Virtual Tape Control System

Command and Utility Reference

MVS Software

Version 6.2

E22967-02

Revision 02

Submit comments about this document to STP_FEEDBACK_US@ORACLE.COM

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

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

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

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

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

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

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

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

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

Preface

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

Audience

This reference provides VTCS and NCS reference information for StorageTek or customer personnel who are responsible for all VTCS tasks. Also see *Installing and Configuring VTCS* for more information.

Prerequisites

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

- MVS or OS/390 operating system
- JES2 or JES3
- System Management Facility (SMF)
- Nearline Control Solution (NCS)

Organization of This Publication

This table provides a general overview of the sections in this publication.

TABLE P-1

Chapter 1, “VTCS Utilities and Commands”	command and utility reference information
Chapter 2, “SMC Support for VSM”	
Chapter 3, “HSC Support for VSM”	
Chapter 4, “MVS/CSC Support for VSM”	
Chapter 5, “LibraryStation Support for VSM”	
Appendix A, “VTCS SMF Record Formats”	HSC SMF subtypes for VTCS
Appendix B, “NCS/VTCS Alphabetic Volsers”	reference and usage information about alphabetic volsers
Appendix C, “Using the HSC Significant Event Notification Facility”	reference and usage information about the HSC SEN

This book also has a glossary and index.

What's New in this Publication?

VTCS 6.2.0, Revision 01

Revision 01 contains technical updates and corrections.

VTCS 6.2.0, Revision R

Revision R contains technical updates and corrections.

Revision Q

The VTCS 6.2, Revision Q of this guide contains information about the VTCS 6.2 enhancements described in TABLE P-2.

TABLE P-2 VTCS 6.2 Updates to VTCS CU Reference, Revision Q

This Enhancement...	...is described in...	...and requires the following PTFs...
Support for VLE 1.0	▪ "CONFIG Utility RECLAIM Statement" on page 28 ▪ "CONFIG Utility RTD Statement" on page 35 ▪ "CONFIG Utility TAPEPLEX Statement" on page 39 ▪ "Display MVCPool Output" on page 80 ▪ "Display CONFIG Output" on page 92 ▪ "Display MIGrate DEtail Output" on page 96 ▪ "MVC Summary Report" on page 140 ▪ "COMMtest" on page 196 ▪ "Route" on page 205 ▪ "SERVER Command" on page 207 ▪ "STORMNGR" on page 211 ▪ "ACTMVCGN" on page 226 ▪ "STORCLAS Control Statement" on page 252	▪ L1A00R6 ▪ L1A00RZ ▪ L1A00SY ▪ L1H158F ▪ L1H158G ▪ L1H158H ▪ L1H155S ▪ L1H155T ▪ L1H15G9 ▪ L1H15H0 ▪ L1H15H7 ▪ L1H15NA ▪ L1H15QL

VTCS 6.2.0, Revision P

Revision P contains technical updates and corrections.

Revision O

The VTCS 6.2, Revision O of this guide contains information about the VTCS 6.2 enhancements described in TABLE P-7.

TABLE P-3 VTCS 6.2 Updates to VTCS CU Reference, Revision O

This Enhancement...	...is described in...	...and requires the following PTFs...
MVCMAINT enhancements	"MVCMAINT" on page 124	n SWS6200 - L1H15GT n SOS6200 - L1H15GV n SMS6200 - L1H15GU

VTCS 6.2.0, Revision N

Revision N contains technical updates and corrections.

VTCS 6.2.0, Revision M

Revision M contains technical updates and corrections.

VTCS 6.2.0, Revision L

Revision L contains technical updates and corrections.

Revision K

The VTCS 6.2, Revision K of this guide contains information about the VTCS 6.2 enhancements described in TABLE P-7.

TABLE P-4 VTCS 6.2 Updates to VTCS CU Reference, Revision K

This Enhancement...	...is described in...	...and requires the following PTFs...
CONFIG CLINK IPIF parameter	"CONFIG Utility CLINK Statement" on page 43	L1H153L, L1H153M and L1A00QO
Tapeless VSM	"MGMTCLAS Control Statement" on page 234	n L1H14XS - SMS6200 n L1H14XT - SOS6200 n L1H14Y7 - SWS6200
Migration control enhancements	n "MIGRSEL Control Statement" on page 244 n "Display MIGrate DEtail Output" on page 96	L1H14M8 (SWS620) and L1H14MA (SOS620)
CONFIG GLOBAL MAXVTVSZ parameter	"CONFIG Utility GLOBAL Statement" on page 22	L1H153L, L1H153M and L1A00QO

Revision J

The VTCS 6.2, Revision J of this guide contains information about the VTCS 6.2 enhancements described in TABLE P-7.

TABLE P-5 VTCS 6.2 Updates to VTCS CU Reference, Revision J

This Enhancement...	...is described in...	...and requires the following PTFs...
INVENTRY utility	"INVENTRY" on page 112	n L1H14OC for SWS6200 n L1H14OA for SOS6200 n L1A00PL for SMC6200 n L1H14OB for SMS6200
MVCATTR Control Statement	"MVCATTR Control Statement" on page 247	n SOS6200 - L1H150A n SWS6200 - L1H1509 n SMS6200 - L1H1508
DISPLAY RTD output enhancements	"Display RTD Output" on page 68	L1H14K7 for SWS6200
DISPLAY MVCPOOL output enhancements	"Display MVCPool Output" on page 80	n L1H14VW (SWS6200) n L1H14VV (SMS6200) n L1A00Q7 (SMC6200)
MVCMAINT enhancements	"MVCMAINT" on page 124	L1H14JK (SWS6200)

VTCS 6.2.0, Revision I

Revision I contains technical updates and corrections.

Revision H

The VTCS 6.2, Revision H of this guide contains information about the VTCS 6.2 enhancements described in TABLE P-7.

TABLE P-6 VTCS 6.2 Updates to VTCS CU Reference, Revision H

This Enhancement...	...is described in...	...and requires the following PTFs...
MVC Initialization	n"CONFIG Utility GLOBAL Statement" on page 22 n"MVCPOOL Control Statement" on page 249 n"Display CONFIG Output" on page 92	L1A00OO (SMC), L1H14DE (SMS), L1H14DF (SOS), L1H14DG and L1H14H5 (SWS)

Revision G

The VTCS 6.2.0, Revision G of this guide contains information about the VTCS 6.2 enhancements described in TABLE P-7.

TABLE P-7 VTCS 6.2.0 Updates to VTCS CU Reference, Revision G

This Enhancement...	...is described in...
Maximum 32 RTDs	▪ "CONFIG Utility GLOBAL Statement" on page 22 ▪ "Display CONFIG Output" on page 92 ▪ "Display CLINK Output" on page 101 ▪ "SLSSMF16 - VTCS SMF Subtype 16 Record" on page 298 ▪ "SLSSMF17 - VTCS SMF Subtype 17 Record" on page 300 ▪ "SLSSMF18 - VTCS SMF Subtype 18 Record" on page 301 ▪ "SLSSMF19 - VTCS SMF Subtype 19 Record" on page 303 ▪ "SLSSMF21 - VTCS SMF Subtype 21 Record" on page 306
Stacked Migrates	"CONFIG Utility GLOBAL Statement" on page 22

32 RTDs/Stacked Migrates support, which applies to only VSM5s, has the requirements described in TABLE P-13.

TABLE P-8 32 RTDs/Stacked Migrates Support Requirements for VTCS/NCS 6.2

32 RTDs/Stacked Migrates Support requires...	..the following VSM4/VSM5 microcode...	...and the following VTCS/NCS 6.2 PTFs...	...and CDS level...
FICON RTDs and FICON ports for the CLINKs	D02.05.00.00 or higher	L1H13ZF (SOS6200) L1H13ZG (SWS6200	"F" or higher

VTCS 6.2.0, Revision F

Revision F contains technical updates and corrections.

VTCS 6.2, Revision E

The VTCS 6.2, Revision Eof this reference contains information about the VTCS 6.1 enhancements described in Table 10.

TABLE P-9 VTCS 6.2 Updates to VTCS Command and Utility Reference, Revision E

This Enhancement...	...is described in...	...and requires...
change to MVCPOOL MVCFREE behavior	"MVCPOOL Control Statement" on page 249	PTF L1H141W - SWS6200

VTCS 6.2.0, Revision D

The VTCS 6.2.0, Revision D of this reference contains information about the VTCS 6.2 enhancements described in TABLE P-10.

TABLE P-10 VTCS 6.2.0 Updates to VTCS Command and Utility Reference, Revision D

This Enhancement...	..the following VSM4/VSM5 microcode...	...and the following VTCS/NCS 6.2 PTFs...	...is described in...
VTSS Synchronous Replication	FICON ports for the CLINKs, D02.03.00.00	L1H13QL (SWS6200), L1A00L3 (SMC6200), and L1H13K8 (SOS6200)	n"CONFIG Utility GLOBAL Statement" on page 22 n"Additional Parameters - Advanced Management Feature" on page 239 n"Display CONFIG Output" on page 92 n"Display CLINK Output" on page 101 n"Display CLUSTER Output" on page 103 n"Display VTD Output" on page 67 n"Additional Parameters - Advanced Management Feature" on page 239 n"SLSSMF28 - VTCS SMF Subtype 28 Record" on page 311 n"SLSSMF30 - VTCS SMF Subtype 30 Record" on page 314
Synchronous Replication SMF Record changes	D02.04.00.E3	L1H13XA - SOS6200 L1H13XB - SWS6200	n"SLSSMF14 - VTCS SMF Subtype 14 Record" on page 294 n"SLSSMF27 - VTCS SMF Subtype 27 Record" on page 309
VTVRPT SUPEMPTY parameter		L1H13X5 - SWS6200	"VTVRPT" on page 187

Synchronous replication, which applies to only VSM4s and VSM5s, has the requirements described in TABLE P-11.

TABLE P-11 Sychronous Replication Requirements for VTCS/NCS 6.2

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

VTCS 6.2.0, Revisions B and C

Revisions B and C contain technical updates and corrections.

VTCS 6.2.0, Initial Release

The VTCS 6.2.0, Initial Release of this reference contains information about the VTCS 6.2 enhancements described in Table 12.

TABLE P-12 VTCS 6.2.0 Updates to VTCS Command and Utility Reference, Initial Release

This Enhancement...	...is described in...
Hierarchical Data Movement	n"ARCHive" on page 4 n"MGMTCLAS Control Statement" on page 234 n"MIGrate" on page 60 n"Display VTV Output" on page 83
VTV Reconciliation	n"RECONcil" on page 153
Media Management enhancements	n"AUDIT" on page 11 n"MVCDRAIN" on page 121
VTCS Locks in a Coupling Facility	n"LOCKSTR=structure-name" on page 24 n"Example DECOM FLATDD output" on page 51 n"Display CONFIG Output" on page 92 n"Display LOCKs Output" on page 99
Display STORCLas	n"DISPLAY" on page 57 n"Display STORCLas Output" on page 91
SMC enhancements	n"POLICY Command" on page 199 n"SERVER Command" on page 207 n"TAPEPLEX Command" on page 213 n"TAPEREQ Control Statement" on page 217
Delete scratch VTVs	n"DELETSCR" on page 53 n"SLSSMF15 - VTCS SMF Subtype 15 Record" on page 296 n"SLSSMF29 - VTCS SMF Subtype 29 Record" on page 313
Conditional VTV Replication	n"CONFIG Utility GLOBAL Statement" on page 22 n"DECOM" on page 50 n"Display CONFIG Output" on page 92 n n"MGMTCLAS Control Statement" on page 234
Standard/Large VTV Pages	n"CDSLEVEL" on page 21 n"VTVPAGE" on page 93 n"PAGE SIZE" on page 94
400Mb/800Mb/2Gb/4gb VTVs	n"CDSLEVEL" on page 21 n"2GB / 4GB" on page 94 n"MAX VTV" on page 192 n"MAXVtvsz" on page 236
65000 VTVs per MVC	n"CDSLEVEL" on page 21 n"MAXVTV=nnn" on page 22 n"MAXVTV" on page 54 n"MAX VTV" on page 192

VTCS 6.2.0 SPEs

TABLE P-13 Updates to VTCS Command and Utility Reference - 6.2 SPEs

This SPE...	...is described in...	...and is available via the following PTF...
SET RMM command	"SET RMM" on page 174	L1H139T

Contents

CHAPTER 1

VTCS Utilities and Commands

This chapter contains reference information for the VTCS commands and utilities, where the "Interfaces" section describes the valid interfaces (command only, utility only, or both). For more information, see "Using VTCS Utilities" on page 2 and "Using VTCS Commands" on page 2.

Using VTCS Utilities

You use the SWSRTV program to run the RTV utility. To invoke all other VTCS utilities, you use the SWSADMIN program, which follows the same syntax rules and accepts the same input parameters as the HSC SLUADMIN utility program (including that the maximum length of a control statement is 32,767 characters).

The SWSADMIN program determines the HSC Primary CDS as follows:

- If your JCL **does not** specify the CDS and HSC is up, SWSADMIN queries HSC for the Primary CDS and SWSADMIN uses that CDS. The JCL examples in this chapter show this method.
- If your JCL specifies all defined copies of the CDS, SWSADMIN queries these copies and uses the correct copy. Note that for the CONFIG utility, you must explicitly specify the CDS in your JCL because you should run CONFIG when HSC is down.

Caution – As described in "Using VTCS Commands", entering VTCS commands requires a .VT before the command name. The SWSADMIN program does not require a .VT before the utility name, and adding a .VT produces an error.

Tip – HSC provides utilities that manage library resources. These utilities are also available to VSM, and include the Scratch Update utilities, which you can use to manage the scratch status of VTVs and MVCs. For more information about the HSC utilities, see *HSC System Programmer's Guide for MVS*.

Note that you cannot use the Scratch Update utilities to scratch MVCs unless you have removed them from the MVC pool.

Using VTCS Commands

Like HSC commands, VTCS commands consist of the HSC command prefix character followed immediately by a command name and required or optional positional parameters and keyword parameters.

The VTCS command prefix is the same command prefix used by the HSC with which the VTCS is communicating. This allows HSC to intercept and interpret the command. For example, to cancel all active and queued VSM processes that use an RTD, enter the following:

```
.VT CAN T(ALL)
```

For more information about the rules governing commands, refer to Chapter 2, "Commands, Control Statements, and Utilities," in *HSC Operator's Guide for MVS*.

All Hosts PTFs

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

The feature string:

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

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

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

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

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

- X’80’: VTCS locks are implemented in a coupling facility structure (the VTCS configuration specifies GLOBAL LOCKSTR=*structure-name*)
- X’40’: reserved
- X'20' CDS has been configured for Synchronous Replication
- X'10' CDS supports 32 RTDs per VTSS

ARCHive

ARCHive archives VTVs (moves them to MVCs in a different Storage Class).

Note – ARCHive is valid only if FEATures VSM(ADVMGMT) is specified.

Syntax

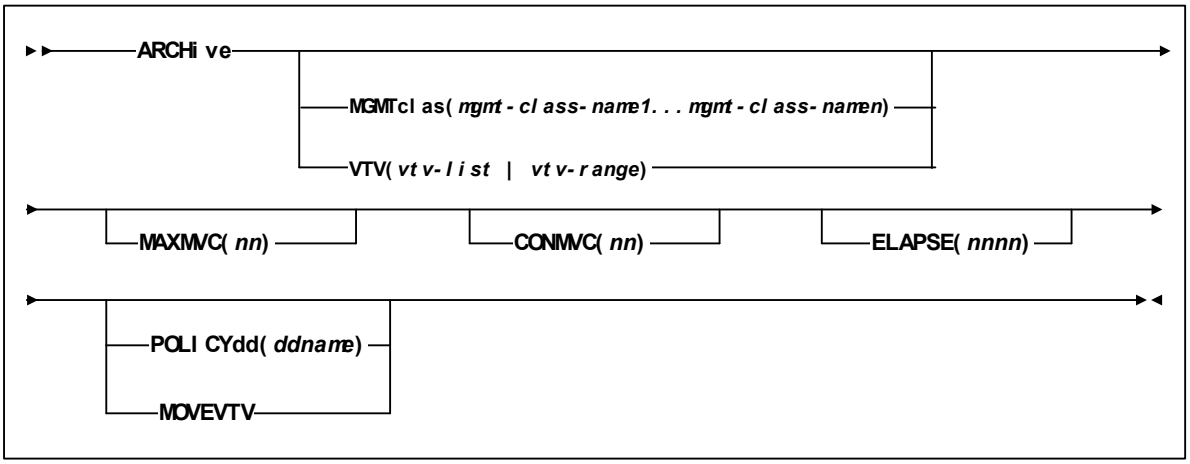


FIGURE 1-1 ARCHive syntax

Parameters

MGMTclas

Specifies one or more Management Classes that define Archive Management Policies.

mgmt-class-name1...mgmt-class-namen

One or more Management Classes.

VTV

Specifies a list or range of VTVs.

vtv-list | *vtv-range*

A list or range of VTVs.

Note – If you do not specify a value for MGMTclas or VTV, VTCS scans (or processes) all VTVs.

MOVEVTV

Archive VTVs per the currently active Management Policies as specified by the MGMTclas statements that apply to the VTVs specified by MGMTclas or VTV. If you do not specify MOVEVTV, only a report is generated.

POLICYdd(*ddname)*

Specifies the *ddname* of a file containing an alternate MGMTclas statement. This is mutually exclusive with MOVEVTV and only generates a report.

MAXMVC(*nn)*

specifies the maximum number of MVCs that will be processed by a single archive task.

Valid values are 1 to 98. If not specified the CONFIG RECLAIM value (or default) is used. This parameter is ignored if MOVEVTV is not specified.

CONMVC(*nn)*

specifies the maximum number of MVCs that VTCS concurrently processes during subsequent recall and migrate operations.

Valid values are 1 to 99. If not specified the CONFIG RECLAIM value (or default) is used. This parameter is ignored if MOVEVTV is not specified.

ELAPSE(*nnnn)*

specifies the maximum time for the archive in minutes. If the maximum time expires, VTCS issues message SLS6682I. If there are no MVCs currently mounted, archive stops when the ELAPSE value is reached. If any MVCs are currently mounted when the ELAPSE value is reached, archive processes the mounted MVCs and then stops.

Valid values are 1 to 1440. If not specified, there is no time limit on the archive process. This parameter is ignored if MOVEVTV is not specified.

Interfaces

SWSADMIN utility only.

JCL Requirements

The following are the required and optional statements for the ARCHive JCL:

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the utility report.

SLSIN

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

ARCHive Reports

FIGURE 1-2 shows an example of an ARCHive report (MOVEVTV not specified).

SWSADMIN (6.2.0) STORAGETEK VTCS SYTEM UTILITY PAGE 0001

TIME 06:32:03 ARCHIVE VTV REPORT DATE 2006-111-20

VTV	SIZE	% COMP	<----------CREATION---------->		MGMT				
VOLSER	(MB)		DATE	TIME	CLASS	MVC1	MVC2	MVC3	MVC4
Y00001	1.62	39	2006JAN15	04:11:18	MJ4	02250	02251	02252	02253
Y00002	1.62	39	2006JAN15	04:03:57	MJ4	02250	02251	02252	02253
Y00003	1.62	39	2006JAN15	03:50:59	MJ3	02254	02255		
Y00004	1.62	39	2006JAN15	03:45:04	MJ2	02256	02257	022568	
Y00005	0.01	0	2005DEC17	23:56:00	MJ1	02259			
Y00006	0.01	0	2005DEC17	23:41:37	MJ1	02259			
Y00007	1.62	39	2006JAN06	06:15:46	MJ4	02250	02251	02252	02253

TOTAL VTVS=23 TOTAL SIZE=29.32MB

TIME 10:07:10 ARCHIVE MVC REPORT DATE 2006-111-20

MVC	MEDIA	MEDIA	STORAGE	LOCATION	CANDIDATE		TOTAL VTV	
VOLSER	TYPE	SIZE (MB)	CLASS	(ACS ID)	VTVS		SIZE (MB)	
022550	9840	20000	SC1	00	3			4.86
Y00001	Y00002	Y00007						
022551	9840C	40000	SC3		3			4.86
Y00001	Y00002	Y00007						
022559	9940A	60000	SC4		17			18.50
Y00005	Y00006	Y00015	Y00027	Y00042	Y00048		Y00053	
Y00059	Y00061	Y00067	Y00073	Y00078	Y00084	Y00088	Y00101	Y00123

TOTAL MVCS=8 TOTAL SIZE=29.32MB

FIGURE 1-2 Example output from ARCHive

ARCHIVE Report Fields

The following list describes the ARCHive VTV report fields. This section of the report is followed by a total line showing the number of candidate VTVs and the size in MB to be recalled and remigrated.

VTV Volser

the VTV volser.

Size (MB)

the uncompressed size of the VTV (MB). **<MOUNT>** indicates that the VTV was mounted when the report ran. **<FENCED>** indicates that the VTV's state is unknown. If **<FENCED>** appears, contact StorageTek software support.

Comp %

the VTV compression percentage achieved. This is the difference between the uncompressed and compressed VTV size expressed as a percentage of the uncompressed VTV size. For example if a 100MB VTV compresses to 40MB then the compression% will be given as 60%. A compression of 0% indicates that no compression was possible on the VTV.

Creation Date and Time

the date and time that the VTV was created.

MGMT Class

the name of the Management Class for the VTV specified.

MVC1, MVC2, MVC3, MVC4

the MVC(s) that contain the VTV. If all of these fields are empty, the VTV has not been migrated or consolidated. If 1 or more of these fields list an MVC volser, the VTV was migrated to each of these MVCs.

MVC Report Fields

The following list describes the ARCHive MVC report fields. The data for each MVC is followed by one or more VTV volsers with copies on the MVC. This section of the report is followed by a total line showing the number of candidate MVCs and the size in MB to be recalled and remigrated.

MVC Volser

the MVC volser.

MVC Media Type

the MVC type.

Media Size (MB)

the size of the MVC (MB). This will only be determined after VTCS has used an MVC. "UNKNOWN" appears in this field until VTCS migrates a VTV to the MVC.

Storage Class

the MVC's Storage Class.

Candidate VTVs

the number of candidate VTVs on the MVC.

Total VTV Size (MB)

the size of all candidate VTVs on the MVC in MBs.

