not be active. This refers to the production CDS status.
- SLSNEW DD statement(s) identifying the DRTEST CDS(s) are required.

START Options:

- SLSCNTL DD statement(s) identifying the production CDS are optional, but must match the active HSC/VTCS if used.



**Note** – START can be specified with other parameters (HOSTID, DRVTSS, DRACS, STORMNGR, and SPARE) when creating the DR test CDS and starting the DR test at the same time.

## STOP

Stops the DR test on the production site.

Example Control Cards:

DRTEST STOP

STOP Requirements:

- HSC/VTCS must be active on the executing system.
- SLSNEW DD statement(s) identifying the DRTEST CDS(s) are required.
- All test systems at the DR test site must be terminated.
- No other parameters are coded on the same control card with STOP.

STOP Options:

- SLSCNTL DD statement(s) identifying the production CDS are optional, but must match the active HSC/VTCS if used.

## RESET

Removes all DR test settings in the production CDS.

Example Control Cards:

DRTEST RESET

RESET Requirements:

- HSC/VTCS can be either active on the executing system or not.
- DR test must not be active. This refers to the production CDS status.
- SLSNEW DD statement(s) identifying the DRTEST CDS(s) are optional.
- No other parameters are coded on the same control card with RESET.

RESET Options:

- SLSCNTL DD statement(s) identifying the production CDS are optional, but must match the active HSC/VTCS if used.

## Usage

Use the DRTEST utility to set up the environment for DR testing and, optionally, to start and stop the test. You use the SWUADMIN program to run the DRTEST utility. You can also use the HSC DRTEST command to start or stop testing.

Starting the DR test sets a state in the production environment where certain HSC and VTCS system functions are inhibited. Before starting the test, you need to quiesce some ongoing HSC and/or VTCS processes. Starting systems with the DR test CDS also inhibits certain HSC and VTCS functions and assures the proper operating state for the DR test.

### ▼ Running DRTEST STOP and RESET in the Same Job

Example Control Cards:

```
DRTEST STOP
DRTEST RESET
```

DRTEST STOP and DRTEST RESET can be run in the same job with two control cards per the example above. Note that STOP must run before RESET. This combination requires:

- HSC/VTCS must be active.
- DR test should be active.
- SLSCNTL DD statement(s) identifying the production CDS are not allowed.
- SLSNEW DD statement(s) identifying the DRTEST CDS(s) are required.

### ▼ PRIMING the Production CDS

Example Control Cards:

```
DRTEST HOSTID (ZIPF, ZIPG) -
DRVTSS (VTSSW) -
DRACS (00) -
STORMNGR (VLE1)
```

The HOSTID, DRVTSS, DRACS, STORMNGR and optionally the SPARE parameters are mutually inclusive. Running the DRTEST utility with a combination of these parameters is called PRIMING the production CDS. Until this PRIMING is done, DRTEST START cannot be run successfully. PRIMING requires:

- HSC/VTCS can be either active or not.
- DR test must not be active during PRIMING.
- SLSCNTL DD statement(s) identifying the production CDS are optional, but must match the active HSC/VTCS if used.
- SLSNEW DD statement(s) identifying the DRTEST CDS(s) are required.

## ▼ PRIMING the Production CDS and Running DRTEST START in the Same Job

Example Control Cards:

```
DRTEST HOSTID (ZIPF, ZIPG) -  
DRVTSS (VTSSW) -  
DRACS (00) -  
STORMNGR (VLE1) -  
START
```

The Production CDS can be primed and DR test started within the same job. This combination requires:

- HSC/VTCS must be active.
- DR test must not be active during priming.
- SLSCNTL DD statement(s) identifying the production CDS are not allowed.
- SLSNEW DD statement(s) identifying the DRTEST CDS(s) are required.

## JCL Requirements

STEPLIB

specifies the link library that contains the CDRT, HSC, and VTCS modules.

**Note –** The SWUADMIN program determines the HSC Primary CDS for both the production and test CDS as follows:

- If your JCL does not specify the CDS and HSC is initialized, SWUADMIN queries HSC for the Primary CDS and SWUADMIN uses that CDS. The JCL examples in this chapter show this method.
- If your JCL specifies all defined copies of the CDS, SWUADMIN queries these copies and uses the correct copy.

SLSCNTL

specifies the current primary copy of the production HSC CDS.

SLSCNTL2

specifies the current secondary copy of the production HSC CDS.

SLSSTBY

specifies the current standby copy of the production HSC CDS.

SLSNEW1

specifies the new primary copy of the test HSC CDS.

SLSNEW2

specifies the new secondary copy of the test HSC CDS.

SLSNEW3

specifies the new standby copy of the test HSC CDS.

SLSJRN00 - SLSJRNnn

DDNAMEs for the DR test journal files, which are only valid if the current CDS defines journaling. There are two files per HOSTID: SLSJRN00 and SLSJRN01 for *hostid1*, SLSJRN02 and SLSJRN03 for *hostid2*, and so forth).

SLSSTATN

DDNAME for LMU station address changes for HOSTIDs. This file is optional, and if not supplied the same station addresses are used in the DR test CDS for the HOSTIDs as in the existing CDS.

Each entry follows the HSC SET SLISTATN utility format; for more information, see HSC/MVS System Programmer's Guide. [CODE EXAMPLE 4-11](#) shows an example of the SLSSTATN file.

```
SET SLISTATN(00E7,00E8),FORACS(01),FORHOST(PRI1)
SET SLISTATN(00E7,00E8),FORACS(01),FORHOST(PRI2)
SET SLISTATN(00E7,00E8),FORACS(01),FORHOST(PRI3)
SET SLISTATN(00E7,00E8),FORACS(01),FORHOST(PRI4)
SET SLISTATN(00E7,00E8),FORACS(01),FORHOST(PRI5)
```

#### **CODE EXAMPLE 4-11** SLSSTATN File Example

SLSVTSS

DDNAME for VSM changes on the DR test CDS. This file is optional and is only used if the DR test configuration includes VSM elements. Typically, this file is used to change RTD hardware connection definitions in the DR test CDS but may be used to modify any VSM definitions on the DR test CDS because the file invokes the VTCS CONFIG RESET utility. Note that improper use of CONFIG RESET against the DR test CDS may render the DR test environment inoperable!

