

Virtual Library Extension

Configuring the MVS Host Software for VLE

Version 1.0

E14094-03



Revision 03

Submit comments about this document to STP_FEEDBACK_US@ORACLE.COM.

Oracle welcomes your comments and suggestions for improving this book. Contact us at STP_FEEDBACK_US@ORACLE.COM. Please include the title, part number, issue date, and revision.

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

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

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

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

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

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

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

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

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

Table of Contents

Preface	5
Audience	5
1 What is Virtual Library Extension?	7
The VLE in a VSM System	9
VLE Hardware and Software	10
2 Configuring the MVS Host Software	13
Key Configuration Values	14
Subsystem Name	14
VTSS Ethernet Port Addresses	14
IP Addresses of VLE Ports for Host (UUI) Communication	14
VMVC Volsers	14
VMVC Reclamation Threshold	14
MVS Host Software Configuration Tasks	15
Updating the SMC OMVS RACF Security Entry	15
Modifying the SMC SCMDS file	16
Updating the VTCS CONFIG Deck to Define VLE	17
Defining the VLE VMVCs to the MVS Host Software and Including VMVCs in an MVC Pool ..	19
Creating VMVC Volume Pools (6.2)	19
Updating the MVS Host Software Policies	20

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 VTCS 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.

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** if you have a VLE system, HSC/VTCS uses SMC communication services to communicate with the VLE. To ensure that these services are available during VTCS startup, Oracle recommends that you first issue the start command for HSC, then immediately issue the start command for SMC, while HSC is initializing.
- **Also 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 one 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.
- Virtual Tape Control Subsystem (VTCS) software and Storage Management Component (SMC).
- VLE hardware and software.

The VLE in a VSM System

FIGURE 1-1 shows a VLE connected to a VSM system.