Location (ACS ID)

the ACS where the MVC resides. If blank, the MVC is not currently in an ACS.

MOVEVTV Report

FIGURE 1-3 shows an example of an ARCHive MOVEVTV report.

```
SWSADMIN (6.2.0STORAGETEK VTCS SYSTEM UTILITY PAGE 0001
 TIME 10:07:10ARCHIVE MOVEVTV REPORTDATE 2006-01-16

Move VTV - MVC 022705 ignored, MAXMVC reached
Move VTV - VTV X04898 ignored, all MVC copies rejected
Move VTV - 4 MVCs selected for processing
Move VTV - 5 VTVs selected for processing
Move VTV - 5 VTV copies to be processed
Move VTV - 0 VTV copies not matched to request
Move VTV - 1 VTV copies rejected by MAXMVC limit
Move VTVs - MVC 023484 selected and contains 1 VTVs
Move VTVs - MVC 022628 selected and contains 1 VTVs
Move VTVs - MVC 022631 selected and contains 2 VTVs
Move VTVs - MVC 022608 selected and contains 1 VTVs

Recall from MVC 022628 to VTSS HBVTSS17
SLS6683I Bulk recall of 1 VTVs issued to MVC 022628
SLS6644I VTV X99909 recalled from MVC:022628 Block:25401431
SLS6637I Recall from MVC 022628 completed

Recall from MVC 023484 to VTSS HBVTSS17
SLS6683I Bulk recall of 1 VTVs issued to MVC 023484
SLS6644I VTV X04897 recalled from MVC:023484 Block:02402581
SLS6637I Recall from MVC 023484 completed

Recall from MVC 022608 to VTSS HBVTSS16
SLS6683I Bulk recall of 1 VTVs issued to MVC 022608
SLS6637I Recall from MVC 022608 completed

Migrate to storage class HBVTSS16 from VTSS HBVTSS17
SLS6681I VTV X99909 migrated to MVC:022589 Block:01400025 StorCl:HBVTSS17 MgmtCl:SIMPLEX
SLS6636I Demand migration to MVC 022589 completed

Recall from MVC 022631 to VTSS HBVTSS16
SLS6683I Bulk recall of 2 VTVs issued to MVC 022631
SLS6644I VTV X99910 recalled from MVC:022631 Block:03400141
SLS6644I VTV X99908 recalled from MVC:022631 Block:05400281
SLS6637I Recall from MVC 022631 completed
```

FIGURE 1-3 Example output from ARCHive MOVEVTV

AUDIT

AUDIT updates the MVC and VTV information in the HSC CDS.

Syntax

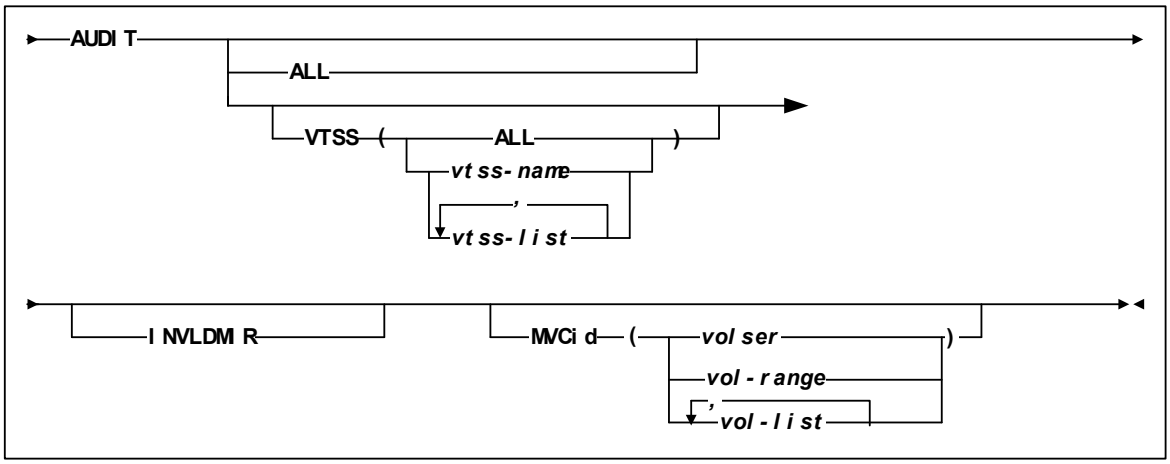


FIGURE 1-4 Audit syntax

Parameters

ALL

specifies an audit of your entire VSM system, including all VTSSs and all MVCs.

Note – The ALL parameter causes VTCS to attempt to audit all defined MVCs. If any MVC is outside the ACS, HSC will prompt you to enter the MVC into the correct LSM.

VTSS

specifies an audit of one or more VTSSs.

ALL

specifies all VTSSs.

vtss-name or *vtss-list*

the names of one or more VTSSs.

INVLDMIR

specifies an audit of MVCs with invalid MIRs.

MVCid

specifies an audit of one or more MVCs.

volser, vol-range or *vol-list*

the volsers of one or more MVCs.

Interfaces

SWSADMIN utility only.

JCL Requirements

The following are the required and optional statements for the AUDIT JCL:

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the utility report.

SLSIN

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

Audit Report

An audit report lists the VTVs and MVCs that are different from those listed in the CDS as shown in FIGURE 1-5. In this figure, the report shows all MVCs or VTVs as new entries in the CDS, which is typical of the output of a VTCS audit run after you lost all copies of the CDS, then ran recovery procedures.

```
SWSADMIN (6.2.0)                    STORAGETEK VTCS SYSTEM UTILITY
TIME 03:15:42                               VTCS AUDIT

AUDIT REPORT FOR MVC EVT500
X28955 VTV ADDED AS PRIMARY COPY (BLOCK:00000000)
X20000 VTV ADDED AS PRIMARY COPY (BLOCK:0940044D)
======== AUDIT OF MVC EVT500 COMPLETED SUCCESSFULLY ========

AUDIT REPORT FOR MVC EVT501
X28956 VTV ADDED AS PRIMARY COPY (BLOCK:00000000)
X20007 VTV ADDED AS PRIMARY COPY (BLOCK:0940044D)
X20010 VTV ADDED AS SECONDARY COPY (BLOCK:11400899)
X20069 VTV NOT CURRENT (BLOCK:1A400CE5)
X20067 VTV NOT CURRENT (BLOCK:334016AB)
======== AUDIT OF MVC EVT501 COMPLETED SUCCESSFULLY ========

AUDIT REPORT FOR VTSS HBVTSS17
X20000 VTV VALID
X20002 VTV VALID
X20005 VTV VALID
X20006 VTV VALID
X20007 VTV VALID
X30052 VTV VALID
X30053 VTV VALID
X30054 VTV VALID
 ======== AUDIT OF VTSS HBVTSS17 COMPLETED SUCCESSFULLY ========

AUDIT REPORT FOR VTSS HBVTSS16
X20183 VTV VALID
X20185 VTV VALID
X20188 VTV VALID
X20190 VTV VALID
X20191 VTV VALID
X20194 VTV VALID
X41091 VTV VALID
X41093 VTV VALID
======== AUDIT OF VTSS HBVTSS16 COMPLETED WITH 1 WARNING ========
AUDIT EXCEPTION REPORT
 VTSS HBVTSS16: 1 WARNINGS REPORTED
 SLS1315I SWS500.V5.CDS WAS SELECTED AS THE PRIMARY CONTROL DATA SET
```

FIGURE 1-5 Example AUDIT utility report

Note – An audit also generates:

- MVC summary and detail reports.
- Display VTSS summary and detail output.
- For every VTV resident on the VTSS, the VTV volser, size in MB, and Management Class.

Audit Report Messages

For every VTV found on an MVC or VTSS, the audit report lists one of following:

vvvvvv VTV possibly corrupt (Block:*bbbbbb*)

During the audit, an I/O error occurred for VTV *vvvvvv* at block *bbbbbb* on the MVC being audited.

vvvvvv VTV not found [, no MVC copies left]

The audit did not find VTV *vvvvvv on the MVC or VTSS being audited. If no MVC copies left* appears, no MVCs contain copies of t he VTV.

vvvvvv VTV not found on CDS (Block:bbbbbb)

The audit expected but did not find VTV *vvvvvv at block bbbbbb on the MVC being audited.*

vvvvvv VTV not current (Block:bbbbbb)

The audit found a non-current copy of VTV *vvvvvv at block bbbbbb on the MVC being audited.*

vvvvvv VTV copy valid (Block:*bbbbbb*)

The audit found a valid copy VTV *vvvvvv* at block *bbbbbb* of the MVC being audited; its location matches the CDS entry for the VTV.

vvvvvv VTV Added as primary copy (Block:*bbbbbb*)

The audit found the most current copy of VTV *vvvvvv* at block *bbbbbb* of the MVC being audited; the audit added this location to the CDS as the primary MVC copy of the VTV.

vvvvvv VTV Added as secondary copy (Block:*bbbbbb*)

The audit found the second most current copy of VTV *vvvvvv* at block *bbbbbb* of the MVC being audited; the audit added this location to the CDS as the secondary MVC copy of the VTV.

vvvvvv Duplicate copy ignored (Block:*bbbbbb*)

The audit found a duplicate copy of VTV *vvvvvv* at block *bbbbbb* and ignored this copy.

vvvvvv Link to old version on MVC *mmmmmm* removed

The audit found a newer version of the VTV and removed the link to the old version from the CDS.

vvvvvv Old VTV version deleted from VTSS *ssssssss*

The audit found an old version of the VTV and deleted it from the VTSS.

vvvvvv Old version of VTV retained [VTSS *ssssssss]*

The audit found an old version of the VTV, which is the only copy, and retained this version. If VTSS *ssssssss* appears, the audit found the VTV on a different VTSS than the one that was audited.

vvvvvv Version older than MVC copies [VTSS *ssssssss]*

The audit found a version of the VTV that is older than copies on the MVC. If VTSS *ssssssss* appears, the audit found the VTV on a different VTSS than the one that was audited.

vvvvvv Newer version of VTV found [on VTSS *ssssssss]*

The audit found a newer version of the VTV and updated the CDS with this location. If on VTSS *ssssssss* appears, the audit found the VTV on a different VTSS than the one that was audited.

vvvvvv VTV discovered [VTSS *ssssssss]*

The audit found a current version of the VTV on a VTSS whose location was unexpected and updated the CDS with this location. If on VTSS *ssssssss* appears, the audit found the VTV on a different VTSS than the one that was audited.

vvvvvv VTV valid [VTSS *ssssssss]*

The audit found a valid version of the VTV and updated the CDS with this location. If VTSS *ssssssss* appears, the audit found the VTV on a different VTSS than the one that was audited.

*** *vvvvvv* no access to VTSS *ssssssss* ***

The audit found a valid version of the VTV which is on a VTSS that the host cannot access.

MVC *mmmmmm* STATUS CHANGED FROM EXPORT TO CONSOLIDATE VOLUME

The audit discovered current VTVs on an export MVC that was created by export by VTV or Management Class. The audit changed the MVC status from export to consolidate and updated the CDS to add the MVC and its VTVs.

EXPORT MVC *mmmmmm* IS NOW MADE EMPTY IN THE CDS

The audit discovered no current VTVs on an export MVC that was created by export by VTV or Management Class. The audit marked the MVC as empty.

WARNING MVC *mmmmmm* IS AN OUTPUT MVC FROM AN EXPORT OPERATION - FORCING READONLY

The audit forced read-only status on export MVC *mmmmmm*.

Audit terminated. Unable to determine the position of the end of VTV *vvvvvv* on MVC *mmmmmm*

VTCS issued an Inventory MVC ECAM request to determine the position and volser of a VTV on the MVC being audited. VTSS indicated, in the ECAM response, that it was unable to determine the position of the end of the VTV. Because VTCS needs that information to determine the position of the next VTV on the MVC (assuming end of tape has not been reached), VTCS had to terminate the audit with RC=8. The MVC is left in Audit status. To resolve the condition, drain the MVC. If you cannot drain the MVC, contact StorageTek Software Support.

CANCEL

CANcel stops active and queued processes that use an RTD.

Syntax

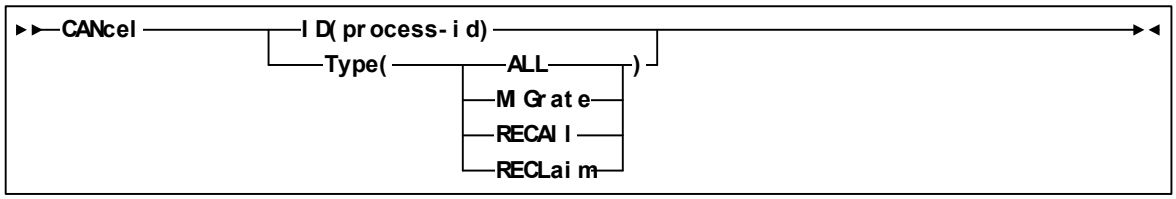


FIGURE 1-6 CANcel syntax

Parameters

ID

specifies a process to cancel.

process–id

the process ID.

Type

specifies the type of process to cancel.

ALL

cancel all processes.

MIGrate

cancel all migration processes.

RECAll

cancel all recall processes.

RECLaim

cancel all reclaim processes.

Interfaces

SWSADMIN utility and VT command.

JCL Requirements

The following are the required and optional statements for the AUDIT JCL:

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the utility report.

SLSIN

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

CONFIG

CONFIG defines or modifies the VSM configuration stored in the HSC CDS.

The following sections show the syntax of the CONFIG utility and of the input statements to CONFIG. You create a single file that contains the CONFIG statement and its input statements.

CONFIG Utility

The CONFIG utility specifies the CDS level and, if required, RESET.

Syntax

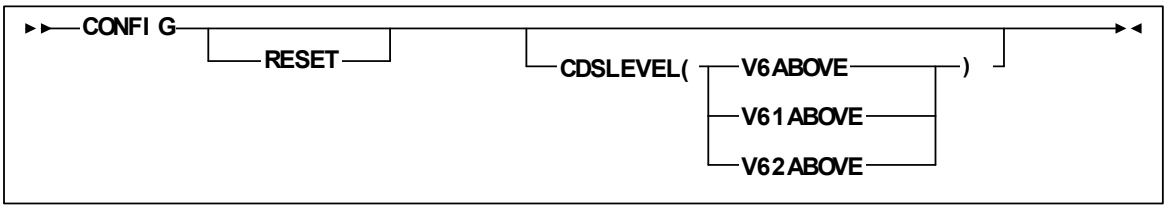


FIGURE 1-7 CONFIG statement syntax

Parameters

RESET

is required as described in TABLE 1-1.

TABLE 1-1 RESET Requirements

CDS is at "F" Level or Above	CDS is below "F"
Required **only** when changing the CDS to a lower level. RESET is not required when going from F to G level. **Also note**, however, that all hosts accessing the CDS must be running VTCS/NCS 6.2 when you convert from F to G level.	You **must** specify RESET when you make hardware changes such as: n Adding a VTSS to your configuration. n Adding RTDs, removing RTDs, or reordering their sequence in your configuration. n Physically removing a VTSS from your configuration. n Changing the CDS level. You **do not** need to specify RESET if you: n Change VSM volumes (such as adding VTVs and MVCs). Note that VTV and MVC volumes cannot be removed by RESET or any other VTCS utility. Removal of VTV and MVC volumes requires the definition of a new CDS and the use of MERGEcds. n Change VSM policies (such as changing AMT values).

Note –

- HSC must be down on all hosts when you run CONFIG RESET. The changes you made to RTD definitions will take effect when you restart HSC.
- Although some changes can be done dynamically by running CONFIG and not restarting HSC/VTCS, there may be some additional parameter changes required to use any additional resources.
- If you run a CONFIG with RESET, then the status flags are reset (and message SLS6746E disappears). If a VTSS was offline when you run the config (with RESET), please remember to audit the VTSS.

CDSLEVEL

One of the following CDS levels:

V6ABOVE

creates an “E” level CDS, which supports the following features:

- Full VSM4 Support
- 4 MVC copies
- 800MB VTVs (see for additional requirements)

V61ABOVE

creates an “F” level CDS, which supports the following features:

- Full VSM4 Support
- 4 MVC copies
- 800MB VTVs (see for additional requirements)
- Near Continuous Operations (NCO)
- Bi-directional clustering

V62ABOVE

creates a “G” level CDS, which supports the following features:

- Full VSM4 Support
- 4 MVC copies
- Near Continuous Operations (NCO)
- Bi-directional clustering
- 400Mb/800Mb/2Gb/4gb VTV sizes
- Standard/Large VTV Pages
- 65000 VTVs per MVC

Interfaces

SWSADMIN utility only.

CONFIG Utility GLOBAL Statement

The CONFIG utility GLOBAL statement specifies VTCS global values. This statement is required.

Syntax

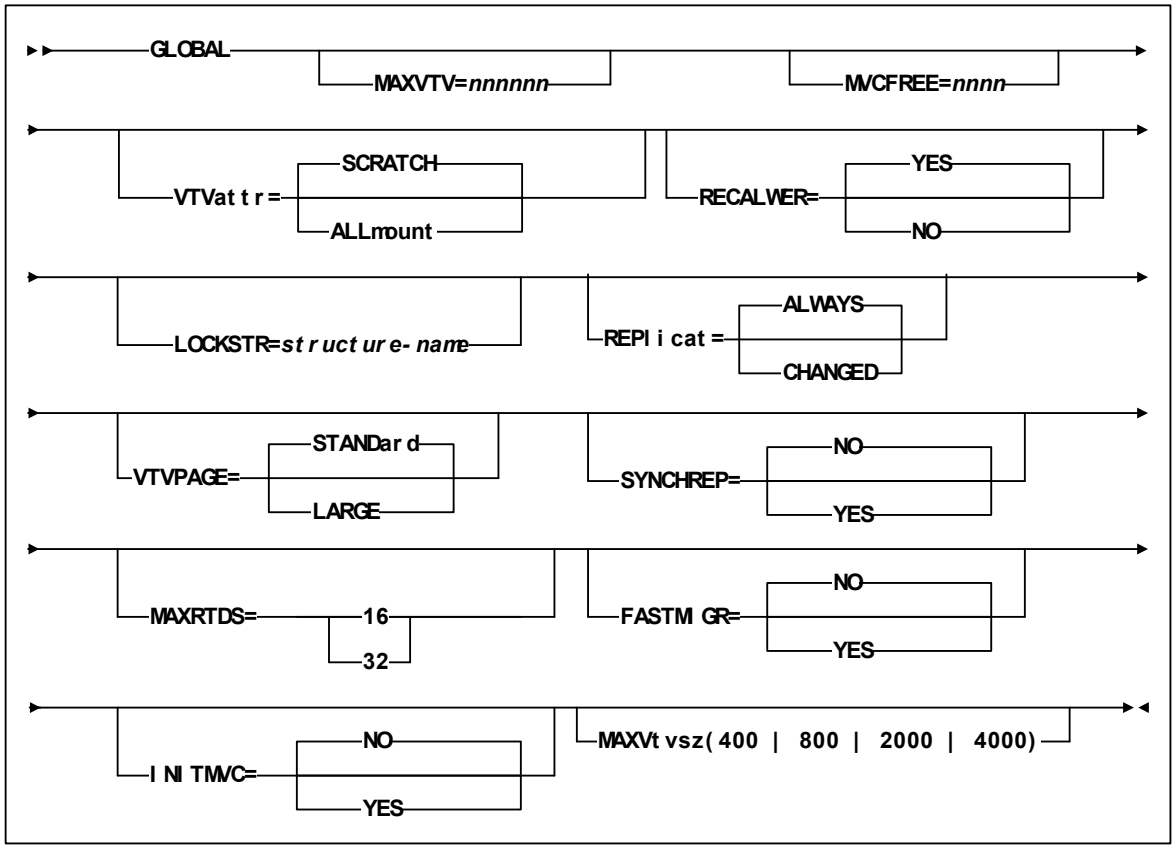


FIGURE 1-8 GLOBAL statement syntax

Parameters

MAXVTV=*nnn*

specifies the maximum number of VTVs that can be migrated to a single MVC. The default is 32000 for a D, E, or F level CDS and 65000 for a G level CDS. Valid values are:

- 4 to 32000 for a D, E or F level CDS.
- 4 to 65000 for a G level CDS.

For more information about CDS levels, see "CONFIG Utility" on page 20.

Note – For compatibility purposes, a hard limit of 32001 VTV copies will be enforced upon a 'G' level CDS if the MAXVTV parameter has a value of <=32000. This is to automatically inhibit the generation of MVCs that will cause problems with regressing the CDS back to a D, E or F level.

MVCFREE=*nnn*

specifies the minimum number of free MVCs in the MVC pool. A free MVC has 100% usable space and does not contain any migrated VTVs. Valid values are 0 to 255. The default is 40.

If free MVCs is equal or less than this value, VTCS issues message SLS6616I and starts an automatic space reclamation.

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

VTVattr=SCRATCH | ALLmount

specifies when VTCS assigns a Management Class to a VTV.

SCRATCH

Assign a Management Class only when VTCS does a scratch mount of the VTV (the default).

ALLmount

Assign a Management Class whenever VTCS mounts the VTV.

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

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

RECALWER

specifies whether VTCS recalls VTVs with read data checks (applies to recall and drain operations).

YES

recall VTVs with read data checks (the default).

NO

Do not recall VTVs with read data checks.

LOCKSTR=*structure-name*

specifies the Coupling Facility Structure that holds VTCS Lock Data. The Structure Name must be 16 characters or less and conform to IBM's standard for naming Coupling Facility Structures. For more information, see *Installing ELS*.

Caution – CONFIG RESET **is required** to add LOCKSTR=*structure-name* to a CDS that did not previously use LOCKSTR=*structure-name* and to remove LOCKSTR=*structure-name* from a CDS. CONFIG RESET is **not** required to change lock structure names (for example, going from LOCKSTR=VTCSL1 to LOCKSTR=VTCSL2).

REPlicat

specifies when VSM replicates the VTV.

ALWAYS

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

CHANGED

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

Was changed while it was mounted **or**

Was only read while mounted but less than the expected number of MVC copies of the VTV exist.

Note – Regardless of the CONFIG GLOBAL REPlicat setting, replication **also** requires that:

- The VTV must be dismounted in a VTSS that supports replication **and** there cannot be an identical copy of the VTV in the other VTSS in the Cluster.
- In addition to the CONFIG GLOBAL REPlicat value, you **must** specify REPlicat(YES) on a VTV's Management Class for replication to occur. For more information, see"REPlicat" on page 241.