Each entry follows the format of the VTSS, VTD, RTD, and HOST statements of the VTCS CONFIG utility; for more information, see VTCS Installation, Configuration, and Administration Guide. [CODE EXAMPLE 4-12](#) shows an example of the SLSVTSS file.

```
CONFIG RESET
VTSSNAME=VTSS01  LOW=70 HIGH=80  MAXMIG=1  MINMIG=1
RETAIN=10
    RTDNAME=VTS18800 DEVNO=8800 CHANIF=0A
    RTDNAME=VTS18801 DEVNO=8801 CHANIF=0I
    RTDNAME=VTS18802 DEVNO=8802 CHANIF=1A
    RTDNAME=VTS18803 DEVNO=8803 CHANIF=1I
    HOST NAME=MVS1
    VTDLOW=8900 HIGH=893F
VTSSNAME=VTSS02  LOW=70 HIGH=80  MAXMIG=8 MINMIG=8
RETAIN=10
    RTDNAME=VTS28805 DEVNO=8805 CHANIF=0A
    RTDNAME=VTS28806 DEVNO=8806 CHANIF=0E
    RTDNAME=VTS28807 DEVNO=8807 CHANIF=0I
    RTDNAME=VTS28808 DEVNO=8808 CHANIF=0M
    RTDNAME=VTS28809 DEVNO=8809 CHANIF=1A
    RTDNAME=VTS2880A DEVNO=880A CHANIF=1E
    RTDNAME=VTS2880B DEVNO=880B CHANIF=1I
    RTDNAME=VTS2880C DEVNO=880C CHANIF=1M
    HOST NAME=MVS2
    VTD      LOW=9900 HIGH=993F
```

#### **CODE EXAMPLE 4-12** SLSVTSS File Example

SLSOUT

specifies the destination for the SWUADMIN processing messages.

SLSIN

specifies the input to the SWUADMIN program (DRTEST utility name and parameters).

## Route - 6.2 and Above

The SMC Route command is used to request routing of transactions from SMC to a defined TapePlex or VLE. **Only** the following are supported:

- VTCS commands
  - n HSC commands with UUI support, with the exception of VOLRPT.

**Note –** The HSC MOUNT, DISMOUNT, ENTER and MOVE commands can now be issued from SMC via the ROUTE command. Refer to the *ELS Programming Reference* for information about supported commands.

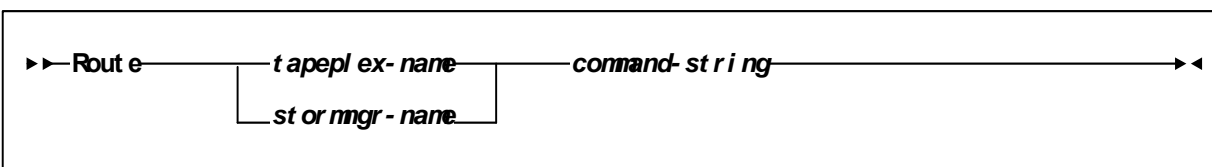
---

**Interfaces:** UUI: All (no XML/CSV output)

**Subsystem Requirements:** Active SMC required, or may be input to the SMCUSIM utility

---

## Syntax



**FIGURE 4-20** Route syntax

## Parameters

### *tapeplex-name*

specifies a TapePlex name as defined on an SMC TAPEPLEX command. SMC routes the request to the specified TapePlex using the currently active TapePlex path. Commands may be sent to either a local or remote HSC TapePlex. This command is **not** supported for MVS/CSC TapePlexes.

### *stormngr-name*

specifies a VLE name as defined on an SMC STORMNGR command. SMC routes the request to the specified VLE using the currently active VLE path. Commands may be sent to either a local or remote HSC TapePlex or VLE. This command is **not** supported for MVS/CSC TapePlexes.

### *command-string*

specifies the command string to be routed to the requested TapePlex or VLE. Note that VTCS commands should not be prefixed with VT; the HSC UUI interface routes VTCS commands to the correct functional processor without the VT prefix. The command string must be a command supported by the HSC UUI or any VTCS command (except DISPLAY MSG and DISPLAY CMD).

## SERVer - 6.2 and Above

The SMC SERVer command defines a named path to a remote library server; an HSC subsystem executing on another host. The SERVer command describes the communication path to the HTTP server.

### Note –

- Before a SERVer is defined, the TapePlex that it references must be defined using a TAPEPLEX command. The TapePlex name associated with a SERVer **cannot** be changed.
- Similarly, before a SERVer is defined, the VLE that it references must be defined using a STORMNGR command. The VLE name associated with a SERVer **cannot** be changed. See “[STORMNGR - 6.2 and Above](#)” on page 88 for more information.
- You **cannot** define a server if the corresponding TAPEPLEX defines a LOCSUBSYS for an MVS/CSC subsystem. For example, the following is **not** valid (assuming the subsystem CSC1 is an MVS/CSC system).

```
TAPEPLEX NAME (LIB1) LOCSUBSYS (CSC1)
SERVER NAME (REM1) TAPEPLEX (LIB1)
```