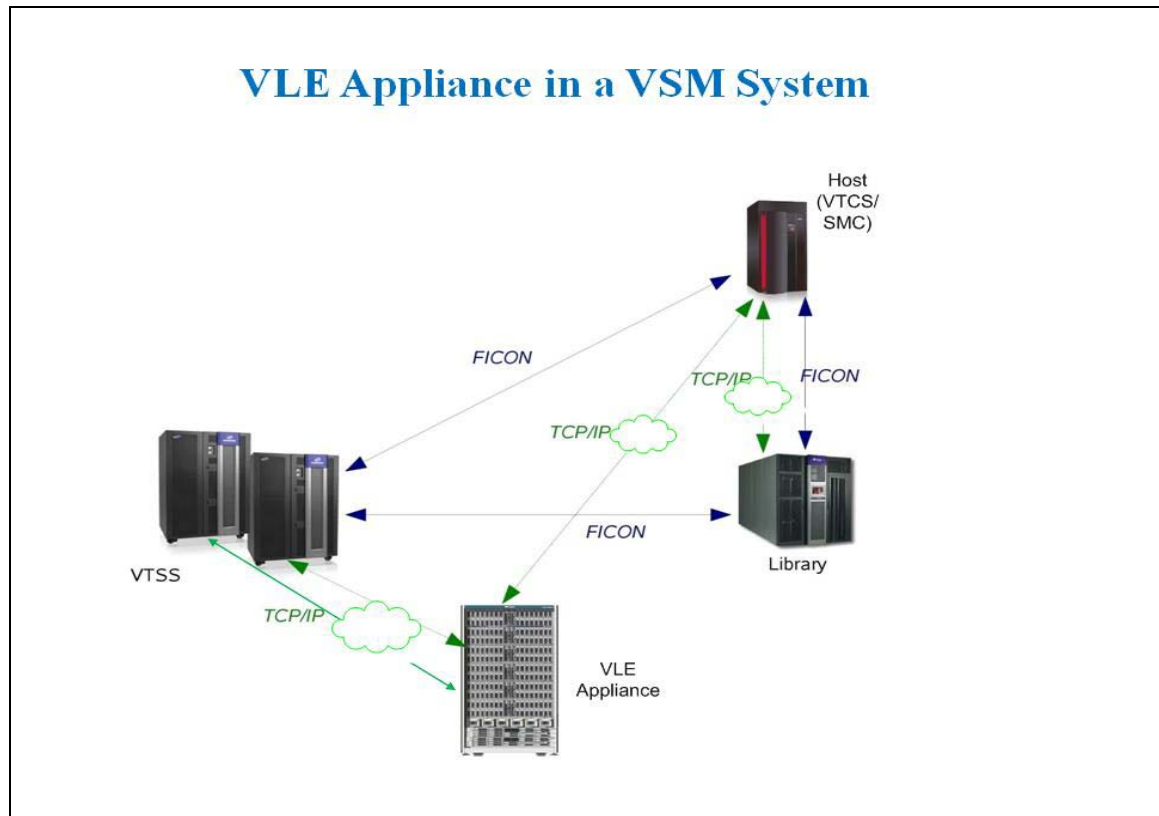


FIGURE 1-1 VLE in a VSM System

As FIGURE 1-1 shows:

- Multiple TCP/IP connections (between the VTSS's 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 MVS hosts running SMC and VTCS.

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 UUI 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 220TB, 440TB, 660TB, or 880TB (assuming a 4 to 1 compression ratio when the data is migrated to the VLE).

The VLE software consists of:

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

FIGURE 1-2 shows the VLE subsystem architecture.