VTVPAGE

specifies that the page size used to store VTV data in the VTSS and on the MVCs. This setting only applies to 400 and 800 MB VTVs.

STANDard

Standard page size, which is compatible with all VSM3/VSM4 models and microcode levels.

LARGE

Large page size, which can provide improved performance within the VTSS and for migrates and recalls. Large page size requires a G level CDS. For more information on CDS levels, see "CONFIG Utility" on page 20. For 2 and 4 GB VTVs (`MAXVtvsz` 2000 or 4000) a `VTVPAGE` setting of `LARGE` is always used.

Note –

- `VTVPAGE` **does not** apply to VSM2s. `VTVPAGE(LARGE)` requires VSM4/VSM5 microcode D02.02.00.00 or VSM3 microcode N01.00.77.00. No installed option is required.
- `MGMTCLAS VTVPAGE`, if specified, overrides the `CONFIG GLOBAL VTVPAGE` value. If `VTVPAGE` is not specified on either the `MGMTclas` statement or the `CONFIG GLOBAL` statement, the default is `STANDard`.

Caution –

- The page size of a VTV can only be changed by a VTV scratch mount. Additional restrictions may also apply for scratch VTVs that were previously resident in a VTSS.
- If you specify `LARGE` and the CDS level and/or VTSS microcode **do not** support `LARGE`, VTCS issues warning messages and defaults to `STANDard`.
- If you specify `STANDard` for 2 or 4 GB VTVs VTCS issues warning messages and defaults to `LARGE`.
- Creating VTVs with large pages makes these VTVs **unreadable** in configurations that do not support large VTV pages.

SYNCHREP

specifies whether VTV synchronous replication feature is enabled.

NO

Synchronous replication is not enabled (the default).

YES

Synchronous replication is enabled.

Note – SYNCHREP=YES merely enables synchronous replication. To actually implement synchronous replication, you must create a Management Class that specifies REPLicat= YES_SYNC. For more information, see"Syntax - Advanced Management Feature" on page 238.

MAXRTDS

specifies the maximum number of RTDs supported.

16

up to 16 RTDs supported.

32

up to 32 RTDs supported.

Note – MAXRTDS = 32 has the prerequisites described in TABLE P-8 on page viii.

FASTMIGR

Specifies if the stacked migrates feature is enabled for all VTSSs that support this feature.

YES

Enable stacked migrates.

No

Disable stacked migrates (the default).

Note – FASTMIGR=YES has the prerequisites described in TABLE P-8 on page viii. For this feature to be enabled, **all hosts** must be running the prerequisites, otherwise:

- If a host is active that does not support or tolerate stacked migrates, this will cause the CONFIG utility to return an error.
- If a host is started and does not tolerate or support this feature, the host will shut down.

-

INITMVC

specifies whether uninitialized MVCs in the named MVC Pool are initialized when first mounted on an RTD.

NO

Uninitialized MVCs are not initialized.

YES

Uninitialized MVCs are initialized.

Note –

- MVCPOOL INITMVC overrides GLOBAL INITMVC. There is no default for MVCPOOL INITMVC; if not specified for a named MVC Pool the CONFIG GLOBAL value (or default) is used.
- Initialization of MVCs in the DEFAULTPOOL is controlled by the GLOBAL INITMVC specification (or default).
- MVC Initialization applies only to VSM4/5 and requires microcode level D02.05.00.00 or higher. If this level of microcode is not installed on all VTSSs in the configuration, MVC initialization will be limited to the VTSSs that have it installed.

MAXVtvsz

specifies the maximum VTV size. Valid values for this parameter depend on both the CDS level and the microcode levels of the applicable VTSSs.

400

400MB (the default, if not specified).

800

800MB. The CDS must be at a E level or above.

2000

2GB. The CDS must be at a G level or above.

4000

4GB. The CDS must be at a G level or above.

Note –

- The size of a VTV changes *only* after it goes through a scratch cycle. Therefore, if you change the Management Class and DISP=MOD, then it will still retain the original size.
- If you specify a VTV size that is not supported by the configuration, VTCS issues warning messages and `MAXVtvsz` defaults to the largest VTV size supported by the configuration.
- `MAXVtvsz` **does not** apply to VSM2s.
- `MAXVTVSZ(2000 | 4000)` requires VSM4/VSM5 microcode D02.02.00.00 or VSM3 microcode N01.00.77.00. No installed option is required.

CONFIG Utility RECLAIM Statement

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

Syntax

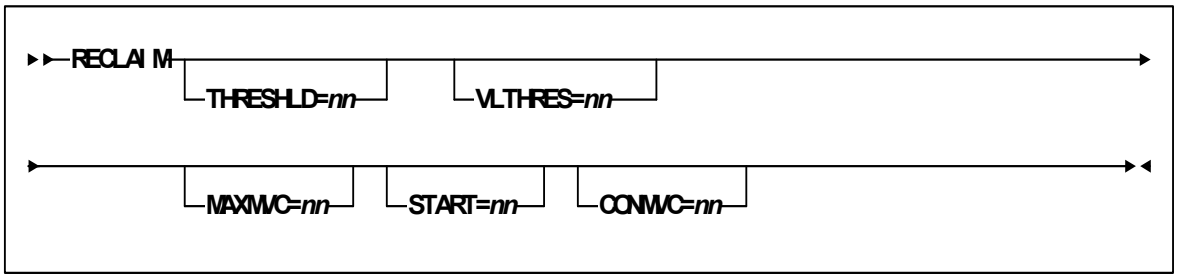


FIGURE 1-9 RECLAIM statement syntax

Parameters

THRESHLD=*mm*

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*

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*

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:

(*Reclaim Candidates*/*Reclaim Candidates* + *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*

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 Utility VTVVOL Statement

The CONFIG utility VTVVOL statement defines a range of VTVs. This statement is required.

You can only add new VTV ranges. A range can consist of a single volume. You cannot delete or modify existing ranges. You can, however, respecify existing VTV ranges when you add new ranges (for example, by adding new VTV ranges to the output of the DECOM utility).

Note – The following restrictions when you respecify existing ranges:

- If you respecify any existing range, you must respecify all existing ranges.
- The high and low volume serial numbers of each respecified range must exactly match the previously specified range.
- The volume type for each respecified range must be the same as the original specification (MVC or VTV).
- Each range can be respecified only once.

Syntax

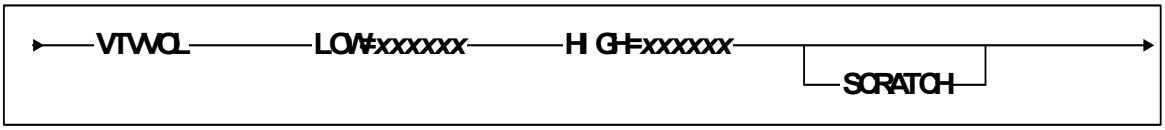


FIGURE 1-10 VTVVOL statement syntax

Parameters

LOW=*xxxxxx*

specifies the start of a range of VTVs.

HIGH=*xxxxxx*

specifies the end of a range of VTVs.

Note – The incremental part of a range cannot exceed 5 characters for numeric and 4 characters for alpha ranges.

SCRATCH

specifies that the VTVs added to the CDS are placed in scratch status, which is not the default for the VTVVOL parameter.

Note – If you are using the ExLM SYNCVTV function for VTV scratch synchronization, StorageTek recommends that you define VTV ranges in scratch status. If you do not, you must use the HSC SLUADMIN utility to scratch these volumes.

CONFIG Utility MVCVOL Statement

The MVCVOL statement defines a range of MVCs available to VTCS. This statement is required.

You can only add new MVC ranges. A range can consist of a single volume. You cannot delete or modify existing ranges. You can, however, respecify existing MVC ranges when you add new ranges (for example, by adding new MVC ranges to the output of the DECOM utility).

Note – The following restrictions when you respecify existing ranges:

- If you respecify any existing range, you must respecify all existing ranges.
- The high and low volume serial numbers of each respecified range must exactly match the previously specified range.
- The volume type for each respecified range must be the same as the original specification (MVC or VTV).
- Each range can be respecified only once.

Syntax

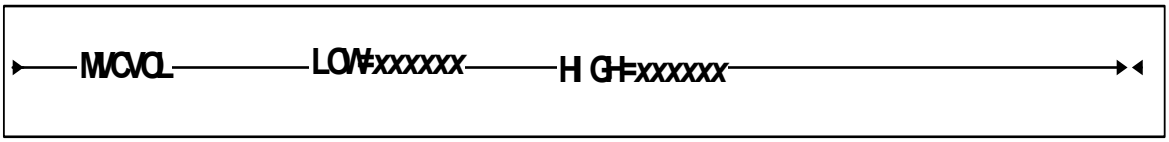


FIGURE 1-11 MVCVOL statement syntax

Parameters

LOW=*xxxxxx*

specifies the start of a range of MVCs.

HIGH=*xxxxxx*

specifies the end of a range of MVCs.

CONFIG Utility VTSS Statement

The CONFIG utility VTSS statement defines a VTSS and sets its operating values. This statement is required.

When you define a new VTSS, place its definition after any existing VTSS definitions, which must remain in their original order.

Syntax

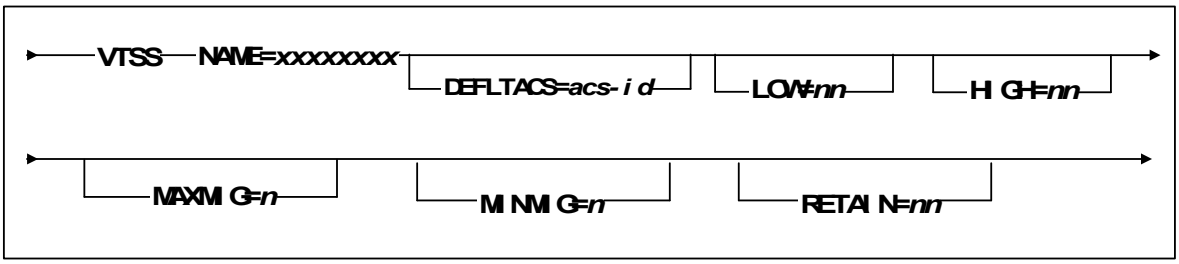


FIGURE 1-12 VTSS statement syntax

Parameters

Note – If you physically remove a VTSS from your configuration, reconfigure the VTSS with a VTSS statement only and no parameters.

NAME=*xxxxxxxx*

specifies the VTSS name. This parameter is required; there is no default value.

Caution – Note the following:

- The VTSS name can consist of the characters "A-Z", "0-9", "@", "$", and "#".
- You specify the VTSS name *only* via the NAME parameter, which sets the VTSS name in both the VTSS microcode (as displayed in the Subsystem Name field in the LOP or VOP) and in the configuration area of the HSC CDS. After VSM is put into operation, the VTSS name is also stored in each VTV record in the CDS. Each VTV record contains the VTSS name on which that VTV is resident or, if the VTV is migrated, the VTV record contains the VTSS name from which the VTV was migrated.
- Once you set the VTSS name via the NAME parameter, you *cannot* change this identifier in the HSC CDS. That is, the CONFIG utility *will not* let you change the NAME parameter after an initial setting and changing the VTSS name using the Subsystem Name field of the LOP or VOP *cannot* change the VTSS name in the HSC CDS.
- It is especially critical that you *do not* attempt to rename a VTSS that contains data on VTVs, which includes VTSS-resident VTVs and migrated VTVs!
- For an initial setting *only* (not a change), you can set the VTSS name in the NAME parameter only if the VTSS name value in the VTSS microcode is:
  - The factory setting (all blanks).

- A value of 99999999 (eight 9s).

- Therefore, for an initial setting *only,* if the name in the VTSS microcode *is not* all blanks or 99999999, your StorageTek hardware representative must use the VTSS LOP or VOP to set the VTSS name to 99999999 so you can set the VTSS name to the value you want via the NAME parameter.

Caution –

DEFLTACS=*acs-id*

VTCS supports multi-VTSS confirmations, and supports connecting multiple ACSs to each VTSS. In configurations where a VTSS is connected to multiple ACSs, you can use the DEFLTACS parameter to specify the default ACS from which MVCs will be selected for migration, consolidation, and reclaim processing.

Note that the actual number of ACSs to which you can migrate a VTV is limited by whether you have Basic or Advanced Management Feature specified (for more information, "FEATURES Control Statement" on page 229:

- With the Basic Management Feature, you are limited to duplexing VTVs to two separate ACSs (per MGMTclas statement) via the MGMTclas ACSlist and DUPlex parameters (and ignores the DEFLTACS parameter, as described below). For more information, see "Parameters - Basic Management Feature" on page 235.
- With the Advanced Management Feature, you can migrate up to 4 VTV copies to separate ACSs (per MGMTclas statement) via the MGMTclas MIGpol parameter (and ignores the DEFLTACS parameter, as described below). For more information, see "Additional Parameters - Advanced Management Feature" on page 239.

If you do not specify DEFLTACS, the default value is x'FF', which allows VTCS to select MVCs from any ACS.

Note – VTCS ignores the value on the DEFLTACS parameter if you specify the DEFLTACS parameter and do either of the following:

- Specify the ACSlist parameter of the MGMTclas statement.
- Use a Storage Class.

LOW=*nn*

specifies the low automatic migration threshold (LAMT) for this VTSS.

Valid values are 5 to 95 and must be less than the HIGH default threshold. The default is 70.

HIGH=*nn*

specifies the high automatic migration threshold (HAMT) for this VTSS.

Valid values are 6 to 95 and must be greater than the LOW default threshold. The default is 80.

MAXMIG=*n*

specifies the maximum number of concurrent automatic migration, immediate migration, and migrate-to-threshold tasks for this VTSS.

Valid values are 1 to the number of RTDs on the VTSS. See also "MAXRTDS" on page 25. The default is half the number of RTDs attached to the VTSS.

MINMIG=*n*

specifies the minimum number of concurrent automatic migration, immediate migration, and migrate-to-threshold tasks for this VTSS.

Valid values are 1 to the MAXMIG setting. The default is 1 task.

RETAIN=*nn*

specifies the number of minutes that VTCS will retain an MVC on an RTD in idle mode after a migration or recall. Retaining the MVC can reduce MVC mounts.

Valid values are 1 to 60. The default is 10.

CONFIG Utility RTD Statement

The VTCS `CONFIG RTD` statement defines the devices (RTDs or ELS appliances) 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 ELS appliances.

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

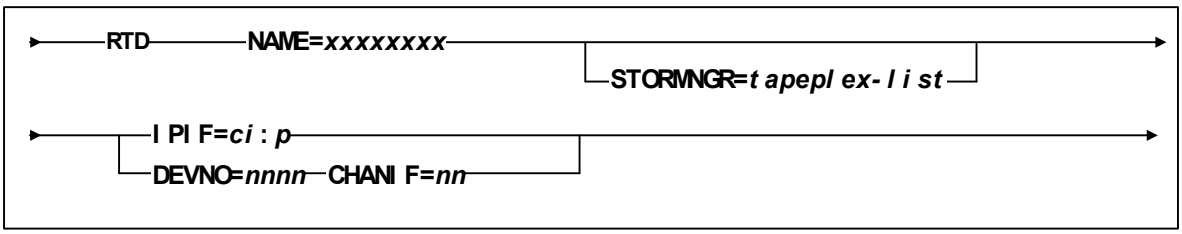


FIGURE 1-13 RTD statement syntax

Parameters

NAME=*xxxxxxxx*

specifies the 1 to 8 character identifier of the RTD or ELS appliances. 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 ELS appliances, use any meaningful 1 to 8 character identifier. Unlike RTDs, ELS appliances, which do not use the DEVNO parameter, do not have actual MVS unit addresses. VTCS, however, generates a unit address for ELS appliances which you may want to use when specifying a NAME value for a ELS appliance.

VTCS generates the unit address in the form V*xxi*

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 1-2.

TABLE 1-2 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 ELS appliance.

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 ELS appliance (that is, when the IPIF parameter is specified).

IPIF=*ci:p*

specifies the IP interface on the VTSS IFF3 card that communicates with the ELS appliance. 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 ELS appliance, 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 ELS appliance.

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 ELS appliance.

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.

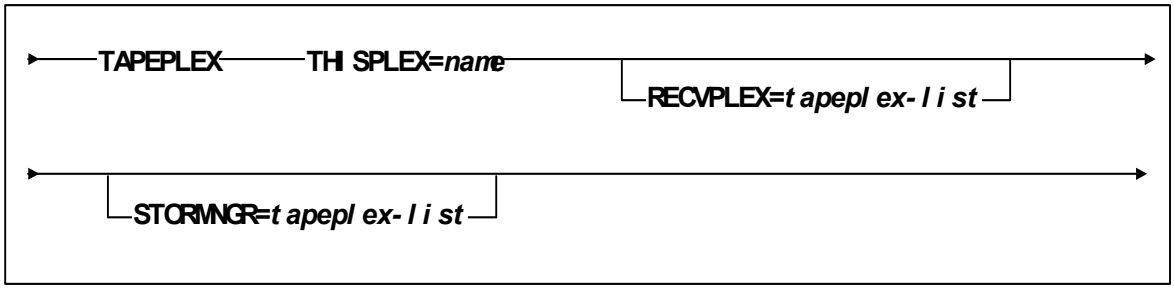
CONFIG Utility TAPEPLEX Statement

The VTCS `CONFIG TAPEPLEX` statement defines values for replication to a VLE appliance.

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



Parameters

THISPLEX

specifies the TapePlex name for this configuration.

name

the TapePlex name. This name must match the name specified on an SMC `LIBrary` or `TAPEPlex` command for this TapePlex's SMC.

STORMNGR

optionally, specifies Subsystem Names of the VLE appliances attached to this TapePlex.

stormngr-list

the list of Subsystem names.

CONFIG Utility VTD Statement

The CONFIG utility VTD statement defines the MVS unit address range of the VTDs in a VTSS. This statement is required and must follow the VTSS statement where the VTDs reside.

Note – VSM2s and VSM3s provide 64 VTDs per VTSS. VSM4s provide 256 VTDs per VTSS.

You can specify the VTD unit addresses to either apply to all hosts or to define which VTDs are available to specific hosts; for more information, see "Specifying VTD Unit Addresses" on page 46.

Syntax

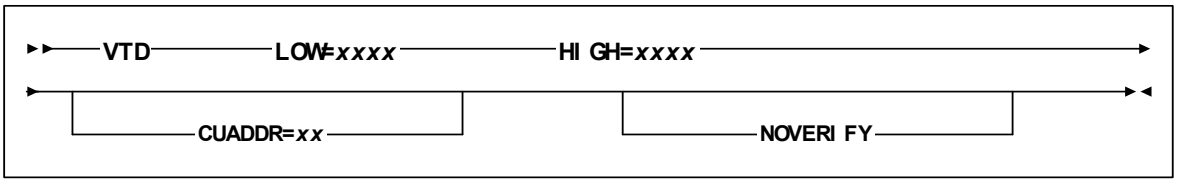


FIGURE 1-14 VTD statement syntax

Parameters

LOW=*xxxx*

specifies a four character valid MVS unit address as the start of a range of VTDs.

HIGH=*xxxx*

specifies a four character valid MVS unit address as the end of a range of VTDs.

CUADDR=*xx*

specifies a control unit identifier that matches its value in the IOCP. Valid values are 0 through 15 for VSM4 and 0 through 3 for VSM2/3. This statement is required only if a partial VTD range is specified for a host and the host does not have a path to the VTDs.

NOVERIFY

Normally, VTCS will attempt the verification of all predicted VTD identifiers associated with Virtual Tape Drives. In order to do this each MVS I/O address must be available in order that the associated Host may issue the ECAM-T request Virtual_Device_Identify.

In specific cases where, for example, VTCS is providing support for a remote client Host connected via MVS/CSC, the local host, acting as a server for the remote client, may not have paths to the MVS I/O address. In these cases, any attempt at verification of the predicted VTD identifier will fail and will result in error messages posted to the operator. NOVERIFY suppresses verification and prevents these failures.

The parameter indicates to VTCS that the VTD should be considered as existing, even though it may not be defined in MVS (or not be defined as a tape device). Although VTCS will then advertise its availability back to SMC (it could be accessible from another host), it will not perform any I/O to the device itself.

CONFIG Utility CLUSTER Statement

The CONFIG utility CLUSTER statement defines two VTSSs in a Cluster.

Syntax

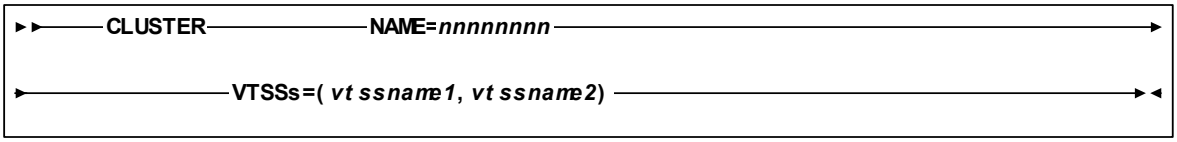


FIGURE 1-15 CLUSTER statement syntax

Parameters

NAME=*nnnnnnnn*

specifies the 1 to 8 character identifier of the Cluster.

This parameter is required; there is no default value.

VTSSs=(*vtssname1,vtssname2)*

Specifies the names of two VTSSs in a cluster. The relationship between the two VTSS is defined by the CONFIG CLINK statement; for more information, see "CONFIG Utility CLINK Statement" on page 43.

There is no default value. This parameter replaces and is mutually exclusive with the PRIMARY and SECONDRY parameters.