---

**Interfaces:** UII: All (no XML/CSV output)

**Subsystem Requirements:** Active SMC required, or may be input to the SMCUSIM utility

---

## Syntax

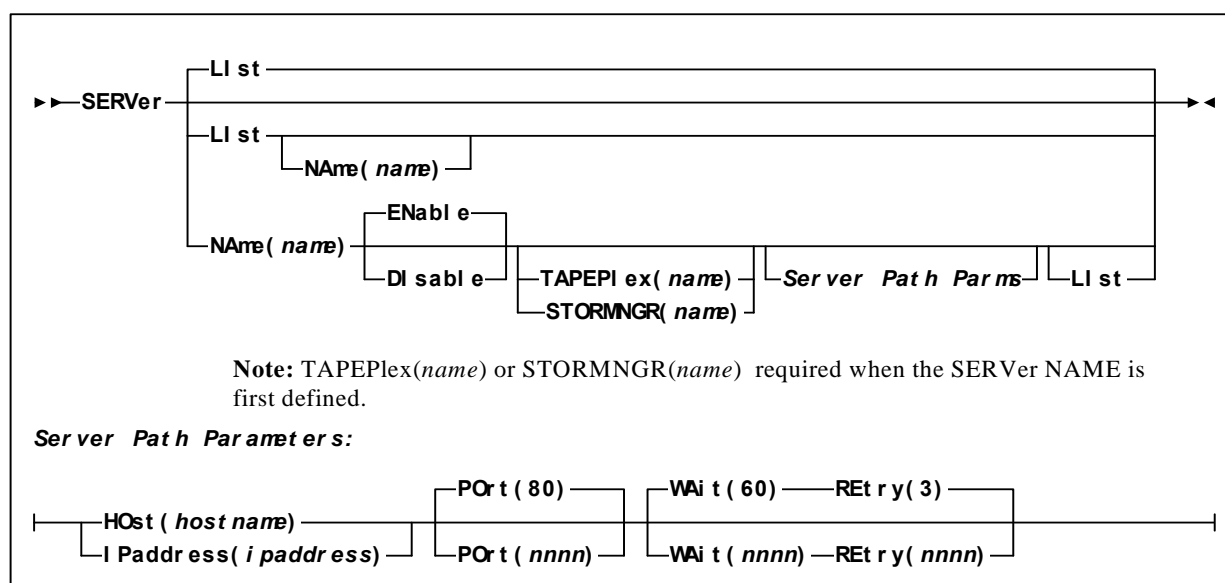


FIGURE 4-21 SERVer syntax



# Parameters

## List

optionally, displays status information for TapePlex server paths

- List is the default when no parameters are specified on the SERVer command. In this case, all library server paths are listed.
- List may be specified with other parameters. When specified with parameters other than NAmE, the List is generated **after** the other parameters are processed.

## NAmE

optionally, specifies a server path for which status is displayed

*name*

the server path name

## NAmE

optionally, specifies the server name to be defined or modified.

*name*

the server path name. This name is reported in any communications error message when the error did not occur while communicating with an HSC on the same host. The following rules apply:

- The value must be between 1 and 8 characters in length.
- The first character must be either an alpha character or digit.
- The last character must be either an alpha character or digit.
- Any character between the first and last must be either an alpha character, digit, or hyphen.

**Note** – If you are using the StorageTek HTTP server, there can be multiple NAMEd server paths for a single TapePlex configuration. Refer to the publication Configuring and Managing SMC for more information.

## ENable

optionally, enables the specified server path to be selected for allocation or mount requests

## Disable

optionally, disables the specified server path. If this is the only path to the TapePlex, the TapePlex is unavailable for allocation or mount requests.

## TAPEPlex

optionally, specifies the TapePlex name associated with an actual ACS hardware configuration as defined on an SMC TAPEPlex command. The TAPEPlex parameter **must** be specified when a new server is defined (i.e., added).

*name*

a TapePlex name as specified on the TAPEPlex command. This name is reported in any TapePlex error message.

## STORMNGR

optionally, specifies a VLE name as defined on an SMC STORMNGR command. The STORMNGR parameter **must** be specified when a new server is defined (i.e., added).

*name*

a VLE name as defined on an SMC STORMNGR command. This name is reported in any VLE error message.

**Note –** TAPEPlex and STORMNGR are mutually exclusive.

Host

optionally, specifies the IP resolver host name on which the remote HSC subsystem resides

*hostname*

the name of the remote host. The following rules apply:

- The value must be between 1 and 8 characters in length.
- The first character must be either an alpha character or digit.
- The last character must be either an alpha character or digit.
- Any character between the first and last must be either an alpha character, digit, hyphen, or dot.

**Note –** Host and IPaddress are mutually exclusive.

IPaddress

optionally, specifies the subsystem IP address

*ipaddress*

the IP address

**Note –** IPaddress and Host are mutually exclusive.

Port

optionally, specifies the server port

*nnnn*

the server port. The default is 80. For SMC communication with a VLE, the SERVER PORT parameter is always 60000.

Wait

optionally, specifies the maximum wait time for any single request made over the network before the SMC assumes that a communication or server problem exists

*nnnn*

the wait time in seconds. The default is 60.

**Note –**

- The default for a mount or dismount request is 10 minutes (600 seconds) or more if the specified WAIT time is greater than 600 seconds.
- If your HSC CDS backup job runs longer than the specified wait time, set your wait time to match the normal execution time of your CDS backup job.

REtry

optionally, specifies the number of retry attempts for any single request before the task is allowed to resume, and a failure recorded

*nnnn*

the number of retries. The default is 3.

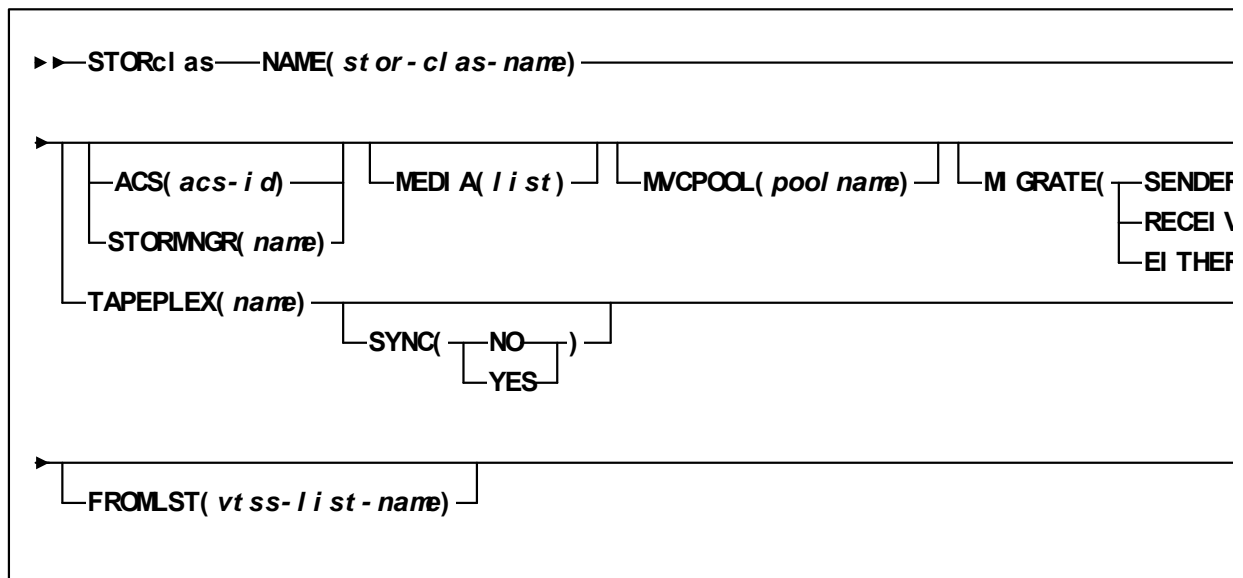
## STORclas Control Statement - 6.2 and Above

The HSC `STORclass` control statement defines a VSM Storage Class. It is loaded by the `MGMTDEF` command.