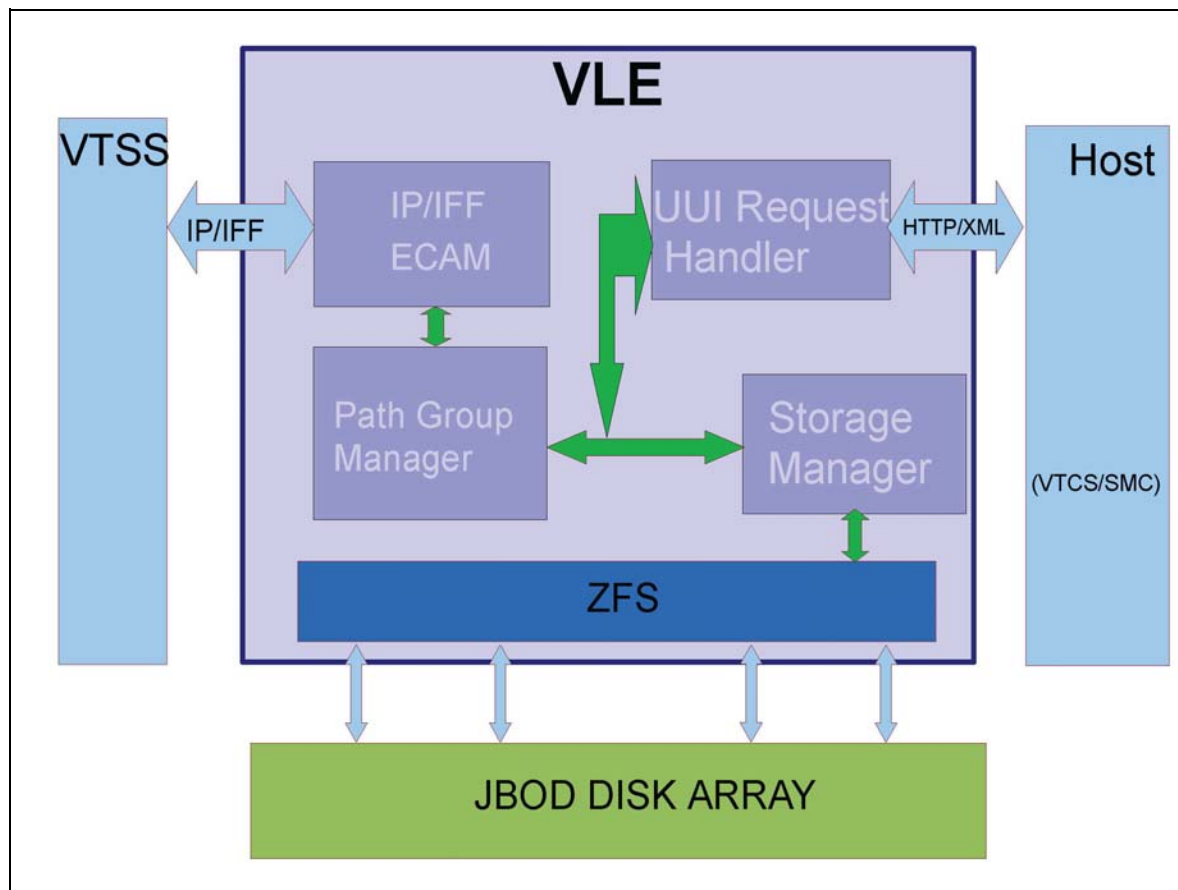


FIGURE 1-2 VLE Subsystem Architecture

As [FIGURE 1-2 on page 10](#) shows, the VLE application software is comprised of:

- HTTP/XML is the data protocol for MVS 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 appliance.
- The Storage Manager to schedule all report generation.
- The VLE Storage Manager component manages the VMVC/VTV data and meta data on the VLE appliance. 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 MVS host to VLE communications, where the IP/IFF/ECAM component handles communications between the VTSS and the VLE appliance.

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 14](#)
- [“MVS Host Software Configuration Tasks” on page 15](#)

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 can be reset by the VLE GUI, 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.

VTSS Ethernet Port Addresses

The VTSS Ethernet port addresses are required to configure the VTSS to VLE IP connection via the CONFIG RTD IPIF parameter. For VSM5s, this value must match the values specified on the VSM5 IFF Configuration Status Screen. For VSM 6s, this must be unique for each VTSS but does **not** correspond to an actual value on the VSM 6 TCP/IP ports.

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 19.](#)

VMVC Reclamation Threshold

For more information, see [“VMVC Reclamation Threshold” on page 14.](#)

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 15](#)
- [“Modifying the SMC SCMDS file” on page 16](#)
- [“Updating the VTCS CONFIG Deck to Define VLE” on page 17](#)
- [“Defining the VLE VMVCs to the MVS Host Software and Including VMVCs in an MVC Pool” on page 19](#)
- [“Updating the MVS Host Software Policies” on page 20](#)

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 LNKLST` 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 (ALTSESV) TAPEPLEX (TMVSA) +
HOSTNAME (MVSX) PORT (8888)
STORMNGR NAME (VLE1)
SERVER NAME (VLESESV1) +
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 (`ALTSESV`) running on another host.
- A `STORMNGR` command that defines a VLE (`VLE1`) .
- A second `SERVER` command that defines a UUI communication path to the VLE appliance, where:
 - The server name is `VLESESV1`.
 - The `STORMNGR` parameter value is `VLE1` .
 - The `IP` parameter value is the VLE port IP address of 192.168.1.10 for UUI communications.
 - The `PORT` parameter value is 60000; this value is always used for the `SERVER PORT` parameter for SMC communication with a VLE appliance.

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. 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

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 name of the RTDPATH.
 - The connections to the defined VLEs (STORMNGR=VLE1).
 - The IPIF value for each VTSS to VLE port connection in *ci:p* format where:
 - *c* is 0 or 1.
 - *i* is A or I.
 - *p* is 0 through 3.

Note – For VSM5s, this value must match the values specified on the VSM5 IFF Configuration Status Screen. For VSM 6s, this must be unique for each VTSS but does **not** correspond to an actual value on the VSM 6 TCP/IP ports.

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 volsers 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 node or grid and a “remote” Storage Class (VREMOTE) on the on the VLSERV2 node or grid. 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*

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.