Note –

- The VTSS names that you specify on a CLUSTER statement must be known to VTCS. That is, the VTSS names must be already defined in the CDS or must be specified in VTSS statements that already exist in the current set of CONFIG statements. For, example, the following is valid:

  ```
  VTSSNAME=VTSS1  LOW=70 HIGH=80  MAXMIG=3  RETAIN=5
  VTSSNAME=VTSS2  LOW=70 HIGH=80  MAXMIG=3  RETAIN=5
  CLUSTER NAME=CLUSTER1 VTSSs=(VTSS1,VTSS2)
  ```

The following is **not** valid unless VTSS1 and VTSS2 are already defined in the CDS:

```
CLUSTER NAME=CLUSTER1 VTSSs=(VTSS1,VTSS2)
VTSSNAME=VTSS1  LOW=70 HIGH=80  MAXMIG=3  RETAIN=5
VTSSNAME=VTSS2  LOW=70 HIGH=80  MAXMIG=3  RETAIN=5
```

- Also note that the DECOM utility outputs CLUSTER statements with blanks instead of commas as separators, which is also valid input to CONFIG. For example, DECOM outputs the following, which is valid input to CONFIG:

  ```
  CLUSTER NAME=CLUSTER1 VTSSs=(VTSS1 VTSS2)
  ```

CONFIG Utility CLINK Statement

The CONFIG utility CLINK statement defines the channel interface between two VTSSs in a Cluster. The VTSSs can be in one of two modes:

- Primary-Secondary, in which case you write CLINK statements for *only* the Primary.
- Peer-to-Peer, in which case you write CLINK statements for both VTSSs to enable bi-directional VTV replication.

For examples, see *Beyond the Basics: VTCS Leading Edge Techniques*.

Syntax

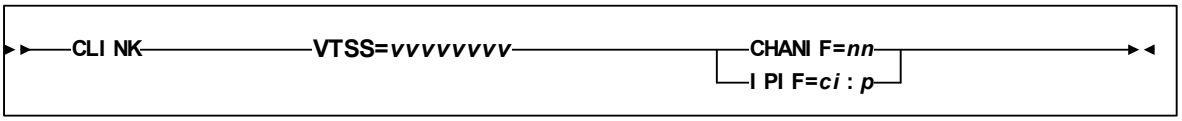


FIGURE 1-16 CLINK statement syntax

Parameters

VTSS=*vvvvvvvv*

specifies the name of the one VTSS in a Cluster.

This parameter is required; there is no default value.

CHANIF=*nn*

specifies the channel interface for communication between two VTSSs in a Cluster. This value must match the Nearlink channel interface defined at the VTSS LOP by your StorageTek hardware representative at VTSS installation and configuration.

This parameter is required; there is no default value.

The value must be two characters in length and have a value from 0A to 1P. The first digit is the VTSS Cluster ID (valid values are 0 or 1). The second digit is the group or adapter ID (valid values are A to P).

Caution – Do not use the LINK number shown at the LOP instead of the VTSS Cluster ID for the first character of the CHANIF value!

Note – The VTSS name that you specify on a CLINK statement must be known to VTCS. That is, the VTSS name must be already defined in the CDS or it must be specified in a VTSS statement that already exists in the current set of CONFIG statements. For, example, the following is valid:

```
VTSSNAME=VTSS1  LOW=70 HIGH=80  MAXMIG=3  RETAIN=5
CLINK VTSS=VTSS1 CHANIF=0A
```

The following is **not** valid unless VTSS1 is already defined in the CDS:

```
CLINK VTSS=VTSS1 CHANIF=0A
VTSSNAME=VTSS1  LOW=70 HIGH=80  MAXMIG=3  RETAIN=5
```

CLINK IPIF=*ci:p*

specifies Ethernet connection of two Native IP (IFF3 card) ports, where:

c

is the VTSS cluster number (0 or 1)

i

is the IP interface letter (A or I)

p

is the path on the interface (0 to 3)

Note –

- The CLINK statement must contain either the `CHANIF` or the `IPIF` parameter, but not both.
- The *ci:p* values are shown on the VTSS IFF IP Configuration Status screen for each IFF Ethernet port.

CONFIG Utility HOST Statement

The CONFIG utility HOST statement is an optional statement that defines an MVS host and, optionally, the NOMIGRAT and/or NORECLAM parameters.

Note the following:

- If specified, the HOST statement must follow the VTSS statement for the VTSS attached to that host.
- You must either specify all host definitions or none; if you specify only some of the hosts attached to a VTSS, VTCS will issue an error.

Syntax

FIGURE 1-17 HOST statement syntax

Parameters

NAME=*xxxx*

specifies the LIBGENed hostname.

NOMIGRAT

specifies that this host cannot do migrations, consolidations, or export by VTV or Management Class from the VTSS(s) that the host accesses. NOMIGRAT controls both automatic and demand migrations and consolidations. This parameter is optional.

Note – Specifying NOMIGRAT also causes NORECLAM to be set.

IMMEDmig KEEP and IMMEDmig DELETE are mutually exclusive with CONFIG HOST NOMIGRAT. If you specify both, the IMMEDmig value overrides NOMIGRAT, and VTCS does not issue a message about this override.

NORECLAM

specifies that this host cannot initiate automatic or demand reclaim processing using the VTSS(s) that the host accesses (the host can still do demand MVC drains via MVCDRain). This parameter is optional.

Specifying VTD Unit Addresses

You can specify VTD addresses by doing one of the following:

- Specify the VTD unit addresses on a VTD statement following a VTSS statement and do *not* specify any HOST statements following the VTSS statement. All hosts physically connected to the VTSS have access to its VTDs by the default addresses specified on the VTD statement.
- Do *not* specify the VTD unit addresses on the VTD statement following a VTSS statement. Instead, place a VTD statement after a HOST statement for only those hosts for which you want to define connections to the previously defined VTSS. You must specify a placeholder (HOST NAME with no VTD parameter) for any hosts that you do not want connected to this VTSS.

  Note that the VTVs created and MVCs initially written to from a VTSS are considered that VTSS's resources, so only hosts with access to a VTSS also have access to its VTVs and MVCs. In this type of "restricted" access configuration, therefore, each host should have a separate VTV scratch pool to ensure that each host has accurate scratch counts. Similarly, free MVCs and MVC reclaim counts are reported on each host for the MVCs associated with the VTSS to which the host is connected.

  You can specify different address ranges for each host, although StorageTek recommends that you specify the same address ranges for all hosts for consistency of operations. If you specify different address ranges for different hosts, use the HSC SET DRVHOST and SMC DRIVEMAP statements if you have a client/server configuration. **Also note that** you must include UNITATTR MODEL(IGNORE) commands for ALL overgenned devices (including VTDs) on all MVS hosts.

Caution – In a multi-host, multi-VTSS configuration, you can use this VTD addressing method to deny access to VTSSs to which hosts are physically connected. You **must**, however, use this method to deny access from hosts that are *not* physically connected to a VTSS. If you do not deny access, VTCS on a host that does not have physical connections to a VTSS may wait trying to communicate with the VTSS while VSM operations may be stalled on all other hosts.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the utility report.

SLSIN

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

CONSolid

CONSolid consolidates VTVs on MVCs.

Syntax

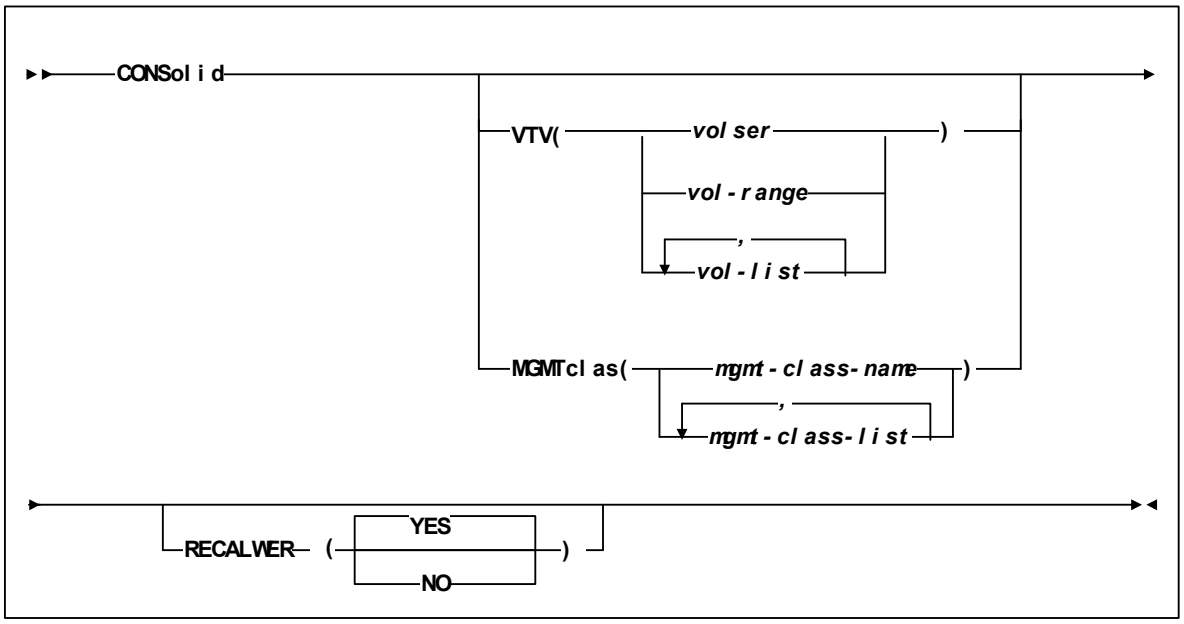


FIGURE 1-18 CONSolid utility syntax

Parameters

VTV

specifies one or more VTVs to consolidate.

volser, vol-range or *vol-list*

the volsers of one or more VTVs. You can specify a maximum of 2,000 VTVs.

MGMTclas

specifies the names of one or more Management Classes that determine the VTVs to consolidate.

mgmt-class-name | *mgmt-class-list*

the names of one or more Management Classes that you defined on the MGMTclas control statement; for more information, see "MGMTCLAS Control Statement" on page 234. You can consolidate a maximum of 2,000 VTVs by specifying a Management Class.

RECALWER

specifies whether VTCS recalls VTVs with read data checks.

YES

recall VTVs with read data checks (the default).

NO

Do not recall VTVs with read data checks.

Interfaces

SWSADMIN utility only.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the utility report.

SLSIN

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

Consolidation Reports

The consolidation report displays the following messages:

MIGRATE ONLY FROM VTSS v*tssname*

The VTV is resident on VTSS *vtssname*.

REMIGRATE FROM MVC *mvcname* VIA VTSS *vtssname*

VTCS is recalling a VTV from MVC *mvcname* to consolidate the VTV.

VTV *vtvname* NOT SELECTED; VTV IS SCRATCH

VTCS will not consolidate the specified VTV, which is either scratch or not initialized.

VTV *vtvname* NOT SELECTED; VTV ALREADY CONSOLIDATED

The specified VTV is already consolidated.

VTV *vtvname* NOT SELECTED; VTV RECORD NOT FOUND

VTCS will not consolidate the specified VTV, which has no record in the CDS.

VTV *vtvname* NOT SELECTED; VTV STILL MOUNTED ON DRIVE

VTCS cannot consolidate the specified VTV, which is mounted or in recovery.

REDRIVING REQUEST BECAUSE OF ERROR

VTCS is retrying an unsuccessful consolidation request.

CONSOLID CMD PROBLEM DECODING VCI REQUEST FROM HSC

The consolidation failed.

VTV *vtvnumber* NOT SELECTED: LIMITED ACCESS TO VTSS

The consolidation request failed because a host not enabled for consolidation (via the NOMIGRAT parameter) issued the request.

MIGRATE NO MVCS AVAILABLE

Sufficient free MVCs are not available to complete the request.

DECOM

DECOM lists the VSM configuration information in the HSC CDS.

Syntax

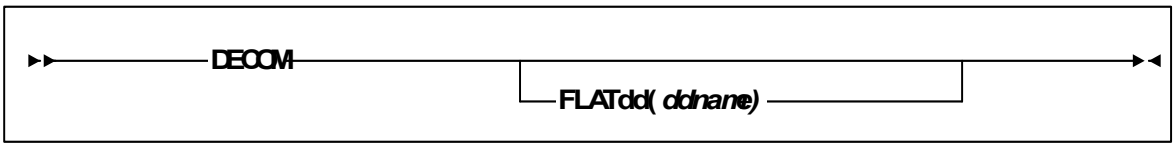


FIGURE 1-19 DECOM utility syntax

Parameters

FLATdd

specifies the output destination ddname if a flat file is required (must be RECFM= FB,LRECL=80).

ddname

the ddname of the flat file included in the JCL.

Interfaces

SWSADMIN utility only.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the utility report.

SLSIN

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

DECOM Output

FLATDD Parameter Output

FIGURE 1-20 shows an example of DECOM output with the FLATDD parameter.

```
 CONFIG CDSLEVEL(V62ABOVE)
GLOBALMAXVTV=32000 MVCFREE=40 VTVattr=ALLmount RECALWER=YES
REPlicat=ALWAYS LOCKSTR=STK_VTCS_LOCKS
 RECLAIMTHRESHLD=70 MAXMVC=40  START=35
 VTVVOL LOW=905000 HIGH=999999 SCRATCH
 VTVVOL LOW=C00000 HIGH=C25000 SCRATCH
 VTVVOL LOW=RMM000 HIGH=RMM020 SCRATCH
 MVCVOL LOW=N25980 HIGH=N25989
 MVCVOL LOW=N35000 HIGH=N35999
 VTSSNAME=VTSS1  LOW=70 HIGH=80  MAXMIG=3  RETAIN=5
 RTDNAME=VTS18800 DEVNO=8800 CHANIF=0A
 RTDNAME=VTS18801 DEVNO=8801 CHANIF=0I
 RTDNAME=VTS18802 DEVNO=8802 CHANIF=1A
 RTDNAME=VTS18803 DEVNO=8803 CHANIF=1I
 VTDLOW=8900 HIGH=893F
 VTSSNAME=VTSS2  LOW=70 HIGH=80  MAXMIG=3  RETAIN=5
 RTDNAME=VTS28804 DEVNO=8804 CHANIF=0A
 RTDNAME=VTS28805 DEVNO=8805 CHANIF=0I
 RTDNAME=VTS28806 DEVNO=8806 CHANIF=1A
 RTDNAME=VTS28807 DEVNO=8807 CHANIF=1I
 VTDLOW=9900 HIGH=993F
```

FIGURE 1-20 Example DECOM FLATDD output

SLSPRINT Output

FIGURE 1-21 shows an example of DECOM output to SLSPRINT.

```
SWSADMIN (6.2.0)STORAGETEK VTCS SYTEM UTILITYPAGE 0002
TIME 09:07:06 CONTROL CARD IMAGE LISTING
DECOM
SLS1315I SSRDMP.P775644.TESTCDS WAS SELECTED AS THE PRIMARY CONTROL DATA SET
TIME 09:07:06 VTCS DECOMPILE
CONFIG CDSLEVEL(V62ABOVE)
GLOBALMAXVTV=32000 MVCFREE=40 VTVattr=ALLmount RECALWER=YES
REPlicat=ALWAYS LOCKSTR=STK_VTCS_LOCKS
RECLAIMTHRESHLD=70 MAXMVC=40  START=35
VTVVOL LOW=905000 HIGH=999999 SCRATCH
VTVVOL LOW=C00000 HIGH=C25000 SCRATCH
VTVVOL LOW=RMM000 HIGH=RMM020 SCRATCH
MVCVOL LOW=N25980 HIGH=N25989
MVCVOL LOW=N35000 HIGH=N35999
VTSSNAME=VTSS1  LOW=70 HIGH=80  MAXMIG=3  RETAIN=5
RTDNAME=VTS18800 DEVNO=8800 CHANIF=0A
RTDNAME=VTS18801 DEVNO=8801 CHANIF=0I
RTDNAME=VTS18802 DEVNO=8802 CHANIF=1A
RTDNAME=VTS18803 DEVNO=8803 CHANIF=1I
VTDLOW=8900 HIGH=893F
VTSSNAME=VTSS2  LOW=70 HIGH=80  MAXMIG=3  RETAIN=5
RTDNAME=VTS28804 DEVNO=8804 CHANIF=0A
RTDNAME=VTS28805 DEVNO=8805 CHANIF=0I
RTDNAME=VTS28806 DEVNO=8806 CHANIF=1A
RTDNAME=VTS28807 DEVNO=8807 CHANIF=1I
VTDLOW=9900 HIGH=993F
```

FIGURE 1-21 Example DECOM SLSPRINT output

DELETSCR

DELETSCR deletes scratch VTVs from VTSSs and unlinks any migrated VTVs from MVCs.

Syntax

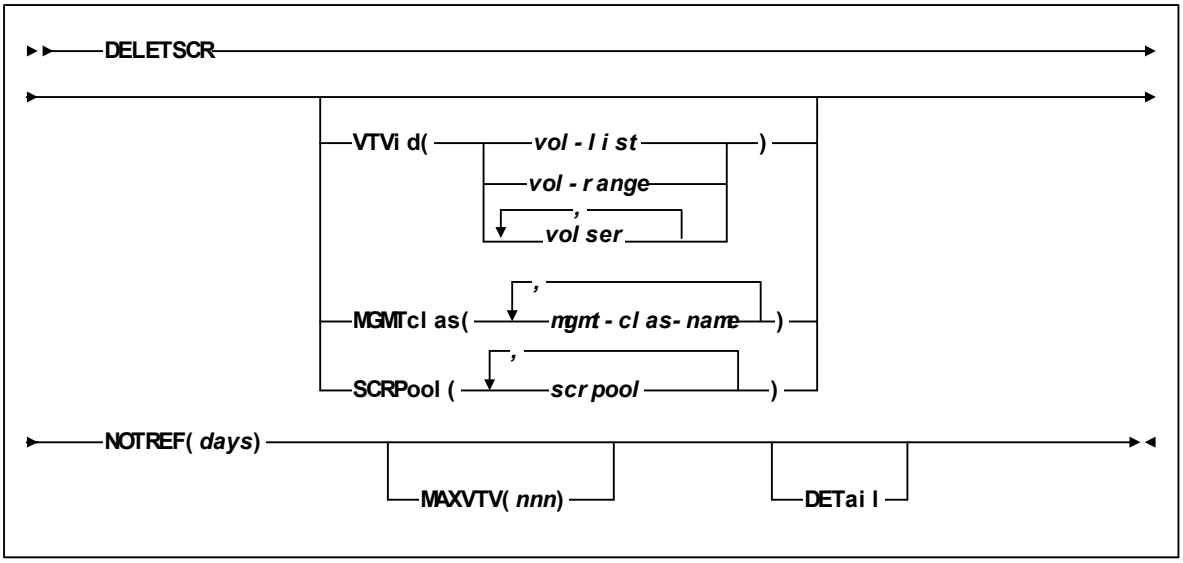


FIGURE 1-22 DELETSCR syntax

Parameters

VTVid

specifies the VTVs.

volser, vol-range or *vol-list*

the volsers of one or more VTVs.

MGMTclas

the Management Class(es) of the VTVs.

mgmt-class-name

the Management Class name that you specified on the MGMTclas control statement.

SCRPool

specifies the scratch pool(s) to be deleted. If the pool contains VTVs and real volumes, DELETSCR deletes only VTVs.

scrpool

an existing HSC scratch pool.

Note – VTVid, MGMTclas, and SCRPool are mutually exclusive. If you do not specify one of these parameters, VTCS processes **all** scratch VTVs.

NOTREF

specifies the number of days since a VTV was last referenced for it to be deleted by DELETSCR.

days

the number of days (1-999).

MAXVTV

specifies the maximum number of VTVs that DELETSCR deletes. Note that this is a maximum, not a target.

nnn

the maximum number of VTVs (0-999). If not specified, DELETSCR deletes all scratch VTVs. If you specify 0, DELETSCR does not delete any VTVs, but the **summary** report shows **how many** VTVs would have been deleted...at the point at which you ran DELETSCR (that is, the report is just a snapshot).

DETail

Produce a detailed report that also shows detail and summary of VTVs that fall within the NOTREF period.

Interfaces

SWSADMIN utility only.

Optional and Required JCL

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the utility report.

SLSIN

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

DELETSCR Report

FIGURE 1-23 shows an example of a DELETSCR report for the following batch invocation:

DELETSCR MGMTCLAS(MC1) NOTREF(60) MAXVTV(10) DET

```
SWSADMIN (6.2.0)          STORAGETEK VTCS SYTEM UTILITY          PAGE 0002
TIME 06:32:03                  SCRATCH VTV DELETE                DATE 2006-10-19
SLS6833I VTV VTV100 deleted from MVC M00001

SLS6835I VTV VTV101 excluded – referenced within 60 days
SLS6833I VTV VTV102 deleted from MVC M00003
SLS6835I VTV VTV103 excluded – referenced within 60 days
SLS6835I VTV VTV104 excluded – referenced within 60 days
SLS6833I VTV VTV105 deleted from MVC M00007
SLS6833I  VTV VTV106 deleted from MVC M00157
SLS6834I VTV VTV107 deleted

SLS6833I VTV VTV108 deleted from MVC M00072
SLS6833I VTV VTV110 deleted from MVC M00757
SLS6833I VTV VTV111 deleted from MVC M00767
SLS6833I VTV VTV112 deleted from MVC M01057
SLS6834I VTV VTV113 deleted


SUMMARY:
10 scratch VTVs deleted
1 VTV EXCLUDED - NOT SCRATCH VOLUME
0 VTVS EXCLUDED - NOT INITIALIZED

3 SCRATCH VTVS EXCLUDED - REFERENCED WITHIN 60 DAYS

 6 scratch VTV delete candidates bypassed after MAXVTV limit 10 reached
```

FIGURE 1-23 DELETSCR Report

DELETSCR Report Contents

A DELETSCR report shows:

- A line for each deleted VTV.
- If DETail is specified, a line for each VTV excluded (did not fall within the NOTREF period).
- A summary showing:
  - Total VTVs deleted.
  - Total VTVs excluded - not scratch.
  - Total VTVs excluded - not initialized.
  - Total VTVs excluded - not resident or migrated.
  - Total VTVs excluded - referenced within the NOTREF period.
  - Total VTVs excluded - MAXVTV limit has been reached.

  Total VTVs excluded - not in the specified Management Class.

DISPLAY

Display displays the status of the following:

- VTSSs.
- VTDs.
- RTD usage or the status of active or queued processes that use an RTD. Use Display to determine the ID of a process you want to cancel with the CANCEL.
- Storage Classes
- Scratch subpools.
- MVC pools.
- Specific VTVs and MVCs.
- CONFIG parameter settings.
- Migrations.
- Tasks.
- Locks.
- Cluster links.
- Clusters.
- VTV replications.
- Usage information about a VTCS command or help information about an HSC message (including but not limited to the messages listed in the “HSC Messages for VTCS Events” section of *VTCS Messages*).