This statement can specify whether a VTV copy is to be written to:

- An MVC (with required attributes).
- The name of a remote TapePlex to which a copy of the VTV is exported.
- The Subsystem Name of a VLE.

## Syntax



**FIGURE 4-22** STORclas syntax

## Parameters

### NAME

specifies the name of the Storage Class.

*stor-clas-name*

the Storage Class name. This name must be 1 to 8 alphanumeric characters beginning with an alpha character and must follow SMS naming conventions.

### ACS

optionally, specifies the ACSs from which RTDs and MVCs are selected.

*acs-id*

Specifies the ACS ID. An ACS ID has a hexadecimal value from 00 through FE.

### STORMNGR

Optionally, specifies the Subsystem Name of a VLE. If the specified Subsystem does not exist, then any migrations fail and the VTVs are “stuck” in their source VTSS.

*name*

a Subsystem name.

### MEDIA

optionally, specifies a preference list of MVC media types. This list supersedes the default media selection list. Refer to *Configuring HSC and VTCS* for more information.

*list*

preference list of media types

### MVCPPOOL

optionally, specifies the Named MVC Pool from which volumes are selected. If you do not specify an MVC Pool name, the volumes are selected from the default pool (DEFAULTPOOL).

*poolname*

the name of an MVC Pool that you defined on the MVCPool control statement.

### MIGRATE

optionally, for Management Classes with REPLICAT(YES) that reference this Storage Class, specifies the source VTSS (in a Cluster) for VTV migration. This parameter cannot be specified if FROMLST is specified.

### RECEIVER

VTSS that receives the replicated VTV (the default), which is the Secondary VTSS in a Primary-Secondary Cluster.

### SENDER

VTSS that sends the replicated VTV, which is the Primary VTSS in a Primary-Secondary Cluster.

### EITHER

Either VTSS in a Peer-to-Peer Cluster. The source VTSS is randomly selected.

#### TAPEPLEX

optionally, specifies the name of the TAPEPLEX to which a copy of the VTV is exported. At least one VTSS in the configuration must also specify this name in a CLINK definition.

#### SYNC

optionally, specifies whether the exporting of a VTV to TAPEPLEX is performed synchronously.

#### NO

export of VTV to TAPEPLEX is performed asynchronously. This is the default.

#### YES

export of VTV to TAPEPLEX is performed synchronously

If a VTV is specified with two storage classes that specify synchronous exporting, only the first one is honored while the second is exported asynchronously. Likewise, if synchronous replication is specified in the management class, then synchronous exporting is ignored.

## FROMLIST

optionally, when migrating or exporting to a Storage Class, provides VTCS with a list of VTSSs to source the VTV. FROMLIST applies to both Clustered VTSSs and to VLE to VLE connections.

If you do **not** specify FROMLIST, the **default** behavior is as follows:

- For Clustered VTSSs, if a VTV copy resides on multiple VTSSs in the cluster, the VTV can be sourced from any available VTSS, which may not be optimal if the VTSS and the connected ACS are geographically distant from each other.
- For VLE to VLE connections, if a VTV copy resides on both a VTSS and one VLE and you want to migrate it to a connected VLE, the default is to use the VLE to VLE connection. Similarly, this may not be optimal if the connected VLEs are geographically distant from each other.

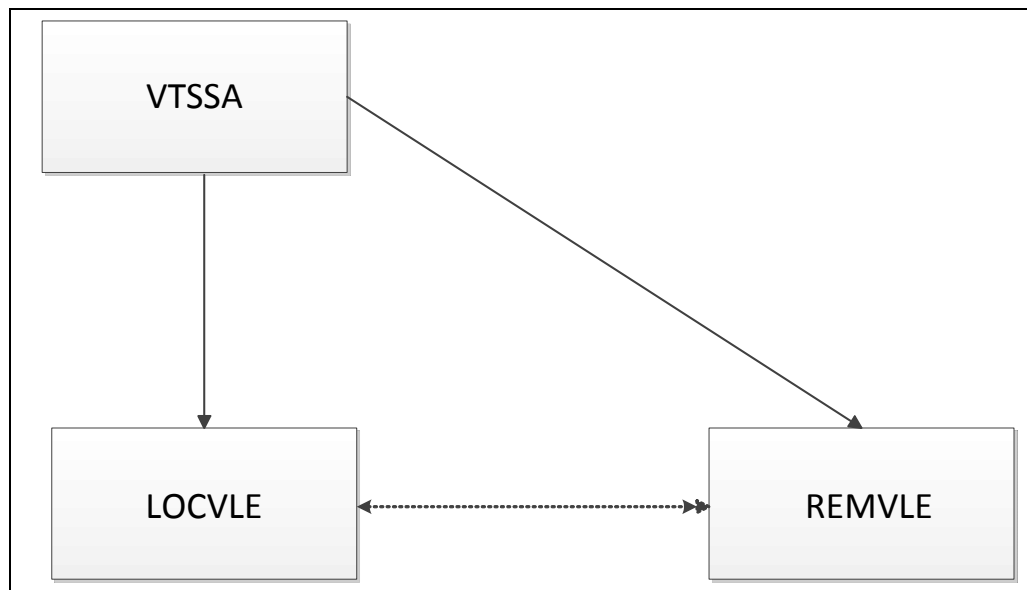
FIGURE 4-23 shows a DR scenario with a local (LOCVLE) and remote (REMVLE) connected to VTSSA, where we want to migrate two VTV copies:

- First, a copy to REMVLE, which is the DR vault site.
- Next, a local copy for quick access to LOCVLE.

We do this via the following policies:

```
STORCLAS NAME (FORREMOT) STORMNGR (REMVLE)
STORCLAS NAME (FORLOCAL) STORMNGR (LOCVLE) FROMLIST (VTSSA)
MGMTCLAS NAME (DRVLE) MIGPOL (REMVLE, LOCVLE)
```

The MGMTCLAS statement specifies to migrate a VTV copy to both REMVLE and LOCVLE. If the remote copy occurs first then for the local copy the default behavior is to migrate to LOCVLE by performing a data transfer from REMVLE. We have the VLE to VLE connections for redundancy, but in this case, we prefer **not** to use geographically distant VLE to VLE connections. Instead, by specifying FROMLIST (VTSSA), we specify that the local copy is sourced from VTSSA, **not** from REMVLE.



**FIGURE 4-23** VLE to VLE Replication Versus VTSS to

*:vtss-list-name*

a comma separated list of VTSS names (there is no limit on the number of names).

# STORMNGR - 6.2 and Above

The SMC STORMNGR command defines a VLE.

**Note** – STORMNGR and SERVER commands are required to access a VLE. The STORMNGR command can also list VLEs that the SMC tries to communicate with and report their status.

## Syntax

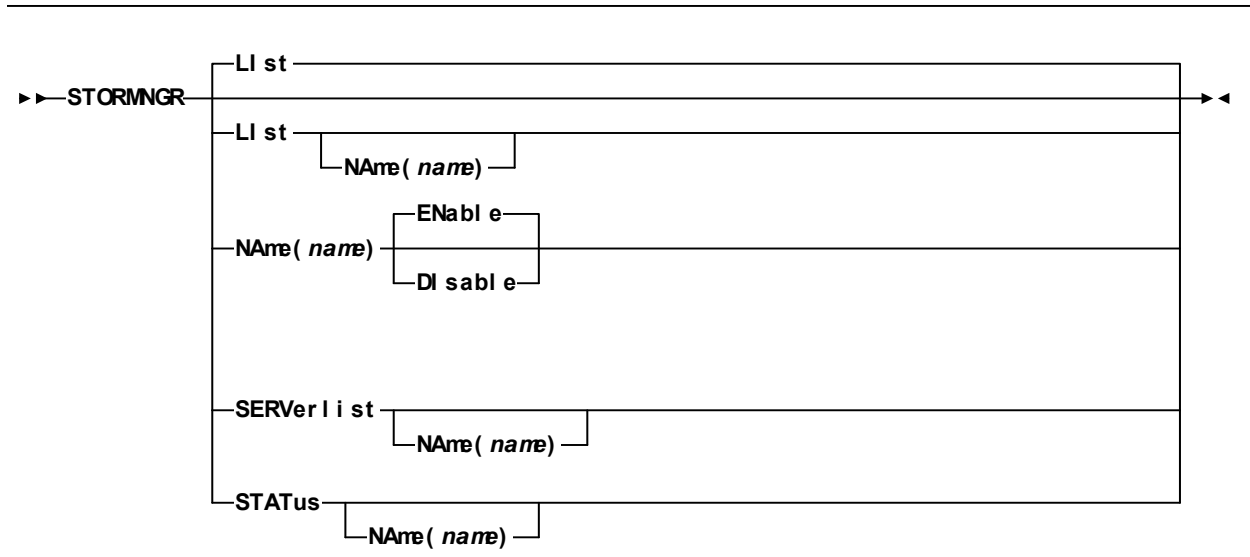


FIGURE 4-24 STORMNGR syntax



# Parameters

## List

optionally, lists all or a specific VLE.

## NAME

If specified, lists the VLE with the specified name.

## NAME

Specifies VLE to be added, modified or listed. If NAME is not specified, then all VLEs are listed.

## *name*

the VLE Subsystem name, which is also defined to VTCS. The following rules apply:

- The value must be between 1 and 8 characters in length.
- The first character must be either an alpha character or digit.
- The last character must be either an alpha character or digit.
- Any character between the first and last must be either an alpha character, digit or hyphen.

## ENABle

enables the specified VLE. This is the default when new VLE is added.

## DISABle

disables the specified VLE.

## SERVerlist

optionally, lists defined VLEs, their attributes and associated servers. The SERVerlist parameter can also be specified with the NAME parameter to limit the display to a single VLE.

## NAME

specifies the VLE name for which servers are to be listed.

## *name*

the VLE name.

## STATus

optionally, lists current status of all VLEs or a single named VLE.

# DISPLAY Command and Report Outputs

## Display CONFIG Output

[CODE EXAMPLE 4-13](#) shows an example of Display CONFIG output:

SLS6603I Configuration information						539
MaxVTV	MVCFree	VTVAttr	RECALWER	REPlicat	VTVPAGE	
4000	2	Scratch	Yes	Always	Standard	
This tapeplex: LOCALPLX						
CDS level support: V5/5.1 V6 V6.1 V6.2 V7						
* * *						
Reclaim : Threshold Max MVC Start Conmvc						
30 4 10 1						
Auto Migr Thr Migr Tasks Default VSM 2GB/ Page						
VTSSNAME	Low	High	Min	Max	ACS	Model4GB Size
DVTSS16	65	70	2	5	FF 5	Yes Large
Devno	RTD	Type	ACS	Retain	VTSSNAME	RTD NAME CHANIF
2A00	36	CTrack	00	10	DVTSS16	SS162A00 00 0A
2A01	36	CTrack	00	10	DVTSS16	SS162A01 01 0C
2A04	STK1	RC34	00	10	DVTSS16	SS162A04 02 1A
V010	VL		LEPRIM	10	DVTSS18	SS18PRIM 02 1A:0
V011	VL		LEPRIM	10	DVTSS18	SS18PRIM 02 1A:0
S011	VL		LEPRII	10	DVTSS18	SS18PRIM 03 1A:0

### CODE EXAMPLE 4-13 Example Display CONFIG output

In [CODE EXAMPLE 4-13](#), note that:

- In the Devno column, VTCS generates device numbers for VLE. Single-node VLE device numbers start with a “V” (in this case, devices V000 and V010). Multi-node VLEs are prefixed by an ‘S’.
- The RTD Type column now contains the same value as reported by HSC in the 'QUERY DRIVES' command, including the VLE device type.
- For local tape drives, the 'ACS' column displays the ACS and LSM to which the drive is attached. For remote RTDs or VLE devices, the ACS column displays the name of the VLE.

## Display MIGrate DETail Output

[CODE EXAMPLE 4-14](#) shows an example of the fields from Display MIGrate DETail output.