Syntax

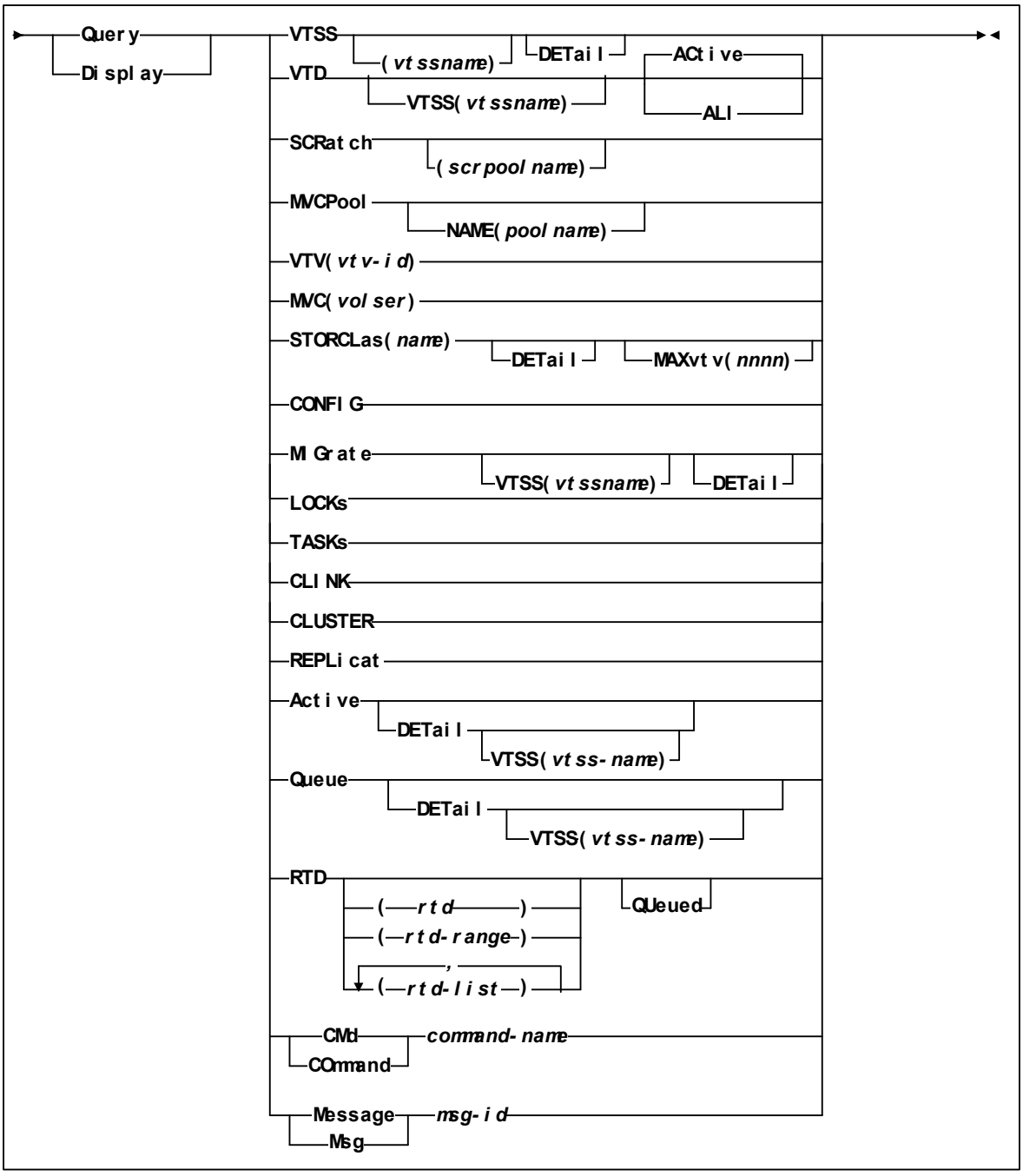


FIGURE 1-24 Query Display syntax

Parameters

VTSS

display VTSS information.

vtssname

the VTSS name. If no *vtssname* is specified, all VTSSs are displayed.

DETail

display detailed host status.

VTD

display VTD information. Lists and ranges of RTDs are limited to 64 items for VSM2s and VSM3s and 256 items for VSM4s.

VTSS

display status for the VTDs connected to the specified VTSS.

vtss-name

the VTSS name.

ACtive | ALl

display status for VTDs that have VTVs mounted (ACtive) or all VTDs (ALl).

SCRatch

display scratch subpool information.

poolname

the name of an MVC Pool that you defined on the MVCPool control statement; for more information, see "MVCPOOL Control Statement" on page 249. Alternatively, you can specify ALL to display information for all Named MVC pools (including the default pool DEFAULTPOOL).

MVCPool

display MVC pool information.

NAME

display information for the specified Named MVC Pool.

poolname

the name of an MVC Pool that you defined on the MVCPool control statement; for more information, see "MVCPOOL Control Statement" on page 249. Alternatively, you can specify ALL to display information for all Named MVC pools (including the default pool DEFAULTPOOL).

VTV

display information about a specific VTV.

vtv-id

the ID of the VTV.

MVC

display information about a specific MVC.

volser

the volser of the MVC.

STORCLas

display information about a Storage Class.

stor-clas-name

the Storage Class name, which can be any Storage Class, not just those defined via the MGMTDEF command. Thus, *stor-clas-name* can take the value of a *vtssname* or !ERROR, in order to allow details of migrations to these Storage Classes to be displayed.

DETail

Optional parameter. If specified, the output will list VTVs currently queued for automatic migration or immediate migration with this Storage Class.

MAXvtv(*nnnn*)

Optional parameter. If specified, indicates the maximum number (0-9999) of VTVs to be listed in a single automatic migration or immediate migration list (for a VTSS). MAXvtv(*nnnn*) implies DETail. If not specified, a default value of 100 is used.

Caution – Use caution when specifying MAXvtv(*nnnn)*. High values can cause temporary system degradation due to the number of WTO (write to operator) messages issued.

CONFIG

display CONFIG parameter settings.

MIGrate

display migration status.

DETail

display migration status by Storage Class.

VTSS

the VTSS whose migration status you want to display.

vtssname

the VTSS name.

TASKs

display task status.

LOCKs

display lock status.

CLINK

display Cluster link status.

CLUSTER

display Cluster status.

REPlicat

display VTV replication status.

Active

display active processes.

DETail

display detailed status.

VTSS

display processes for the specified VTSS.

vtss-name

the VTSS name.

Queue

display queued processes.

DETail

display detailed status.

VTSS

display processes for the specified VTSS.

vtss-name

the VTSS name.

RTD

display usage information for the specified RTDs. Lists and ranges of RTDs are limited to 8 items for VSM2s and VSM3s and 16 items for VSM4s.

rtd–id, rtd-range, or *rtd-list*

the unit addresses of one or more RTDs.

the command name.

QUeued

display information about requests queued for the RTD.

CMd or Command

display syntax and use information for a VTCS command.

cmd-name

the command name.

Msg or Message

display detailed HSC message information.

msg-id

the four–digit numerical portion of the message identifier. Leading zeros are not required.

Interfaces

SWSADMIN utility and VT command **except** for the following, which are only valid as commands:

- VT Display CMD or VT Query CMD
- VT Display MSG or VT Query MSG

JCL Requirements

The following are the required and optional statements for the DISPLAY command JCL:

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the utility report.

SLSIN

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

Output

Display VTSS Output

FIGURE 1-25 shows an example of Display VTSS output.

VTSSNAME	CAPACITY(GB)	DBU	HI AMT	LOW AMT	VTV COUNT	MX MT	MN MT	DEF ACS	AUTOMIG	STATE
HBVTSS16	56.209	9	35	30	204	6	1	--		ONLINE-P
HBVTSS17	56.209	7	35	30	218	4	3	02		ONLINE-P
HBVTSS18	N/A	N/A	35	30	N/A	3	1	01		OFFLINE
HBVTSS19	93.184	5	35	30	110	3	1	01		ONLINE

FIGURE 1-25 Example output from Display VTSS

VTSSNAME

the name of the VTSS.

CPCTY(GB)

the total physical capacity in gigabytes of the specified VTSS.

DBU

the percentage of disk buffer used of the total buffer capacity.

HI AMT

the high AMT.

LOW AMT

the low AMT.

VTV COUNT

the number of VTVs resident on the VTSS.

MX MT

the current MAXMIG value.

MN MT

the current MINMIG value.

DEF ACS

the default ACS.

AUTOMIG

indicates which host is performing the auto migration and the threshold to which the VTSS is migrating.

STATE

one of the following global VTSS states for all hosts:

QUIESCING

Quiescing state.

QUIESCED

Quiesced state.

OFFLINE

Offline state.

OFFLINE-P

Offline pending state.

ONLINE

Online state.

ONLINE-P

Online pending state.

STARTED

The VTSS is initialized and in process of going to the requested state (online, offline, or quiesced).

Display VTSS DEtail Output

FIGURE 1-26 shows an example of the additional fields for Display VTSS DEtail output.

VTSSNAME	HOST	NOMIGRAT?	NORECLAM?	STATE
HBVTSS16	EC104	Y	Y	ONLINE
HBVTSS16	EC21	N	Y	ONLINE
HBVTSS17	EC21	N	Y	QUIESCED
HBVTSS17	EC10	Y	Y	OFFLINE
VTSS	TYPE	SUPPORTED FEATURES		
HBVTSS16	VSM5	2/4GB VTVS		
		LARGE PAGE VTVS		
		NOWAIT ON RTD MOUNT		
		BUDDY QUEUING		
		SYNC REPLICATE		
		LAST USE HINT		
		MGMT CLASS ON MVC		
		CONNECT TO 32 RTDS		
		STACKED MIGRATES		

FIGURE 1-26 Example additional output from Display VTSS Detail

VTSSNAME

the VTSSs that the hosts in the **HOST** field can access.

HOST

the hosts that have access to the VTSSs in the **VTSSNAME** field.

NOMIGRAT

whether NOMIGRAT is set on for this host.

NORECLAM

whether NORECLAM is set on for this host.

STATE

one of the following VTSS states for this host:

QUIESCING

Quiescing state.

QUIESCED

Quiesced state.

OFFLINE

Offline state.

OFFLINE-P

Offline pending state.

ONLINE

Online state.

ONLINE-P

Online pending state.

STARTED

The VTSS is initialized and in process of going to the requested state (online, offline, or quiesced).

TYPE

VTSS model (VSM2, VSM3, VSM4, VSM5, or unknown).

SUPPORTED FEATURES

One or more of the following:

400MB | 800MB | 2/4GB VTVS

Default VTV size.

LARGE PAGE | STANDARD PAGE VTVS

Default VTV page size.

NOWAIT ON RTD MOUNT

VTSS supports polling for RTD mount completion. It does not lock the Nearlink interface while waiting for the mount.

BUDDY QUEUING

VTSS supports queueing of requests to more than one RTD on a Nearlink interface.

REPLICATION

Asynchronous replication enabled.

SYNC REPLICATE

Synchronous replication enabled.

LAST USE HINT

VTSS supports cache management hints indicating when a VTV will be accessed in the near future.

MGMT CLASS ON MVC

Audit MVC is able to return management class for VTVs.

CONNECT TO 16 | 32 RTDS

Maximum number of RTDs per VTSS enabled.

STACKED MIGRATES

Stacked migrates enabled.

UNKNOWN VTSS FEATURE

VTSS feature found not supported by VTCS software level.

NONE

No VTSS features defined.

Display VTD Output

FIGURE 1-27 shows an example of Display VTD output.

DRIVE	LOCATION	VTV	STATUS
A800	HBVTSS16	X00778	ECAM only
A801	HBVTSS16		ECAM only
A802	HBVTSS16		ECAM only
A803	HBVTSS16		ECAM only

FIGURE 1-27 Example output from Display VTD

DRIVE

the MVS device address of the VTD. If the device has not been defined to this host in the CONFIG, then this will contain physical address within the VTSS prefixed by '##'.

LOCATION

the VTSS that contains the VTD.

VTV

the VTV volser if applicable.

STATUS

one of the following VTD statuses:

Available

the VTD is available for work.

Mounted

the VTV volser shown in the VTV column is mounted on the VTD.

Mounting

the VTV volser shown in the VTV column is in the process of being mounted on the VTD. Typically, this indicates that an auto recall is in progress.

Dismounting

the VTV volser shown in the VTV column is was mounted on the VTD and the VTD has been unloaded. VTCS either has not received the dismount request or is currently in the progress of synchronizing the VTV and CDS information.

ECAM only

ECAM communication only (for replication).

Undefined

the VTD has been defined in the VTCS CONFIG but does not physically exist within the VTSS. This is typically only seen on VSM2s when defined with more than 64 VTD addresses

Display RTD Output

FIGURE 1-28 shows an example of Display RTD output.

```
RTD Mount   Alloc  Host     VTSS     Status       Top ID Top host
2A00 -       -      -       DVTSS16  Online/free   -       -
2A01 -       -      -       DVTSS16  Online/free   -       -
2A02 -       -      -       DVTSS16  Online/free   -       -
2A03 -       -      -       DVTSS16  Online/free   -       -
2A04 -       -      -       -        Online/free   -       -
2A05 -       -      -       -        Online/free   -       -
2A06 -       -      -       DVTSS18  Fail/offline  -       -
2A07 -       -      -       DVTSS18  Path offline  -       -
2A08 -       -      -       DVTSS19  Online/free   -       -
2A09 -       -      -       DVTSS19  Online/free   -       -
2A0A -       -      -       -        Online/free   -       -
                            DVTSS18  Path offline
                            DVTSS19  Online/free
                            DVTSS17  Online/free
2A0B -       -      -       -        Online/free   -       -
                            DVTSS18  Path offline
                            DVTSS19  Online/free
                            DVTSS17  Online/free
```

FIGURE 1-28 Example output from a VT Display RTD command

RTD

the unit address of the RTD.

MOUNT

the volser of the MVC currently mounted (an * indicates an in-process mount.

ALLOC

the volser of the MVC allocated for mounting on the RTD.

HOST

the host that currently owns the RTD.

VTSS

the VTSS that is currently connected to the RTD.

Note – For RTDs shared between VTSSs:

- A single line is displayed if the global status of the shared RTD applies to all of the connected VTSSs.
- If the global status of the shared RTD does not apply to all of the connected VTSSs, an individual line is displayed showing the status of each VTSS.

Status

One of the following RTD statuses:

RECOVER RTD

The RTD is being reset after a problem, a vary, or an initialization.

MIGRATE VTV

The RTD is migrating a VTV.

RECALL VTV

The RTD is recalling a VTV.

UNLOAD MVC

A forced unload of the RTD is occurring.

VTV TRANSFER

The RTD is migrating a VTV before recalling it on another VTSS.

AUDIT MVC

An MVC is being audited.

BUSY

The RTD is busy (non-specific task).

IDLE

An MVC is allocated to the RTD but the MVC is not being used.

ONLINE/FREE

The RTD is online and available.

MAINTENANCE

The RTD is in maintenance mode.

OFFLINE

The RTD is offline and unavailable to all hosts and VTSSs

RECOVERY

The RTD is being reset following an error or a vary online mode.

INITIALIZE

The host is verifying RTD status and availability.

SUSPEND

The RTD operations are suspended. This occurs under the following conditions:

- When two RTDs are connected to two separate ports on the same VSM4 ICE3 card CIP or VSM5 ESCON CIP.
- When one or more RTDs and a CLINK are configured on the same port. The

RTDs remain in SUSPEND mode while the CLINK is online.

PATH OFFLINE

The RTD status is unknown because the VTSS cannot contact the RTD or if the paths were not correctly configured.

PATH SUSPEND

The RTD is globally online but the path from the VTSS is suspended due to the RTD being paired with a Clink.

FAIL/OFFLINE

The RTD was placed offline due to a failure.

TOP ID

the process Id of the request that is top of the queue for next using this RTD from this host. The TOP ID column only applies for requests from the host upon which the command has been executed. This host may not have the top claim upon the RTD.

TOP HOST

the host which has the request that is top of the queue for next using this RTD. The TOP HOST indicates which host has the top claim upon the RTD.

Display RTD QUeued Output

FIGURE 1-29 shows an example of Display RTD QUeued output.

ID	Function	Weight	VTSS	MVC list / Storage Class
01360	<X00029>	432*	HBVTSS18	021748 022524
01425	Recall	2	!ALLVTSS	021754
01423	Recall	2	!ALLVTSS	021675
01368	Recall	2	!ALLVTSS	022551
01381	Migrate	2	HBVTSS19 -	*HBVTSS19

FIGURE 1-29 Example output from a VT Display RTD QUeued command

ID

the process ID, which is a unique number in the range 0 - 65536. When the process ID reaches 65536 it wraps back to zero.

FUNCTION

the type of request that is queuing for an RTD. If it is a VTV volser inside angle brackets (<>), then this is an auto recall request for a virtual mount.

WEIGHT

the weighting factor that VTCS has currently assigned to the request. The requests will be considered and queued (reported) according to this factor. An asterisks (*) next to the value indicates that the request has been waiting sufficient long to warrant the stealing of another MVC's allocation.

VTSS

the VTSS or the VTSS list name to be used for selecting RTDs. The special value !ALLVTSS indicates that any VTSS with access to a suitable drive is eligible.

MVC LIST

the list of MVC for which we are attempting to select a RTD. For an auto recall request, this list may run to four MVCs. Conversely, for a migration request that is yet to select a MVC, this will be empty.

STORAGE CLASS

the storage class for which the migration is destined. An asterisks (*) next to the value indicates that is storage class is in an error state.

Display Active and Display Queue Output

FIGURE 1-30 shows an example of Display Active output.

MIGRATES=3	RECALLS=2	RECLAIMS=0

FIGURE 1-30 Example output from the VT Display command (no detail)

FIGURE 1-31 shows example of Display Active DETail output.

Function	ID	VTV	MVC	RTD	VTSS	Task
Reclaim@	01509	-	-	-	-	RCM
.ReclmMVC	01516	*ABORT 021688	-	-	RCM	Active
.ReclmMVC	01517	-	021591	-	-	RCM
. VtvMover	01532	-	021591	-	-	RCM
.	Recall	01533	Y00388 021591	-	!ALLVTSS	Drv
.ReclmMVC	01518	-	021752	-	-	RCM
. VtvMover	01582	-	021752	-	-	RCM
.	Recall	01583	-	021752	-	!ALLVTSS
.ReclmMVC	01519	-	022527	-	-	RCM
.ReclmMVC	01520	-	021650	-	-	RCM
.ReclmMVC	01521	-	022535	-	-	RCM
.ReclmMVC	01522	-	021620	-	-	RCM
.ReclmMVC	01523	-	021545	-	-	RCM
.ReclmMVC	01524	-	021668	-	-	RCM
Migrate@	01563	Demand <=10	-	-	Cmd	Child Finish
.Migrate	01564	-	-	-	HBVTSS19	Cmd
Migrate	01568	X00560 021763	2A0A HBVTSS18	RTD	VTV lock	
Migrate	01569	X00560 021766	2A0B HBVTSS18	RTD	VTV lock	
Migrate	01570	X00560 022534	2A01 HBVTSS18	RTD	VTV lock	
Migrate	01571	X00560 022525	2A06 HBVTSS18	RTD	MVC mount	
Reclaim@	01580	-	-	-	-	RCM
Display@	01590	-	-	-	-	DSP
*SLS5013I Command completed (0)						

FIGURE 1-31 Example output from a VT Display Active DETail command

Note – *ABORT appears in the display of reclaim requests if the request has cancelled or abended.

FIGURE 1-32 shows an example Display Queue DETail output.

FUNCTION	ID	VTV	MVC	RTD	VTSS	Task	REASON
Reclaim@	00003	-	-	-	-	RCM	Child finish
.ReclmMVC	00008	-	DMV053	-	-	RCM	Child finish
. VtvMover	00010	-	DMV053	-	-	RCM	Child finish
. Migrate	00020	-	DMV068	-	DHSS16	DRV	RTD allocation
.ReclmMVC	00009	-	DMV055	-	-	RCM	Child finish
. VtvMover	00012	-	DMV055	-	-	RCM	Child finish
. Recall	00013	DY069	DMV055	2A00	DHSS16	RTD	MVC mount
Reclaim@	01509	-	-	-	-	RCM	Child Finish
.ReclmMVC	01516	*ABORT 021688	-	-	RCM	Active	
.ReclmMVC	01517	-	021591	-	-	RCM	Child Finish
. VtvMover	01532	-	021591	-	-	RCM	Child Finish
.	Recall	01533	Y00388 021591	-	!ALLVTSS	Drv	RTD allocation
.ReclmMVC	01518	-	021752	-	-	RCM	Child Finish
. VtvMover	01582	-	021752	-	-	RCM	Child Finish
.	Recall	01583	-	021752	-	!ALLVTSS	DrvRTD allocation
.ReclmMVC	01519	-	022527	-	-	RCM	Queued
.ReclmMVC	01520	-	021650	-	-	RCM	Queued
.ReclmMVC	01521	-	022535	-	-	RCM	Queued
.ReclmMVC	01522	-	021620	-	-	RCM	Queued
.ReclmMVC	01523	-	021545	-	-	RCM	Queued
.ReclmMVC	01524	-	021668	-	-	RCM	Queued
Migrate@	01563	Demand <=10	-	-	Cmd	Child Finish	
.Migrate	01564	-	-	-	HBVTSS19	Cmd	Active
Migrate	01568	X00560 021763	2A0A HBVTSS18	RTD	VTV lock		
Migrate	01569	X00560 021766	2A0B HBVTSS18	RTD	VTV lock		
Migrate	01570	X00560 022534	2A01 HBVTSS18	RTD	VTV lock		
Migrate	01571	X00560 022525	2A06 HBVTSS18	RTD	MVC mount		
Reclaim@	01580	-	-	-	-	RCM	Queued
Display@	01590	-	-	-	-	DSP	Active
*SLS5013I Command completed (0)							

FIGURE 1-32 Example output from a VT Display Queue DETail command

The Function column shows the hierarchy of requests by indenting. The VTSS column shows either a VTSS name or a VTSS list. The Task column shows which task is currently servicing the request (same information as reported in Display Tasks).