VTSSNAME: DVTSS16									
Active migration tasks: 2									
Immediate migrate: Max wait: 1 minutes									
Immediate delay queue: Not active									
Auto migrate: Not active									
Storage	ACS/ LOCATION	MAX/ RTDs	ONL	REQ	AUTO	IMMED		WEIGHT/ SKIP	
Class				ACT	GB	WAIT	GB		
S1	00	16	16	1	-	1	9	50	0
S2	00	16	16	1	-	1	9	50	0
S3	**ANY**	10	10	1	-	1	0	100	0
S4	LE1	0	0	0	-	-	-	0	0

### CODE EXAMPLE 4-14 Example Display MIGrate DETail output

In [CODE EXAMPLE 4-14](#), note that in the ACS/LOCATION field, for migrations to VLEs, this contains the name of the VLE. If the column contains **\*\*ANY\*\***, then migrations to any location are allowed; subject to any other restrictions.

## Display MVCPool Output

**CODE EXAMPLE 4-15** shows an example of Display MVCPool output (no pool name specified).

ACS	MVCPool INFORMATION						
	MEDIA	FREE-MVCS		RECLAIM-MVCS		USED-MVCS	
		VOLS	GB	VOLS	GB	VOLS	GB
00	ECART	310	248	4	1.2	100	65
00	ZCART	120	192	1	0.5	250	400
00	TOTAL	430	440	5	1.7	350	465
LE1	VL-MVC	90	144	15	6.2	322	485
LE1	***TOTAL	90	144	15	6.2	322	485
**MANY*	ECART	1	1	0	0	0	0
*							
**MANY*	TOTAL	1	1	0	0	0	0
*							
***WARNING - NO RTDS DEFINED FOR THIS ACS							
NON-LIB	STK2P	22	1100	0	0	12	1565
NON-LIB	TOTAL	22	1100	0	0	12	1565

**CODE EXAMPLE 4-15** Example output from Display MVCPool (no pool name specified)

In [CODE EXAMPLE 4-15 on page 92](#), note that:

- For local MVC volumes, the ACS column displays the ACS in which it is currently stored. For remote MVC volumes or VMVCs in VLEs, the ACS column displays the name of the VLE that holds them. For example, LE1 is a VLE name for VMVCs.
- The special name **\*\*MANY\*\*** appears in the 'ACS' column if a MVC with a specific volser has been discovered in more than one location. MVCs in this state are marked unusable.
- ACSs with no RTDs attached are flagged (in this case, LE1, which is a Tapeless configuration with a VLE attached).

# MVCRPT/MVCPLRPT/DISPLAY MVC Output

CODE EXAMPLE 4-16 shows an example of an MVC summary report.

SLUADMIN (7.0.0)					StorageTek VTCS SYTEM UTILITY						PAGE 0002	
TIME 09:26:54					VTCS MVC SUMMARY REPORT						DATE 2009-04-13	
MVC	NUMBER	%USED	%AVAIL	%FRAG	MEDIA	TIMES	STATUS	<-----LAST MOUNTED----->			ACS	OWNER/
VOLSER	OF VTVS				SIZE (GB)	MOUNTED	I B L D R U T M	DATE	TIME	VTSS	ID	CONSOLI DATE TIME
EVS99	200	10.80	84.57	4.63	2	310	I - - - - U - M	2009MAR15	03:20:23	VTSS8	00	S1
EVS100	0	0.00	100.00	0.00	UNKNOWN	206	- - L - - U - -	2009MAR10	05:24:04	VTSS8	--	
EVS101	1009	99.00	0.00	1.00	.4	306	I - - - - U - -	2009MAR15	03:20:23	VTSS8	00	S1
EVS102	5	8.25	91.75	0.00	.4	6	I - - - - U - -	2009MAR15	04:23:04	VTSS8	00	S3
EVS103	EXPVTV	0.12	99.88	0.00	.4	194	I - - - - J - -	2009MAR15	03:20:28	VTSS10	00	VTSS10
EVS104	0	0.00	100.00	0.00	.4	5	I - - - R C - -	2009MAR18	03:49:14	VTSS8	00	2009APR12
												03:49:14
EVS104	200	10.80	84.57	4.63	10.2	254	I - - - R U T -	2009MAR18	04110:09	VTSS8	00	
EVS105	300	15.80	54.57	4.63	10.2	154	I - - - R U W -	2009MAR18	04110:09	VTSS8	00	
EVS106	0	0.00	100.00	0.00	.4	202	I - - - - C - -	2009MAR18	03:49:20	VTSS8	00	
EVS107	0	0.00	100.00	0.00	.4	0	I - - - - R E - -	2009MAR18	04:13:00	VTSS8	00	
EVS110	0	0.00	100.00	0.00	UNKNOWN	0	- - - - - U - -				**MANY**	
EVS460	1	0.01	99.99	0.00	40	1	I - - - - U - -	2009MAR18	04:13:00	VTSS8	SECNDARY	
EVS480	1	10.8	84.57	4.63	250	10	I - - - - U - -	2009MAR18	04:13:00	VTSS8	LEPRIM	

CODE EXAMPLE 4-16 Example MVC summary report

In [CODE EXAMPLE 4-16](#), note that:

- Volume media size is now reported in GB.
- Although VMVCs in a VLE have a nominal capacity of 250GB, their capacity will adjust over time depending upon any additional compression or overheads added to the VMVC. Under certain circumstances this could result in an empty VMVC showing as considerably less than 250GB in capacity.
- For local MVC volumes, the 'Location' column reports the ACS in which it is currently stored. For remote MVC volumes or VMVCs within a VLEs, the 'Location' column reports the name of the corresponding VLE.
- The special name **\*\*MANY\*\*** appears in the 'Location' column if a MVC with a specific volser has been discovered in more than one location. MVCs in this state are marked unusable.

**Note –** To return location information for the remotely located MVCs, SMC must be running when running the reports.

- The same changes also apply to the output of the `DISPLAY MVC` command.

# New/Updated Messages

## HSC/VTCS Messages

### SLS6042E

HOSTID CCCCCCCC|DRACS AA|DRVTSS CCCCCCCC STORMNGR CCCCCCCC  
NOT FOUND IN DATA BASE

**Explanation:** An attempt to create the DR test environment with the SWUADMIN utility has failed because either the host id, ccccccc, the ACS- id, aa, the VTSS name, ccccccc, or the storage manager id, ccccccc specified on the HOSTID, DRACS, DRVTSS, or STORMNGR parameter of the DRTEST control statement did not exist in the current CDS.