Depending upon the parameters supplied, the output may contain only requests for a specific VTSS, be just the requests in a queued state, be just requests in an active state or be all requests. As can be seen above, the VTV and MVC columns are also used to expose additional information about the nature and status of the request.

Function

The type of request. The hierarchy of the requests and their relationship is implied by the indentation of the values in this column. It can take on one of the following values:

AllocSCR

Job allocation request for a scratch VTV.

AllocVTV

Job allocation request for a specific VTV.

Audit#

Audit utility request.

Cancel@

Cancel command.

Consold#

Consolidate or export utility task.

Consolid

Recall VTVs for remigration to a consolidation MVC. This will appear as a child request to an Int_cons or Consold# request.

Dismount

Dismount a VTV from a VTD.

Display@

Display or query command.

Drain

Recall VTVs from MVC for remigration during drain or reclaim processing. This is a child of a VtvMover request.

Drain@

Drain command or utility.

DrainMVC

There is one DrainMVC request per MVC being drained. DrainMVC, which is a child request of a Drain@ request, is responsible for managing the entire drain process for a single MVC. The VTV column is used to indicate the status of the processing against the MVC.

DELETSCR

Delete scratch utility.

GetmgPol

Obtain current management and storage class definitions.

GetConfg

Get configuration information.

HSCChnge

Notification of parameter files being changed.

Import#

Importing of VTV or MVC by a utility.

Int_cons

PGMI initiated consolidate request

Migrate

General request to perform the migrations of VTVs to a MVC. This may appear as a child to a number of other request types.

Migrate@

Migrate command or utility. This includes migrates to threshold and auto migrates. The latter two are signified further details in the VTV and MVC columns as to the source of the command and the target threshold.

Mount

Mount a VTV upon a VTD. Depending upon circumstances, this may be subsequently seen as a VTV transfer or recall request.

Move MVC

There is one Move MVC request per MVC being processed by reconcile or archive. Move MVC, which is a child request of a MoveVTV# request, is responsible for managing the entire VTV movement process for a single MVC. The VTV column is used to indicate the status of the processing against the MVC.

MoveVTV#

This is a request from the ARCHIVE or RECONCILE utility commands to move copies of VTVs between MVCs. The value -TIME- in the VTV column indicates that the ELAPSED parameter was specified.

MvcMaint

MVCMAINT utility request.

MVC_chek

Check status of MVC.

MVC_eot

Reset the end of tape position of a MVC after completing a drain or reclaim. This is a child of either a DrainMVC, ReclmMVC or Move MVC request.

MVC_upd

Reset or update MVC status.

MVC_inv

Audit of an MVC. This will appear as a child request to an Audit# request.

MVCpool#

Obtain details and status of MVC pools for a utility.

PGMI_req

A request received through the PGMI interface that has yet to be decoded.

Query@l

Query or Display command.

Recall

General request to perform the recall of VTVs from a MVC. This may appear as a child to a number of other request types.

Recall@

Recall command or utility.

Reclaim@

Auto reclaim request or a Reclaim command or utility. The value -TIME- in the VTV column indicates that the ELAPSED parameter was specified.

ReclmMVC

There is one ReclmMVC request per MVC being reclaimed. ReclmMVC, which is a child request of a Reclaim@ request, is responsible for managing the entire reclaim process for a single MVC. The VTV column is used to indicate the status of the processing against the MVC.

Reconcil

Perform a crosscheck between the contents of the two VTSSs in a cluster.

Replicat

Perform the replication of VTVs between VTSSs in a cluster.

Scratch

Scratch a VTV request from HSC.

Sel_scr

PGMI select scratch.

Set@

Set command.

Transfer

Mount a VTV upon a VTD by transferring the VTV between two VTSSs.

Unload

Unload MVC from RTD.

Uscratch

Unscratch a VTV request from HSC.

Vary_dev

Perform vary processing against an individual RTD or CLINK. This will appear as a child request to an VARY@ request.

Vary@

Vary command.

VtvMaint

VTVMAINT utility request.

VTVMover

There is one VTVMover request per MVC being drained or reclaimed. This is a child of either a DrainMVC, ReclmMVC or Move MVC request. This request is responsible for the movement of VTVs from one MVC to another.

VTSS_Inv

Audit of a VTSS. This will appear as a child request to an Audit# request.

VTV_list

Obtain a list of VTV resident within a VTSS. This will appear as a child request to a Reconcil or auto migration request.

VTV_upd

Resynchronize VTV status in the VTSS with the CDS.

ID

The process ID, which is a unique number in the range 0 - 65536. When the process ID reaches 65536 it wraps back to zero.

VTV

the volser of the VTV currently being used in the process. For some types of request, this will contain additional information as to the nature and status of the request.

MVC

the volser of the MVC currently used in the process. For some types of request, this will contain additional information as to the nature and status of the request.

VTSS

the VTSS or the VTSS list name associated with the request. The special value !ALLVTSS indicates that any VTSS with suitable requirements will be used.

RTD

the unit address of the RTD currently being used in the process.

TASK

the task that is processing the queue or the task to which the requests is queued (same information as reported in Display Tasks).

REASON

why the request is queued (queued processes only):

ACTIVE

The request is currently being processed.

CANCELLED

The request is terminating after being cancelled

CHILD FINISH

The request has child requests and is waiting for them to finish.

DEVICE LOCK

The RTD or CLINK device that the request requires is currently locked. This generally indicates contention with another host.

TASK LOCK

The request is waiting for a general task lock to free. This generally indicates contention with another host.

MVC LOCK

The request is waiting for a lock on an MVC to free.

VTV LOCK

The request is waiting for a lock on a VTV to free.

VTD LOCK

The request is waiting for a lock on a VTD to free.

RTD LOCK

The request is waiting for a lock on an RTD to free.

QUEUED

The request is sitting in the input queue of the task and is waiting for another request to complete or be rescheduled.

WAIT RESOURCE

The request is held awaiting a (non-specific) resource becoming available.

RTD ONLINE

The request requires an RTD to be brought online to continue.

RTD ALLOCATION

The request is queued awaiting a RTD to become idle or free.

MVC SELECTION

The request is queued awaiting a MVC or migration slot becoming available.

STEAL A RTD

The request is waiting to steal an RTD allocation from another request.

DBU DROP

The request is currently held because the DBU is high.

MVC MOUNT

The request is waiting for a MVC to be mounted.

Display SCRatch Output

FIGURE 1-33 shows an example of Display SCRatch output.

SUBPOOL-NAME	SCRATCH-COUNT	LABEL TYPE	<----------SIZES AVAILABLE---------->				TOT RNG	TOT BLKS
			0.4GB	0.8GB	2GB	4GB		
M0	2	SL	1	1	1	1		
M1	3	SL	1	1	1	1		

FIGURE 1-33 Example output from Display SCRatch

SUBPOOL-NAME

the name of the scratch subpool. Note that VTCS does not display scratch counts for non-subpool VTVs. If there are no VTV subpools defined, VTCS defines the Virtual Tapes pool for all VTVs.

SCRATCH-COUNT

the number of CDS blocks containing scratches for valid VTV sizes (.4GB,.8GB, 2GB, 4GB), plus the number of CDS ranges and the total number of CDS blocks for this scratch pool.

Display MVCPool Output

FIGURE 1-34 shows an example of Display MVCPool NAME (POOL1) output.

	MVCPOOL (POOL1) INFORMATION						
INITMVC	MVCFREE	THRESHOLD	MAXMVC	START			
NO	2	30	4	10			
ACS	MEDIA	FREE-MVCS		RECLAIM-MVCS		USED-MVCS	
		VOLS	GB	VOLS	GB	VOLS	GB
00	ECART	120	96	2	0.5	90	45
00	STK1R	30	600	1	3.5	25	350
00	TOTAL	150	696	3	4.0	115	395

FIGURE 1-34 Example output from Display MVCPool NAME(POOL1)

FIGURE 1-35 shows an example of Display MVCPool output (no pool name specified).

	MVCPOOL INFORMATION						
ACS	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

FIGURE 1-35 Example output from Display MVCPool (no pool name specified)

In FIGURE 1-35, **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 ELSs, 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 ELS attached).

INITMVC

specifies whether un-initialized MVCs are to be initialized when they are first mounted.

MVCFREE

the minimum number of free MVCs.

THRESHOLD

the fragmented space threshold (as a percentage) that determines when an MVC is eligible for demand or automatic reclamation.

MAXMVC

MVC limit for a single reclaim.

START

Specifies a percentage value that represents the ratio of reclaim candidates to total MVCs, which triggers automatic space reclamation.

ACS

the ACS containing the MVC pool. **NONLIB** counts are for initialized MVCs that are now outside the library.

MEDIA

the MVC media type.

FREE-MVCS

MVCs that have 100% usable space and do not contain any migrated VTVs. The storage shown is the total free space based on media type capacity.

RECLAIM-MVCS

MVCs eligible for space reclamation by this host. The storage shown is the total wasted space including those MVCs not yet eligible for space reclaim.

USED-MVCS

Initialized MVCs that are partially or completely full.

Display VTV Output

FIGURE 1-36 shows an example of Display VTV output.

Volser:	**X00000**	
VTSS:	HBVTSS19	
Mounted:	A800	
Uncompressed Size(Mb):	99.01	
Compressed Size(Mb):	97.88	
Maximum VTV size(Mb):	800	
Page Size:	Standard	
Creation Date:	2006Feb21 09:32:22	
Last Mounted:	2006Feb21 09:32:23	
Last Recalled:	2006Apr05 08:15:47	
First Migrated:	2006Apr25 08:58:02	
Last Used Date:	2006Apr25 08:58:28	
MVC(s):	021681	021645
Block-id(s):	1A400D24	25401185
Management Class:	M13	
Migrate Pending:	SA	SB
Status:	Initialized	
	Resident	
	Migrated	

FIGURE 1-36 Example output from Display VTV

VOLSER

the volser of the VTV specified in the query.

VTSS

the VTSS where the VTV last or currently resides.

MOUNTED

if the VTV is mounted on a VTD, the VTD unit address is displayed.

UNCOMPRESSED SIZE(MB)

the uncompressed size of the VTV (MB). This is the size of the VTV as perceived by the application programs.

COMPRESSED SIZE (MB)

the compressed size of the VTV (MB). This is the raw space that will be occupied upon the MVCs or within the VTSSs.

MAXIMUM VTV SIZE (MB)

the maximum (compressed) size of the VTV (400, 800, 2000 or 4000).

CREATION DATE

the date and time when the VTV contents was last changed by an application.

LAST MOUNTED

the date and time when the VTV contents was last mount for access by an application.

LAST RECALLED

the date and time when the VTV was last recalled back from a MVC into a VTSS.

FIRST MIGRATED

the date and time when the first MVC copy of this version of VTV was created.

LAST USED DATE

the date and time when the VTV was last touched by VTCS. This includes most functions that update the status of the VTV in some way.

MVC(S)

the MVC(s) where the VTV resides. This entry only appears when the VTV is migrated.

BLOCK-ID

the logical block ID of the beginning of the VTV upon corresponding MVC. This entry only appears when the VTV is migrated.

MANAGEMENT CLASS

the VTV's Management Class.

MIGRATE PENDING

the Storage classes to which migrations are outstanding.

ARCHIVE PENDING

the Storage classes to which migrations are outstanding in order to satisfy archiving requirements.

STATUS

one or more of the following statuses:

CONSOLIDATED

VSM has consolidated the VTV.

DUPLEXED

The DUPLEX attribute has been assigned to this VTV. When VSM migrates the VTV, a copy will be written to two MVCs.

INITIALIZED

VTCS has used the VTV at least once.

MIGRATED

VSM has migrated the VTV.

RESIDENT

The VTV is resident on the VTSS.

SCRATCH

The VTV is in scratch status.

UNINITIALIZED

The VTV has been defined via the CONFIG utility, but has not ever been used.

REPLICATION REQUIRED

This VTV should be replicated and is currently waiting for replication.

REPLICATION STARTED

Replication is active for this VTV but not yet complete.

REPLICATION COMPLETE

A fully replicated copy of this VTV is now resident in the Secondary VTSS.

Display MVC Output

FIGURE 1-37 shows an example of Display MVC output.

Volser:	022551
Media:	STK1R
ACSid:	01
Size(Mb):	20000
Migrated Count:	32000
Vtv Count:	5
%Used:	4.69
%Fragmented:	0.00
%Available:	95.31
%Usable:	95.31
Times Mounted:	174
Last Mounted:	2006Jun20 04:42:58
Last Migration:	2006Jun19 03:51:07
Last Drain/Reclaim:	2006Jun19 03:18:03
Owner:	SB
VTSS:	HBVTSS19
MVCPOOL:	DEFAULTPOOL
Security access:	No profile
Status:	Initialized

FIGURE 1-37 Example output from Display MVC

VOLSER

the volser of the MVC.

MEDIA

the volume media type or recording technique. Should the value ‘Need PTF’ appear, then this host lacks support for this media type. Another host does have support for this media type.

ACSID

the ACS that contains the MVC. This will not appear if the MVC has been ejected from the library.

SIZE(MB)

the size of the MVC in megabytes.

MIGRATED COUNT

the number of VTV copies that have been migrated onto the MVC.

VTV COUNT

the number of active VTVs on the MVC.

%USED

the percentage of the MVC used by valid VTVs.

%FRAGMENTED

the percentage of the MVC that has invalid VTV space that is not available for use until it is reclaimed or the MVC is drained.

%AVAILABLE

the percentage of the MVC that is physically available for use.

%USABLE

the percentage of space on the MVC that can be used by VTCS. This may be zero even if there is still space physically available. For instance, if the VTV per MVC limit is reached then the %Usable will be reported as 0%. Similarly, if an error has been reported against an MVC then VTCS will not use this MVC for output and the %Usable will be reported as %0.

TIMES MOUNTED

the number of times the MVC has been mounted for writing or reading since it was added to the MVC inventory.

LAST MOUNTED

the date and time at which the MVC was mounted or attempted to be mounted upon a RTD.

LAST MIGRATED

the date and time at which the last VTV migration was performed to the MVC.

LAST DRAIN/RECLAIM

the date and time at which the MVC was last processed by Drain or reclaim processing and had it's end-of-tape pointer reset.

OWNER

the Storage Class that owns the MVC. A MVC only becomes a member of a storage class when it contains migrated VTVs.

VTSS

the last VTSS that performed a migration to the MVC. CONSOLIDATE appears in this field for consolidated VTVs.

MVCPOOL

either an MVC Pool Name (including DEFAULTPOOL) or NO if the MVC is not defined on an MVCPool statement.

SECURITY ACCESS

VTCS permissions for the MVCs defined in an MVCPOOL statement (UPDATE, NO UPDATE, or NO PROFILE).

STATUS

one of the following statuses:

INITIALIZED

the MVC has been initialized.

BEING AUDITED

The MVC is either currently being audited or has been the subject of a failed audit. While in this state the MVC will not be used for migration and can be used for recalls. Due to the inherent state, recalls may fail because the CDS not yet up-to-date with the MVC contents. To clear this condition, rerun the AUDIT utility against this MVC.

LOGICALLY EJECTED

The MVC has either been the subject of an MVCDRain Eject or the MVC was rejected for update by a RACROUTE call. The MVC will not be used again for migration. To clear this condition, use MVCDRain against the MVC without the Eject option.

NOT-INITIALIZED

The MVC has been defined via the CONFIG utility, but has not ever been the subject of a successful VTV migration.

MOUNTED

The MVC is mounted on an RTD.

IN ERROR

This is a generic error that indicates the MVC, drive, or combination of the two has specifically reported a problem whilst accessing the MVC. VTCS attempts to de-preference the usage of MVCs with this state. VTCS considers this to be the most serious affliction on a MVC when considering copies to be used for recalls.

In general, to clear this state:

- If the MVC caused the problem, use a DRAIN(EJECT) command to remove the MVC from service.
- If the RTD caused the problem, use the MVCMAINT utility to reset the MVC state.

Note also that one or more of the following messages is issued for IN ERROR status: SLS6686, SLS6687, SLS6688, SLS6690. For detailed recovery procedures for these messages, see *VTCS Messages and Codes*.

MARKED FULL

The MVC is considered as being full and is not a candidate for further migrations. Due to the general behavior of tape media this can occur before the MVC has reached its nominal capacity.

DRAINING

The MVC is either currently the subject of drain or reclaim processing. Should the processing fail, the MVC maybe left in this state as a safeguard. To clear this condition, perform a MVCDRain against the MVC.

LOST - FAILED TO MOUNT

VTCS attempted to mount an MVC and the mount did not complete within a 15-minute time-out period. VTCS has had no specific error report although there could be combination of hardware problems, HSC problems, or by the MVC being removed from the ACS. VTCS attempts to de-preference the usage of MVCs with this state.

Determine the cause of the error and fix it. You can also use the VTCS MVCMAINT utility to set LOST(OFF) for the following events:

- LOST(ON) was set due to LSM failures or drive errors that have been resolved
- LOST(ON) was set because the MVC was outside the ACS and has been reentered.

This condition is automatically cleared by VTCS if it subsequently requests a mount of the MVC and this is successful.

DATA CHECK

A data check condition has been reported against this MVC. VTCS attempts to de-preference the usage of MVCs with this state. To get into this state, a data transfer must have failed upon two different RTDs.

To clear this state:

- If all VTVs on the MVC are duplexed, use MVCDRain on the MVC without the Eject option. This recovers all VTVs and removes the MVC from service.
- f all VTVs on the MVC are not duplexed, VTCS AUDIT the MVC. The audit will probably fail. After the audit, do an MVCDRAIN (no eject). This recalls the VTVs before the data-check area in ascending block-id order and the VTVs after the data-check area in a descending block-id order. Processing the VTVs in this sequence ensures that VTCS recovers as many VTVs as possible from the media. You then need to recreate the data for any VTVs still on the MVC.

Note that although this indicates that a specific failure has occurred when performing data transfers, this may not be a fault in the media. It could be that a RTD is writing data to the media out of specification. Patterns of failures are therefore important. As an example, lots of DATA CHECK conditions suddenly occurring lots of drives and volumes.

READ ONLY

The MVC has been marked read-only because of one of the following conditions:

- The MVC being the target of an export or consolidation process. The read-only state protects the MVC from further updates.
- The MVC media is set to file protect. Correct the error and use the MVCMAINT utility to set READONLY(OFF).
- The MVC does not having the appropriate SAF rules set to enable VTCS to update the MVC. Correct the error (for more information, see "Defining a Security System User ID for HSC, SMC, and VTCS" in Chapter 2 of *VTCS Installation and Configuration Guide*) and use the MVCMAINT utility to set READONLY(OFF).

RETIRED

The MVC is retired and is considered by VTCS as having reached the end of its useful life. VTCS will recall from, but not migrate to, the MVC. Replace the MVC as soon as possible. Once this has been done, use the MVCMAINT utility to set RETIRED(OFF).

WARRANTY HAS EXPIRED

The MVC's warranty has expired. VTCS continues to use the MVC. You should start making plans to replace the MVC when it reaches Retired state.

INVALID MIR

VTCS has received status from an RTD to indicate the MIR (media information record) for a 9x40 media is invalid. An invalid MIR does not prevent access to data but may cause significant performance problems while accessing records on the tape. The MVC is not capable of high-speed searches on areas of the tape that do not have a valid MIR entry.

VTCS attempts to de-preference MVCs with this condition. For recalls, if the VTV resides on multiple MVCs, VTCS selects MVCs with valid MIRs ahead of MVCs with invalid MIRs. VTCS avoids using MVCs with invalid MIRs for migration, unless the migration is at the beginning of the tape. Migrating from the beginning of tape will correct the MIR. VTCS detects the invalid MIR condition at either mount time or dismount time. If detected at mount time and the operation can be completed with another MVC, VTCS dismounts the first MVC and selects the alternate MVC.

Note that VTCS has only a limited ability to switch to an alternate MVC. That is, it is mainly used for migrate and virtual mount. For MVCs with invalid MIRs, determine the cause of the error, which may be caused by media or drive problems, and fix the error. To recover an MVC with an invalid MIR, you simply need to read the MVC to the end of the tape, which can be done via a VTCS audit. If the media is the problem, run an MVCDRAIN EJECT to recall the VTVs and cause the MVC to be removed from the MVC pool.

MIGRATES NOT SUPPORTED

This host lacks support for performing migrations to this MVC. This is set by another host that does support migration.

RECALLS NOT SUPPORTED

This host lacks support for performing recalls from this MVC. This is set by another host that does support recalls.

RECLAIM NOT SUPPORTED

This host lacks support for considering this MVC for reclaim processing. This is set by another host that does support reclaim processing. This does not inhibit the MVC being processed through the MVCDRain command.

Display STORCLas Output

FIGURE 1-38 shows an example of Display STORCLas output.

```
Summary of storage class XCA8:

MVCs must be in ACS 00 and MVCPool DEFAULTPOOL

RTDs can be any device type
MVCPool DEFAULTPOOL contains no free MVCs

There is 1 VTV awaiting auto-migration from VTSS CLIVSS16
V00002 with MGMTCLAS XCA8

VTSS CLIVSS16 has no suitable RTDs online in ACS 00
VTSS CLIVSS17 has no suitable RTDs online in ACS 00
```

FIGURE 1-38 Example output from Display STORCLas

As shown in FIGURE 1-38, the output from Display STORCLas shows:

- The characteristics of the Storage Class (ACS, MVC Pool, and Media).
- VTVs waiting migration to the Storage Class from any VTSS.
- Requirements of the MVCs to be used for migration.
- The device type(s) of the RTDs needed to write to the migration MVCs.
- Any errors with regard to satisfying the migration requirements.