**System Action:** The SWUADMIN utility terminates.

**User Response:** All HOSTID host ids must exist in the current production HSC CDS. If the HOSTID host id was incorrectly specified, correct the specification and re-execute the SWUADMIN utility. If the HOSTID host id does not exist in the production CDS, add it with the HSC SLUADMIN SLUSET utility function. Similarly, the ACS-id, VTSS names and storage manager id must exist in the current production HSC CDS. If any of these were incorrectly specified, correct it then re-execute the SWUADMIN utility.

### SLS6046E

ACS AA | VTCS CCCCCCCC | STORMNGR CCCCCCCC STATUS  
ON|OFF|SPARE|NOT SPARE DOES NOT MATCH DRTEST CREATE INPUT

**Explanation:** An attempt to create the DR test environment failed because the DRTEST PRIMEPRD or DRTEST CREATE function was previously run, and the current DRTEST CREATE configuration does not match the production CDS. Either:

- The ACS id on the production CDS is set to DRTEST ON but the ACS id is not in the current DRTEST CREATE
- The ACS id on the production CDS is set to DRTEST OFF but the ACS id is in the current DRTEST CREATE
- The VTSS id on the production CDS is set to DRTEST ON but the VTSS id is not in the current DRTEST CREATE
- The VTSS id on the production CDS is set to DRTEST OFF but the VTSS id is in the current DRTEST CREATE
- The VTSS id on the production CDS is set to DRTEST ON and SPARE, but the DRTEST CREATE does not specify SPARE
- The VTSS id on the production CDS is set to DRTEST ON and not SPARE, but the DRTEST CREATE specified SPARE
- The STORMNGR id on the production CDS is set to DRTEST ON but the STORMNGR id is not in the current DRTEST CREATE



- The STORMNGR id on the production CDS is set to DRTEST OFF but the STORMNGR id is in the current DRTEST CREATE

**System Action:** The DRTEST CREATE terminates with a return code of 8.

**User Response:** Execute either the DRTEST RESET or DRTEST PRIMEPRD function to reset the production CDS DRTEST statuses to match the desired DRTEST configuration. Then re-execute the DRTEST CREATE function.

#### **SLS6089E**

STORMNGR CCCCCCCC NOT CONNECTED TO ANY VTSS IN THE DRVTSS  
PARAMETER.

**Explanation:** An attempt to create the DR test environment with the SWUADMIN utility has failed because the storage manager id, ccccccc, specified in the STORMNGR parameter is not connected to any VTSS specified in the DRVTSS parameter of the DRTEST control statement.

**System Action:** The SWUADMIN utility terminates.

**User Response:** Correct the storage manager id specified in the STORMNGR parameter and re-execute the SWUADMIN utility.

## SMC Messages

### SMC0116

Cannot find TAPEPLEX|STORMNGR *PPPPPPP* for SERVER *SSSSSSSS* [at line *nnnn* of SMCCMDS|SMCPARMS]

**Level:** 0

**Explanation:** A SERVER command was issued with a TapePlex or STORMNGR name that was not previously defined.

**System Action:** The server is not added or updated.

**User Response:** Specify a TAPEPLEX command to define the TapePlex or a STORMNGR command to define the STORMNGR, then specify the SERVER command.

### SMC0117

Cannot change TAPEPLEX|STORMNGR name for *existing* SERVER *SSSSSSSS* [at line *nnnn* of SMCCMDS|SMCPARMS]

**Level:** 0

**Explanation:** A SERVER command was issued with the NAME of an existing server, but its TAPEPLEX or STORMNGR name did not match the name set when the server was originally defined.

**System Action:** The command is rejected.

**User Response:** Omit the Tapeplex or STORMNGR name, change the Tapeplex or STORMNGR name to match the existing server, or change the server name to add a new server.

### SMC0128

TapePlex error:

{Fatal comm error detected|

Initialization error number *nn* or {*nn*|unlimited}|

Comm error number *nn* of {*nn*|unlimited}|

Comm error limit exceeded}

JOB=*JJJJJJJ* *IIIIII* TASK=*XXXXXXXXXXXXXXXXXX* {MSG=*XXXXXXXX*}

TAPEPLEX|STORMNGR=*TTTTTTTT* {SUBSYSTEM=AAAA|SERVER=SSSSSSSS}

REQUEST=*FFFF*

{Client {IP=*NNN.NNN.NNN.NNN*} socket=*NN* port={*nnnn*|ANY}}}

{Server IP=*NNNN.NNNN.NNNN.NNNN* port=*nnnn*}

{Bytes out=*nnnn* in=*nnnn*}

{Error=*EEEE....EEEE*}

{Reason=*RRRR....RRRR*}

{Response from STK HTTP server follows: *HHHH...HHHH*}

SMC comm RC=*nnnn*

**Explanation:** The SMC encountered an interface or communication error attempting to communicate with a TapePlex or STORMNGR. The SMC0128 multiline message first lists the jobname, transaction type, and TapePlex or STORMNGR name associated with the error.

If the communication error was produced for a local TapePlex or STORMNGR using cross memory services on this same host (i.e., not using a server) then the next line will list the interface error.

If the communication error was produced for a remote HTTP server or its associated remote HSC TapePlex or STORMNGR using TCP/IP, then one of the message reason lines will be displayed.

Examples of the reason strings include:

- Specific TCP/IP function errors (connect, send, recv, etc.)
- Data error (incomplete or invalid data response)
- Subsystem inactive, not found, or at an incompatible release level
- Subsystem function error
- HTTP server not authorized
- HSC ASCOMM error
- Interface or communication timeout

**Note** – Certain remote errors may result in a display of the entire HTTP server response as follows:

Response from HTTP server follows:

```
HTTP 1.0 401 Unauthorized
```

If the message indicates “Comm error limit (nnn) exceeded” then the SMC0128 message will be followed by an SMC0119 message and the server path will be disabled by the SMC.

If the message indicates an “Initialization error” then the error occurred before any successful communication to the named server path. Such errors are not counted against the cumulative error count on the server path, and will not result in the named server being automatically disabled by the SMC.

Also, “Initialization error” messages will not be generated for every request, but will only be generated at 5 minute intervals until the path is successfully activated.

**System Action:** The allocation or mount event may not be processed by the SMC.

**User Response:** Use the specified error reason to determine the cause of the problem. If the error was produced for a remote server, verify that the HTTP server is active.

### SMC0133

```
TAPEPLEX|STORMNGR=PPPPPPPP  
CCCC....CCCC  
Status={ disabled|active|inactive|never active}  
Requests=nnnn  
[SERVER=SSSSSSSS  
Status={ disabled|active|inactive|never active}]
```

**Level:** 0

**Explanation:** A TAPEplex or STORMNGR command was issued with the LIST keyword. The SMC0133 multiline message lists parameters and status for each TapePlex or STORMNGR defined to the SMC subsystem. Optionally, if the SERVERlist keyword was specified, the server status for all servers associated with this TapePlex or STORMNGR is also displayed.

**Status** indicates the status of the TapePlex or STORMNGR.

- **disabled** indicates that the TapePlex or STORMNGR has been disabled by an operator command.
- **active** indicates that the last communication to this TapePlex or STORMNGR was successful.
- **inactive** indicates that a communication path to this TapePlex or STORMNGR is no longer active, although one was previously active.
- **never active** indicates that a communication path to this TapePlex or STORMNGR was never successfully established.

Requests indicates the total number of requests (configuration, volume lookup, mount, dismount, and swap) that were directed to the specified TapePlex or STORMNGR.

If the SERVER keyword was specified, then each server path defined for this TapePlex or STORMNGR will also be displayed, along with its status.

**System Action:** None.

**User Response:** None.

### SMC0138

```
XML {input|output} parse error RC=nnn; transaction=TTTTTTTT  
TAPEPLEX|STORMNGR=PPPPPPPP
```

**Level:** 0

**Explanation:** The SMC encountered an XML parse error. Input XML errors are produced when the input XML transaction cannot be parsed. Output XML errors occur when transaction response data cannot be converted to XML.

**System Action:** Depending upon the type of error, and server characteristics, the allocation or mount event may not be processed by the SMC.

**User Response:** Contact StorageTek Software Support.

### SMC0167

CCCCCCC summary:

TAPEPLEX|STORMNGR *PPPPPPPP* is {disabled|inactive|active on  
{local subsystem *SSSS*|server *SSSSSSSS*}  
{All TAPEPLEX|STORMNGR(s) active|  
n of n TAPE TAPEPLEX|STORMNGR(s) active|  
WARNING: All TAPEPLEX|STORMNGR(s) inactive|  
WARNING: No TAPEPLEX|STORMNGR(s) defined|  
WARNING: No TAPEPLEX|STORMNGR(s) enabled}

**Level:** 0

**Explanation:** The *CCCCCCCC* command was issued and a TapePlex and STORMNGR resynchronization was performed. Each TapePlex is represented by a line in the multiline WTO displaying its status.

**System Action:** None.

**User Response:** None.

### SMC0172

Specified TAPEPLEX|STORMNGR *PPPPPPPP* not  
{defined|HSC|active|enabled|valid for UI}

**Level:** 0

**Explanation:** An SMC command was issued specifying TAPEPLEX or STORMNGR *PPPPPPPP*. However, the command cannot be completed because the TAPEPLEX or STORMNGR is either not defined to the SMC, or is not eligible.

**System Action:** The command is not processed.

**User Response:** Either name a valid TAPEPLEX or STORMNGR name, or correct the TAPEPLEX or STORMNGR status and reissue the command.

### SMC0173

Response from TAPEPLEX|STORMNGR *PPPPPPPP*:  
*CCCC....CCCC*  
Response RC=*nn*

**Level:** 0

**Explanation:** An SMC Route command was issued that specified TapePlex or STORMNGR *PPPPPPPP*. The SMC0173 message lists the TapePlex or STORMNGR name, followed by the response from the specified TapePlex or STORMNGR, terminated by an SMC0173 message displaying the command return code.

**System Action:** None.

**User Response:** None.

### SMC0175

Communication initialized on TAPEPLEX|STORMNGR=*name* {SERVER=*name*}

**Level:** 4

**Explanation:** The SMC has successfully communicated with the specified TapePlex or STORMNGR for the first time. Additionally, if the communication path selected was a remote server, then the server is also displayed.

**System Action:** Processing continues.

**User Response:** None.

**Note –** This message is produced each time communication switches from one server to another, or communication is re-established after an error.

### SMC0203

COMMTEST:

JOB=JJJJJJJ IIIIIII TASK=XXXXXXXXXXXXXXXXX {MSG=XXXXXXXX}  
TAPEPLEX|STORMNGR=LLLLLLL {SUBSYSTEM=AAAA|SERVER=SSSSSSS}  
REQUEST=FFFF  
{Client {IP=NNN.NNN.NNN.NNN} socket=NN port={nnnn|ANY}}  
{Server IP=NNNN.NNNN.NNNN.NNNN port=nnnn}  
{Bytes out=nnnn in=nnnn}  
{Error=EEEE....EEEE}  
{Reason=RRRR....RRRR}  
{Response from STK HTTP server follows: HHHH...HHHH}  
Current LIBPATH status={active|inactive|never active|disabled}  
SMC comm RC=nnnn elapsed time=nn.nn

**Level:** 0

**Explanation:** A COMMtest command was entered. The SMC0203 message is displayed for each communication path attempted.

**System Action:** None.

**User Response:** None.

### SMC0204

No eligible COMMPATH(s) found

**Level:** 0

**Explanation:** A COMMtest command was entered, but the specified TAPEPlex, STORMNGR, SERVER, and status parameters resulted in no eligible communication paths selected for the test.

**System Action:** None.

**User Response:** Correct and reissue the COMMtest command.

**SMC0207**

Specified SERVER *SSSSSSSS* not {found|defined for  
TAPEPLEX|STORMNGR=*TTTTTTTT*}

**Level:** 0

**Explanation:** A COMMtest command was entered specifying a specific TapePlex or STORMNGR and server. However, the server is either not defined to the SMC, or is not defined for the specified TapePlex or STORMNGR.

**System Action:** None.

**User Response:** Correct and reissue the COMMtest command.

**SMC0226**

Path switch from SERVER=*SSSSSSSS* to *PPPPPPPP* for  
TAPEPLEX|STORMNGR=*TTTTTTTT*

**Level:** 4

**Explanation:** The SMC automatically switched the communication path from the secondary server *SSSSSSSS* to the primary server *PPPPPPPP* for TapePlex or STORMNGR *TTTTTTTT*.

**System Action:** Processing continues.

**User Response:** None.