Display CONFIG Output

FIGURE 1-39 shows an example of Display CONFIG output.

```
MAXVTV          MVCFREE     VTVATTR     RECALWER    SYNCHREP    REPlicat    VTVPAGE     MAXRTDS
4000            10          SCRATCH     YES         YES         ALWAYS      LARGE       32
FASTMIGR        INITMVC
YES             YES
LOCK DATA IS HELD IN COUPLING FACILITY STRUCTURE STK_VTCS_LOCKS


CDS LEVEL       V5/5.1      V6          V6.1        V6.2
SUPPORT:
                                        *           *


RECLAIM:        THRESHOLD   MAXMVC      START       CONMVC
                30          10          10          2


VTSSNAME        AUTO MIGR THR           MIGR TASKS  DEFAULT     VSM         2 GB/4 GB   PAGE
                                                                                        SIZE
                LOW         HIGH        MIN   MAX   ACS         MODEL
HBVTSS16        60          80          1     1     FF          4           Y           LARGE
HBVTSS17        60          80          1     4     02          4           Y           LARGE
HBVTSS18        60          80          4     4     01          4           Y           LARGE
HBVTSS19        60          80          1     1     01          4           Y           LARGE


DEVNO           RTD TYPE    ACS         RETAIN      VTSSNAME    RTD NAME    CHANIF
2A00            STK1RC      00          10          HBVTSS16    SS162A00    0A:0
2A01            STK1RC      00          10          HBVTSS16    SS162A01    0E:0
2A02            STK1RC      00          10          HBVTSS16    SS162A02    0A:0
2A0C            STK1RC      02          10          HBVTSS16    SS162A0C    0M:0
V010            VL          LEPRIM      10          DVTSS18     SS18PRIM    01 1A:0
V011            VL          LEPRIM      10          DVTSS18     SS18PRIM    01 1A:0
```

FIGURE 1-39 Example output from Display CONFIG

In FIGURE 1-39, **note that**:

- In the **Devno** column, ELS device numbers start with a “V” (in this case, devices V000 and V010).
- The **RTD Type** column now contains the same value as reported by HSC in the 'QUERY DRIVES' command, including the **ELS** device type.
- For local tape drives, the 'ACS' column displays the ACS and LSM to which the drive is attached. For remote RTDs or ELS devices, the ACS column displays the name of the ELS appliance.

MAXVTV

the GLOBAL MAXVTV setting.

MVCFREE

the GLOBAL MVCFREE setting.

VTVATTR

the GLOBAL VTVattr setting (**SCRATCH** or **ALLmount**).

RECALWER

the GLOBAL RECALWER setting (**YES** or **NO**).

SYNCHREP

the GLOBAL SYNCHREP setting (**YES** or **NO**).

REPLicat

the GLOBAL REPLicat setting (**ALWAYS** or **CHANGED**).

VTVPAGE

The VTV page size (STANDARD or LARGE).

MAXRTDS

the GLOBAL MAXRTDs setting (16 or 32).

FASTMIGR

the GLOBAL FASTMIGR setting (**YES** or **NO**).

INITMVC

the GLOBAL INITMVC setting (**YES** or **NO**).

LOCK DATA IS HELD IN...

Name of VTCS Lock Structure, if it exists.

CDSLEVEL SUPPORT

the VTCS level(s) that can access the active CDS. For more information, see *Installing ELS*, "Reconfiguring NCS," "Converting the Formatted CDS to VSM Extended Format."

THRESHOLD

the RECLAIM THRESHLD setting.

MAX MVC

the RECLAIM MAXMVC setting.

START

the RECLAIM START setting.

CONMVC

the RECLAIM CONMVC setting.

VTSSNAME

the VTSS names (VTSS NAME settings).

AUTO MIGR THR, LOW

The low automatic migration threshold setting (LAMT) for the VTSS.

AUTO MIGR THR, HIGH

The high automatic migration threshold setting (HAMT) for the VTSS.

MIGR TASKS, MIN

The minimum number of concurrent automatic migration tasks setting (MINMIG) for the VTSS.

MIGR TASKS, MAX

The maximum number of concurrent automatic migration tasks setting (MAXMIG) for the VTSS.

DEFAULT ACS

The default ACS setting (DEFLTACS) for the VTSS.

VSM MODEL

2, 3, or 4.

2GB / 4GB

VTSS configured for 2 GB / 4GB VTV sizes (Y or N).

PAGE SIZE

VTV page size (STANDARD or LARGE).

DEVNO

the RTD MVS device numbers for the VTSS (RTD DEVNO settings).

RTD TYPE

the RTD type.

ACS

the ACS that contains the RTD.

RETAIN

the VTSS RETAIN setting.

VTSSNAME

the VTSS names (VTSS NAME settings) of the VTSSs connected to the RTD.

RTD NAME

the RTD names for the VTSS (RTD NAME settings).

CHANIF

the RTD channel interface (RTD CHANIF settings).

Display MIGrate Output

FIGURE 1-40 shows an example of Display MIGrate output.

```
VTSSNAME: HBVTSS16

ACTIVE MIGRATION TASKS: 4

IMMEDIATE MIGRATE:                    MAX WAIT: 5 MINUTES

IMMEDIATE DELAY QUEUE: ACTIVE

AUTO MIGRATE:                         HOST: EC20 MIGRATION TARGET: 70%
```

FIGURE 1-40 Example output from Display MIGrate

ACTIVE MIGRATION TASKS

the total number of migration tasks (automatic, immediate, and migrate-to-threshold).

IMMEDIATE MIGRATE

either **Not active** if there are no current or pending immediate migrations or the maximum time that any VTV has been waiting for immediate migration.

Note – This field only shows status for the LPAR on which the query was issued.

AUTO MIGRATE

either **Not active** or the name of the host and migration target (LAMT or specified threshold for a migration-to-threshold) if auto migration is active on any host.

Display MIGrate DEtail Output

FIGURE 1-41 shows an example of the additional 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 Class	ACS/ LOCATION	MAX/ RTDs	ONL	REQ	AUTO ACT	IMMED GB	WAIT	GB	WEIGHT/SKIP	
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	

FIGURE 1-41 Example additional output from Display MIGrate DEtail

In FIGURE 1-41, **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.

STORAGE CLASS

the Storage Class associated with the migration.

Note – If you do not explicitly assign a Storage Class, an MVC's default Storage Class is the name of the last VTSS that wrote to the MVC for reclamation or migration and this class has the VTCS default media selections. To change these defaults, create a Storage Class with the VTSS name and specify the desired media selection order.

ACS

the ACS defined for the Storage Class.

MAX/ONL RTDs

- the maximum number of RTD tasks based on the Storage Class and RTD configuration definitions.
- the maximum number of tasks for those RTDs that are actually online (**MAX TASKS** minus the number of offline RTDS).

SCH

SCHLIMIT and SCHPREF values from MIGRSEL statements that apply to the active migration for the storage class. **Note that** in 6.2 the SCHPREF is not a parameter that can be changed. It is shown here since the display is standardized from 6.2 onwards. In 6.2 this parameter is always displayed as zero.

Req Act

the number of active migration tasks.

ACTIVE TASKS

the number of migration tasks currently active for the Storage Class.

AUTO

indicates whether the Storage Class contains automatic migration VTVs.

IMMED

indicates whether the Storage Class contains immediate migration VTVs.

WEIGHT (percent)

the priority of the Storage Class compared to other Storage Classes for the VTSS. Storage Classes with higher priorities are assigned a greater proportion of migration tasks.

Display TASKs Output

FIGURE 1-42 shows an example of Display TASKs output

TASK NBR	TASK TYPE	SLOT ID 003	VTSS	RTD	CURRENT PROCESS	LOCKS HELD	WAITQ COUNT	PENDQ COUNT
00O	DSP	019			518	T		
001	SS	034	HBVTSS16	SS16B200		T		
002	RTD	035	HBVTSS16	SS16B201		R		
003	RTD	036	HBVTSS16	SS160B79		R		
004	RTD	044	HBVTSS16	SS160B7A		R	1	
005	RTD	045	HBVTSS16	SS160B7C		R		

FIGURE 1-42 Example output from Display TASKs

TASK NBR

the task number for each task on the current host.

SLOT ID

the LOCK identifier within the lock buffer.

TASK TYPE

the task type:

INV Inventory Manager
CMD Command Task
DSP Dispatcher Task
SS VTSS Task
RTD RTD Task
DRV RTD Scheduler
SCR Scratch Manager
RCM Reclaim Manager
MSC Migration Scheduler
CSH Clink Scheduler
CLK CLINK Task
UNK Unknown

VTSS

the VTSS name.

RTD

the RTD name for RTD tasks.

CURRENT PROCESS

the current process ID.

LOCKS HELD

type of lock held:

T

Task Lock.

M

MVC Lock.

V

VTV Lock.

D

VTD Lock.

R

RTD Lock.

WAITQ COUNT

the count of requests waiting for locks.

PENDQ COUNT

the count of pending requests.

Display LOCKs Output

FIGURE 1-43 shows an example of Display LOCKs output.

SLOT	OWNING	TASK	TASK	VTD	MVC	VTV	WAITING	WAITING
ID	HOST	NBR	TYPE				HOST	TASK
002	EC21	006	RTD		EVS101		EC10	007
003	EC20	010	RTD		EVS145	X15328		
004	EC20	010		A91E		X153234		
CF LOCK		CF LOCK				OWNING		
NUMBER		TYPE				HOST		
3		HOST TO HOST (TO ECCL)				EC21		

FIGURE 1-43 Example output from Display LOCKs

SLOT ID

Slot ID of the lock within the lock buffer.

OWNING HOST

the host that owns the lock.

TASK NBR

the task number associated with the lock.

TASK TYPE

the task type.

VTD

the associated VTD address on the issuing host.

MVC

the locked MVC.

VTV

the locked VTV.

WAITING HOST

the host waiting for the lock or **ALL** if multiple hosts are waiting.

WAITING TASK

the task waiting for the lock or **ALL** multiple tasks are waiting.

CF LOCK NUMBER

the Coupling Facility lock number.

CF LOCK TYPE

one of the following VTCS Coupling Facility lock types:

Host Footprint

used to serialize access to the host footprint list.

Host to Host

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

Lock data

used to serialize access to the VTCS lock data.

Formatting

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

System

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

OWNING HOST

the host that owns the lock.

Display CLINK Output

FIGURE 1-44 shows an example of Display CLINK output.

VTSS	CLINK	STATUS	USAGE	HOST
HBVTSS19	0C:0	ON-SYNC	REPLICATING	EC21
	0C:1	ON-SYNC	FREE	
	0E:0	ON-SYNC	FREE	
	0E:1	ON-SYNC	FREE	

FIGURE 1-44 Example output from Display CLINK

VTSS

the Primary or Sending VTSS name.

CLINK

the link ID.

STATUS

one of the following link statuses:

Maint

The link has failed or it has been varied into maintenance mode.

Offline

The link is offline and unavailable to all hosts and VTSSs.

On-Sync

Available for synchronous replication.

On-Async

Available for asynchronous replication.

P_offlne

The link is pending offline

P_online

The link is pending online

Recovery

The link is being reset following an error or a vary online operation.

Unusable

Not available for replication due to hardware errors or assigned-elsewhere conditions.

USAGE

one of the following link usages:

Assigned

Link is assigned to the host in the HOST field but is not currently replicating. This usage occurs when VTCS is starting or terminating link use or is attempting error recovery on the link after a replication failure.

Free

Link is idle (not doing replications).

Replicating

Link is actively doing replications.

HOST

the host that the link is assigned to.

Display CLUSTER Output

FIGURE 1-45 shows an example of Display CLUSTER output.

NAME	VTSS	STATE	DIRECTION	VTSS	STATE	MODE
CLUSTER1	HBVTSS16	ONLINE	------>	HBVTSS17	ONLINE	Sync-replicate
CLUSTER2	HBVTSS18	ONLINE	<------	HBVTSS19	QUIESCED	DEGRADED
CLUSTER3	HBVTSS20	ONLINE	<------	HBVTSS21	ONLINE	Async-replicate
CLUSTER4	HBVTSS22	QUIESCED	<----->	HBVTSS23	ONLINE	DEGRADED

FIGURE 1-45 Example output from Display CLUSTER

NAME

the Cluster name.

VTSS

one of the VTSS in a Cluster.

DIRECTION

One of the following:

----->

OR

<-----

Indicates the direction of VTV replication in uni-directional Cluster. VTVs can only be replicated from the Sending to the Receiving VTSS.

<---->

Indicates that the VTSSs are configured as a Bi-Directional (Peer-to-Peer) Cluster. VTVs can be replicated from either VTSS to the other.

STATE

one of the following VTSS states:

QUIESCING

Quiescing state.

QUIESCED

Quiesced state.

OFFLINE

Offline state.

OFFLINE-P

Offline pending state.

ONLINE

Online state.

ONLINE-P

Online pending state.

STARTED

The VTSS is initialized and in process of going to the requested state (online, offline, or quiesced).

MODE

one of the following Cluster operating modes:

Async-replicate

Both VTSSs in the Cluster are online to VTCS. Production workload can go to either VTSS, but in the case of a Uni-Directional (Primary/Secondary) Cluster, VTVs can only be replicated from the Sending VTSS. Synchronous replication is **not** enabled across the cluster.

Sync-replicate

Both VTSSs in the Cluster are online to VTCS. Production workload can go to either VTSS, but in the case of a Uni-Directional (Primary/Secondary) Cluster, VTVs can only be replicated from the Sending VTSS. Synchronous replication **is** enabled across the cluster.

DEGRADED

One of the two VTSSs in a Bi-Directional (Peer-to-Peer) Cluster is either offline or quiesced. Production workload can go the remaining online VTSS. VTVs requiring replication, however, are allocated to the remaining VTSS only if no other Full-Function clusters are available and suitable. In this case, replicate VTVs are migrated immediately with keep and queued for replication when the other VTSS comes online.

When the other VTSS comes online, VTCS reconciles the contents of both VTSSs.

DEGRADED SECONDARY

The Primary is online to VTCS and the Secondary is either offline or quiesced. Workload can run on the Primary. VTVs requiring replication, however, are allocated to the Primary only if no other Full Function Clusters are available. In this case, Replicate VTVs are migrated immediately with keep and are queued for replication, which occurs when the Secondary comes online.

DEGRADED PRIMARY

The Secondary is online to VTCS and the Primary is either offline or quiesced. Workload can run on the Secondary. VTVs requiring replication, however, are allocated to the Secondary only if no other Full Function Clusters are available. When the Primary comes back ONline, VTCS reconciles the contents of the Primary and Secondary.

NON-OPERATIONAL

No workload is possible on this Cluster.

Display REPLicat Output

FIGURE 1-46 shows an example of Display REPLicat output.

VTSS	HOST	QDEPTH
HBVTSS19	EC10	0
	EC20	0
	EC21	1
	ECCL	0
	ECCY	1
	EC31	0

FIGURE 1-46 Example output from Display REPLicat

VTSS

the Primary VTSS name.

HOST

the hosts attached to the Primary VTSS.

QDEPTH

the number of VTVs waiting to be replicated.

EXPORT

EXPORT consolidates VTVs (if required) and creates a manifest file that lists the VTVs and MVCs available for export from a VSM system.

Note – EXPORT is valid only if FEATures VSM(ADVMGMT) is specified; for more information, see "FEATURES Control Statement" on page 229.

Syntax

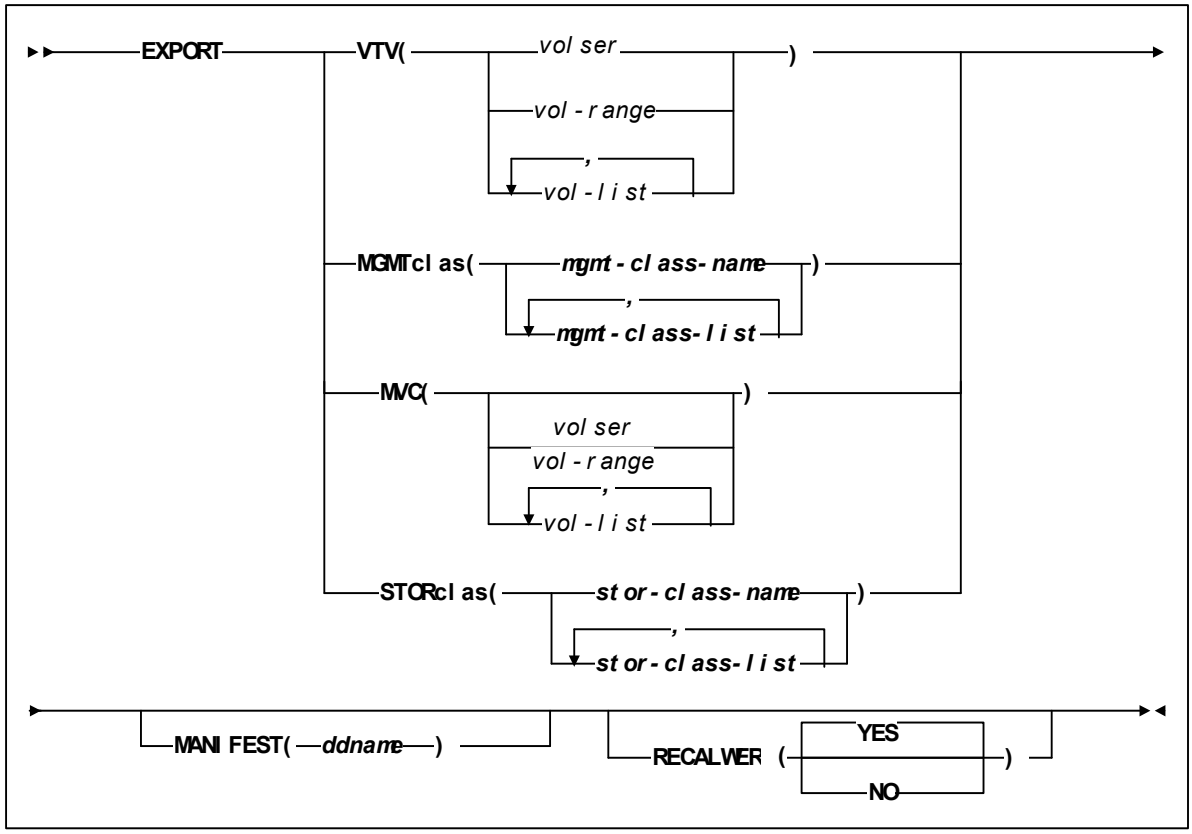


FIGURE 1-47 EXPORT utility syntax

Parameters

VTV

specifies one or more VTVs to consolidate for export.

volser, vol-range or *vol-list*

the volsers of one or more VTVs. You can specify an unlimited number of VTVs.

MGMTclas

specifies one or more Management Classes that determine one or more VTVs to consolidate for export.

mgmt-class-name | *mgmt-class-list*

the names of one or more Management Classes that you defined on the MGMTclas control statement; for more information, see "MGMTCLAS Control Statement" on page 234.

MVC

specifies one or more MVCs for export.

volser, vol-range or *vol-list*

the volsers of one or more MVCs.

STORclas

specifies one or more Storage Classes that determine one or more MVCs for export.

stor-clas-name | *stor-clas-list*

the names of one or more Storage Classes that you defined on the STORclas control statement; for more information, see "STORCLAS Control Statement" on page 252.

MANIFEST

specifies the output destination ddname of the manifest file.

ddname

ddname of the manifest file. The default is MANIFEST.

RECALWER

specifies whether VTCS recalls VTVs with read data checks.

YES

recall VTVs with read data checks (the default).

NO

Do not recall VTVs with read data checks.

Interfaces

SWSADMIN utility only.

Optional and Required JCL

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules (required).

manifest file DD

DD statement for the manifest file (optional).

SLSPRINT

specifies the destination for the report (required).

SLSIN

specifies the input to the SWSADMIN program (utility name and parameters) (required).

IMPORT

IMPORT imports VTVs and MVCs listed on a manifest file into a VSM system.

Note –

- IMPORT is valid only if FEATures VSM(ADVMGMT) is specified; for more information, see “FEATURES Control Statement” on page 229.
- Ensure that the "to" CDS has the same features (enabled by CDS level) as the "from" CDS. For example, if the “from” CDS has Large VTV page sizes enabled and 2/4 Gb VTVs have been created, then the "to CDS" must have the same capabilities, otherwise the import fails.

Syntax

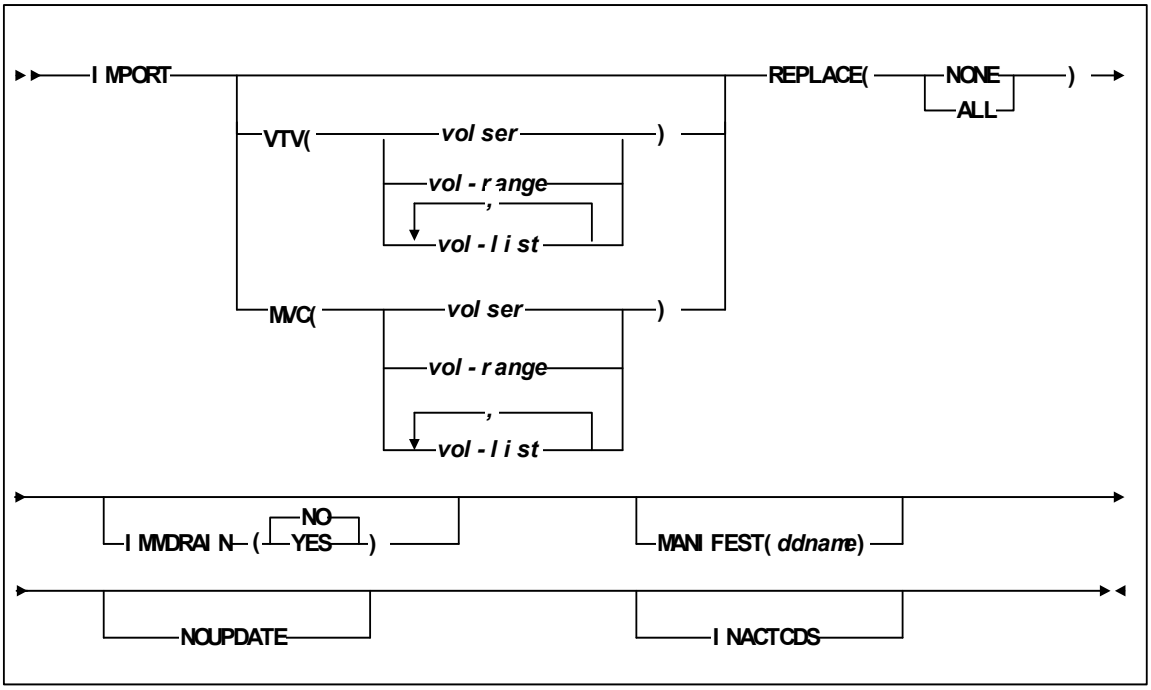


FIGURE 1-48 IMPORT utility syntax

Parameters

VTV

specifies one or more VTVs to import.

volser, vol-range or *vol-list*

the volsers of one or more VTVs.

MVC

specifies one or more MVCs to import.

volser, vol-range or *vol-list*

the volsers of one or more MVCs.

REPLACE

specifies whether VSM replaces the VTV record in the target CDS. There is no default; you must specify a value.

NONE

Do not replace the VTV record. VTCS only creates new records for VTVs that are not duplicates and replaces records for VTVs not initialized in the target CDS.

ALL

Replace any duplicate VTV records in the target CDS.

Caution – Ensure that you actually want to replace duplicate VTV records in the target CDS before you specify the ALL parameter! You may want to do a "validate" run with NOUPDATE to see which VTV records will be replaced.

Also note that if a VTV record is replaced, all existing VTSS and MVC copies of the VTV are invalidated.

Finally, you cannot import an MVC if the target CDS records show that the MVC contains VTVs, even if you specify REPLACE(ALL). In this situation, you must first drain (with eject) the MVC on the target system and eject it from the ACS. You can then import the MVC that you exported from the source system.

IMMDRAIN

specifies whether VSM will immediately drain imported MVCs.

NO

Do not drain MVCs (the default).

YES

Drain MVCs.

MANIFEST

specifies the input ddname of the manifest file.

ddname

ddname of the manifest file. The default is MANIFEST.

NOUPDATE

specifies that VSM does not update the CDS, validates the import operation, and writes information messages to the job log.

INACTCDS

specifies that the import uses a different CDS from the CDS currently active on the HSC system where you are running the import job. You specify the alternate CDS in the SWSADMIN JCL. **Note that** if HSC is down on the system where you are running the import job, the CDS on that system is assumed to be inactive, so you do not need to specify INACTCDS.

Interfaces

SWSADMIN utility only.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

manifest file DD

DD statement for the manifest file.

SLSPRINT

specifies the destination for the utility report.

SLSIN

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

SLSCNTL

Specifies the Primary CDS. **Not required** if importing to the CDS being used by the HSC/VTCS system that is active on the system where the import job runs. Otherwise it is required, to specify the Primary CDS to be updated by import.

If you specify the INACTCDS parameter, this CDS must be **different** from the CDS being used by HSC/VTCS (if active).

SLSCNTL2

Specifies the Secondary CDS, if it exists. **Not required** if importing to the CDS being used by the HSC/VTCS system that is active on the system where the import job runs. Otherwise it is required, to specify the Secondary CDS to be updated by import.

If you specify the INACTCDS parameter, this CDS must be **different** from the CDS being used by HSC/VTCS (if active).

SLSSTBY

Specifies the Standby CDS, if it exists. **Not required** if importing to the CDS being used by the HSC/VTCS system that is active on the system where the import job runs. Otherwise it is required, to specify the Standby CDS to be updated by import.

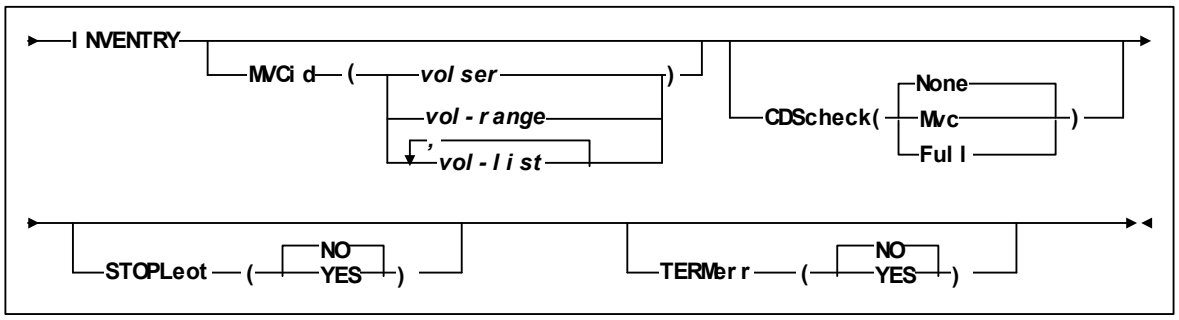
If you specify the INACTCDS parameter, this CDS must be **different** from the CDS being used by HSC/VTCS (if active).

INVENTRY

The INVENTRY utility lists all VTVs on specified MVCs. Additionally, you can use INVENTRY to:

- Request that the contents of the specified MVCs are cross-checked with the information recorded in the CDS.
- Terminate the processing based on the end-of-tape position recorded in the CDS for the MVC, rather than at the MVC's physical end-of-tape.
- Have the processing terminated for an MVC the first time a VTV is discovered that is incompatible with the information in the CDS.

Syntax



Parameters

MVCid

specifies the MVC(s) to be inventoried.

CDScheck

specifies whether the contents of each inventoried MVC is checked against the information recorded in the CDS.

None

Each VTV found on an inventoried MVC is listed via message SLS6933I. The CDS is not checked. None is the default.

Mvc

Each VTV found on an inventoried MVC is listed via message SLS6933I. The details of the VTV are obtained from the CDS. The MVC and CDS copies of the VTV are compared and listed as a comment on SLS6933I. If the CDS shows a resident or MVC copy of the VTV exists that differs from the copy listed in SLS6933I, its details are listed via message SLS6934I.

Full

Each VTV found on an inventoried MVC is listed via message SLS6933I.

The details of the VTV are obtained from the CDS. The MVC and CDS copies of the VTV are compared and listed as a comment on SLS6933I. If the CDS shows a resident or MVC copy of the VTV exists that differs from the copy listed in SLS6933I, its details are listed via message SLS6934I. For each inventoried MVC, the CDS is checked to see which VTVs are shown to have a copy on the MVC. Any VTVs that were not found on the MVC are listed via message SLS6935E.

STOPleot

specifies whether processing is to terminate when the logical end-of-tape position is reached. The logical end-of-tape position is the end-of-tape position recorded in the CDS for this MVC.

NO

All VTVs are to be processed on each inventoried MVC. NO is the default.

YES

VTVs are processed on each inventoried MVC until:

- The MVC's physical end-of-tape is reached, or
- The first VTV is found on the MVC that is beyond the logical end-of-tape position, if this is earlier than the physical end-of-tape position. Invalid with CDScheck(None).

TERMerr

specifies whether the inventory of an MVC is to terminate when it encounters the first inconsistency between the contents of the MVC and the VTVs that the CDS indicates are on the MVC. Such inconsistencies are listed via message SLS6935E, SLS6936E or SLS6938E.

NO

The inventory of an MVC is not to terminate when it encounters the first inconsistency between the contents of the MVC and the VTVs that the CDS indicates are on the MVC. NO is the default.

YES

The inventory of an MVC is to terminate when it encounters the first inconsistency between the contents of the MVC and the VTVs that the CDS indicates are on the MVC. Invalid with CDScheck(None).

MERGMFST

MERGMFST merges multiple manifest files produced by EXPORT into a single file.

Syntax

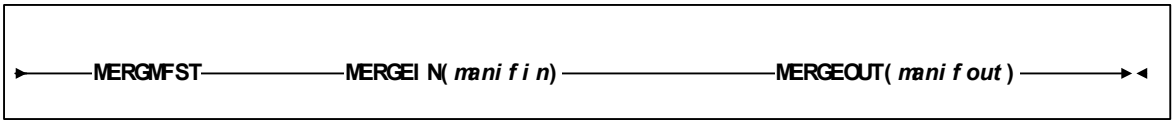


FIGURE 1-49 MERGMFST utility syntax

Parameters

MERGEIN

specifies the DD statement for one or more input manifest files.

manifin

the DD name.

MERGEOUT

specifies the DD statement for the merged manifest file.

manifout

the DD name.

Interfaces

SWSADMIN utility only.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

manifin DD

DD statement for the input manifest file(s).

manifout DD

DD statement for the merged manifest file.

SLSPRINT

specifies the destination for the report.

SLSIN

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

MIGRATE

MIGrate migrates VTVs to MVCs.

Syntax - Format 1

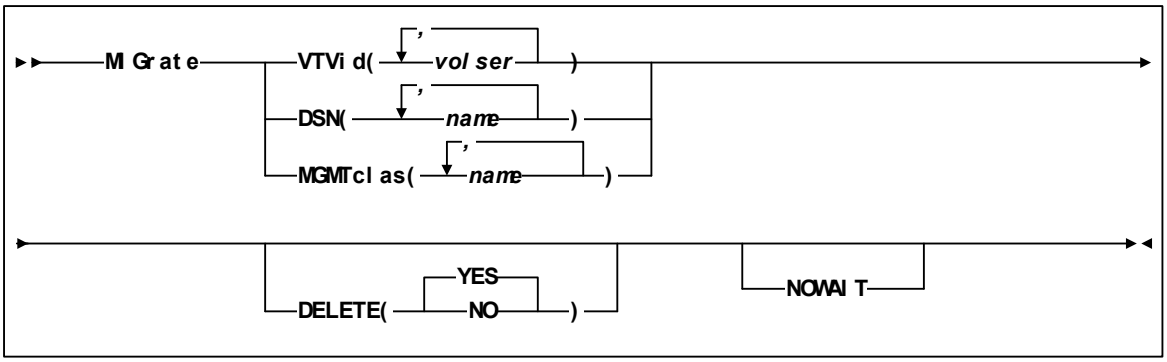


FIGURE 1-50 MIGrate utility syntax - Format 1

Parameters - Format 1

VTVid

specifies the VTVs that VSM migrates.

volser

the volsers of one or more VTVs. You can also specify one or more ranges.

DSN

specifies data sets used to select VTVs to migrate.

name

the data set name. TABLE 1-3 describes the valid wild cards for data set names. You cannot address a member of a GDG using a wildcard.

Note – Wildcard are only supported on MVS systems running DFSMS/MVS 1.4 or greater. At systems below this level the catalog search does not support wildcard.

TABLE 1-3 Valid Wildcards

Wildcard	Stands for...
*	A qualifier or one or more characters within a qualifier. An asterisk can precede or follow a set of characters.
**	zero or more qualifiers. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a period or a blank.
% or ?	Exactly one alphanumeric or national character.
%% or ??	One to eight percent signs or question marks can be specified in each qualifier.

MGMTclas

specifies one or more Management Classes that determine one or more VTVs to migrate.

mgmt-class-name | *mgmt-class-list*

the names of one or more Management Classes that you defined on the MGMTclas control statement; for more information, see "MGMTCLAS Control Statement" on page 234.

Note – The VTVid, DSN, and MGMTclas parameters are mutually exclusive.

DELETE

specifies whether VSM deletes VTVs from the VTSS after migrating the VTVs.

NO

do not delete VTVs from the VTSS after migrating the VTVs.

YES

delete VTVs from the VTSS after migrating the VTVs (the default).

NOWAIT

specifies that the utility does not wait for the operation to complete and returns after the request is submitted.

Syntax - Format 2

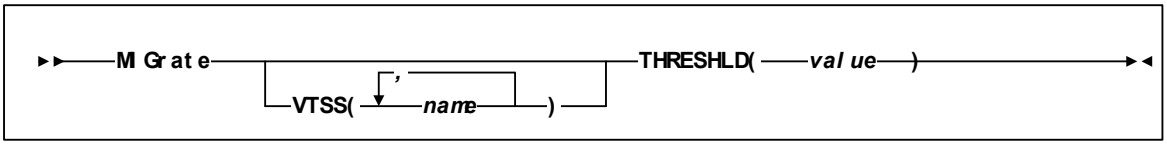


FIGURE 1-51 MIGrate utility syntax - Format 2

Parameters - Format 2

VTSS

specifies one or more VTSSs to migrate to the specified threshold.

name

the names of one or more VTSSs.

THRESHLD

specifies that VTCS runs the VTSS space management/VTV migration cycle until VTSS space reaches the specified threshold. Valid values are 0 to 95%.

value

the threshold to migrate to (percent of VTSS space).

Interfaces

SWSADMIN utility and VT command.

JCL Requirements

The following are the required and optional statements for the MIGrate JCL:

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the report.

SLSIN

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

MVCDEF

The VT MVCDEF command loads the MVCPool statements from a specified definition data set.

Syntax

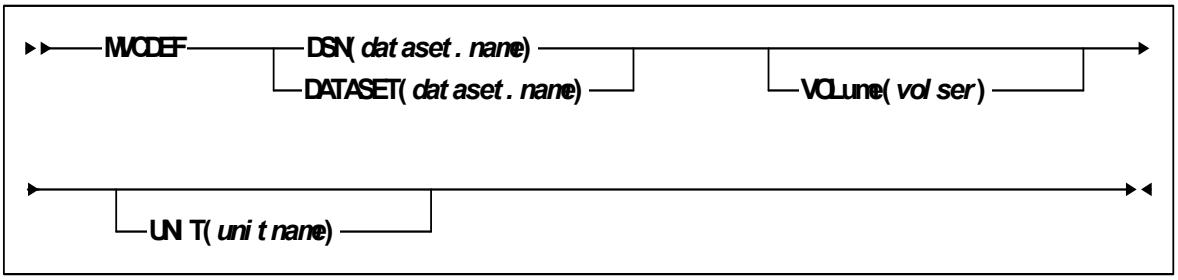


FIGURE 1-52 VT MVCDEF command syntax

Parameters

DSN or DATASET

specifies the definition data set that contains the MVCPool statements to load.

dataset.name

the data set name. If the data set name includes a member name, enclose the data set name in quotes.

VOLume

specifies the DASD volume where the definition data set resides. This parameter is optional, unless the data set is not cataloged, or the data set resides on a volume other than the volume indicated by the catalog.

volser

the DASD volser.

UNIT

specifies the DASD device where the definition data set resides.

unitname

the DASD unit name. If the definition data set is not cataloged and this parameter is omitted, the unit name defaults to SYSALLDA.

Interfaces

VT command only.

MVCDRAIN

MVCDRain recalls all current and scratched VTVs from an MVC and, optionally, “virtually” ejects the MVC (makes it unavailable for VSM use without physically ejecting it from the library). You can use the MVCDRain to override the CONFIG RECLAIM CONMVC setting.

Syntax

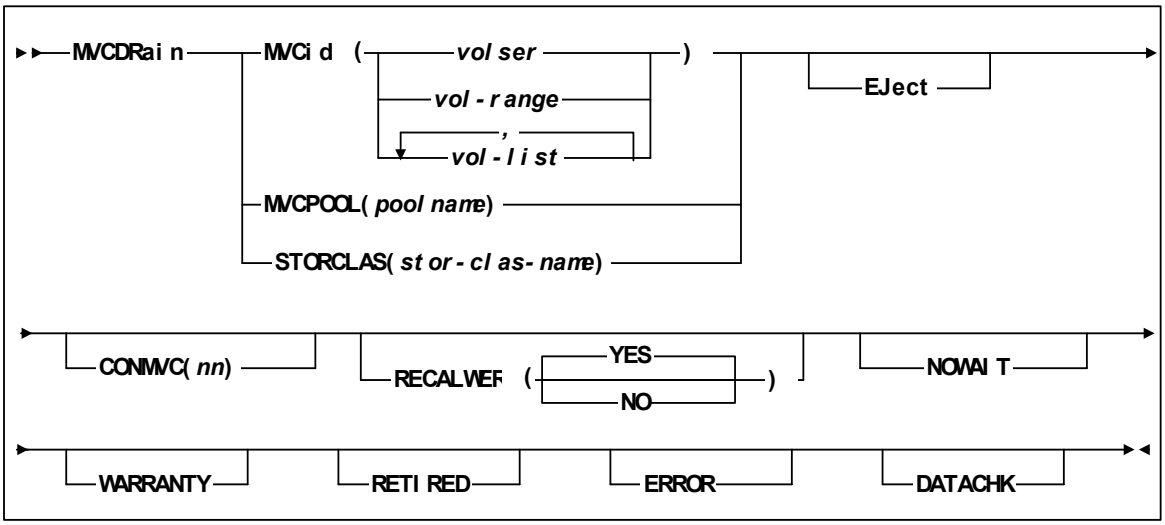


FIGURE 1-53 MVCDRain syntax

Parameters

MVCid

drain one or more MVCs by volser.

volser, vol-range, or *vol-list*

the volsers of one or more MVCs up to a maximum of 50.

MVCPOOL

drain the MVCs in the specified Named MVC Pool.

poolname

the name of an MVC Pool that you defined on the MVCPool control statement; for more information, see “MVCPOOL Control Statement” on page 249.

STORCLAS

drain the MVCs in the specified Storage Class.

stor-class-name

the name of a Storage Class that you defined on the STORclas control statement; for more information, see "STORCLAS Control Statement" on page 252.

EJect

specifies that VTCS "virtually" ejects the MVC (the MVC will not be used for output).

CONMVC(*nn)*

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

Valid values are 1 to 99. If not specified, the default is the CONMVC value specified on the CONFIG RECLAIM statement.

RECALWER

specifies whether VTCS recalls VTVs with read data checks.

YES

recall VTVs with read data checks (the default).

NO

Do not recall VTVs with read data checks.

NOWAIT

specifies that the utility does not wait for the operation to complete and returns after the request is submitted.

WARRANTY

selects MVCs with expired warranties (denoted by a W in the Status T column on an MVC Report).

RETIRED

selects MVCs that are retired (denoted by a T in the Status T column on an MVC Report).

ERROR

selects MVCs that are in error (denoted by a B in the Status B column on an MVC Report).

DATACHK

selects MVCs that have a data check (denoted by a D in the Status D column on an MVC Report).

Interfaces

SWSADMIN utility and VT command.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the report.

SLSIN

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

MVCMAINT

MVCMAINT sets MVC attributes.

Syntax

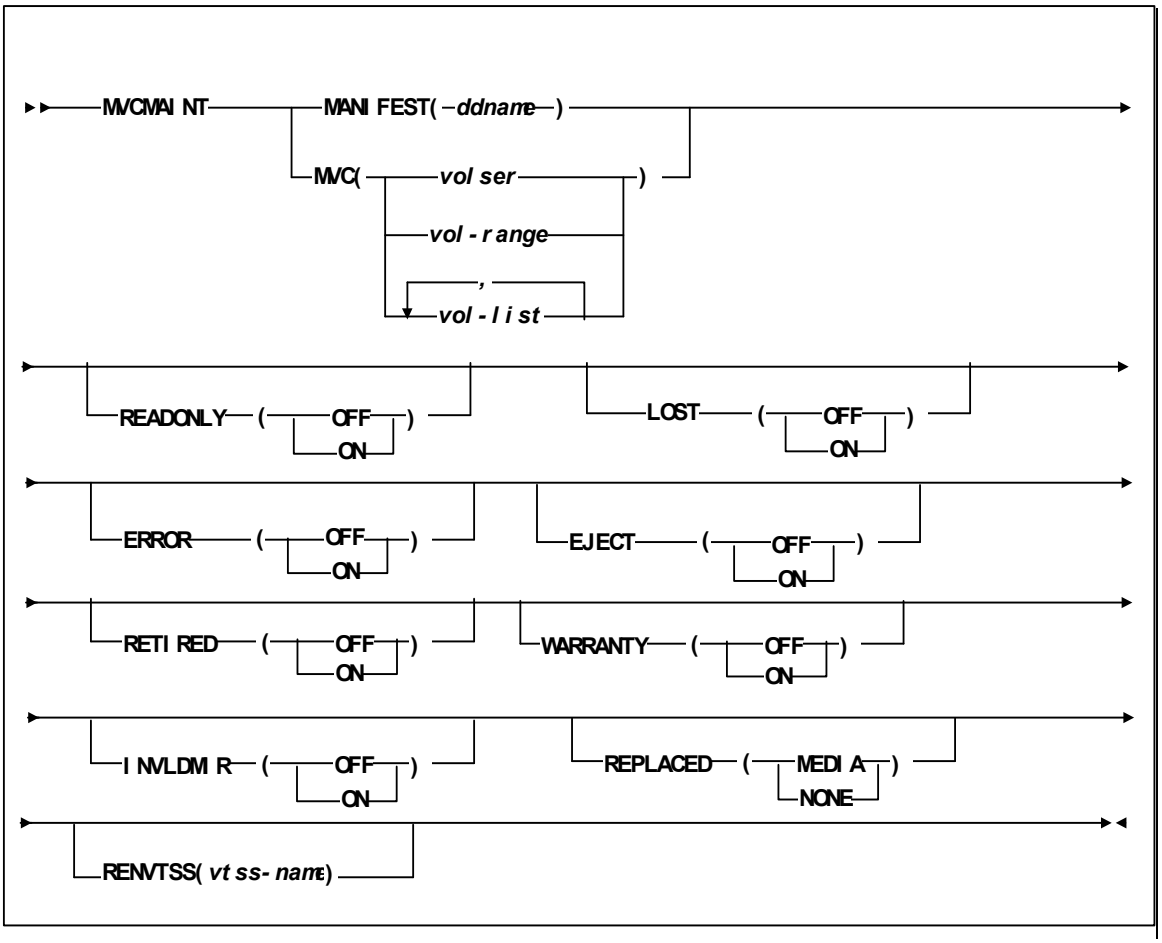


FIGURE 1-54 MVCMAINT syntax

Parameters

MANIFEST

specifies the input ddname of the manifest file. **Note that** you can specify a merged manifest file or multiple manifest files.

ddname

ddname of the manifest file. The default is MANIFEST.

Note – When you specify `MVCMAINT MANIFEST`, VTCS does the following:

- Reads the Manifest file to discover which MVCs are to be updated (all MVCs in the manifest file).
- Updates the MVCs in the CDS with the action specified on the other parameters (`LOST`, `READONLY`, and so forth).

MVC

specifies the MVCs whose read/write attribute is changed.

volser, vol-range or *vol-list*

the volsers of one or more MVCs.

READONLY

sets the read/write status of the MVC. There is no default; you must specify a value.

ON

MVC is readonly.

OFF

MVCs is writable.

LOST

sets the “lost” status of the MVC. There is no default; you must specify a value.

ON

MVC is lost.

OFF

MVC is not lost.

ERROR

sets the error status of the MVC. There is no default; you must specify a value.

ON

MVC is in error.

OFF

MVC is not in error.

EJECT

sets the “logical eject” status of the MVC. There is no default; you must specify a value.

ON

MVC is “logically ejected”.

OFF

MVC is not “logically ejected”.

MVCMAINT will only allow EJECT(ON) to be set for an MVC if ALL of the following are true

- The MVC is not marked Full, and
- The MVC is not Draining, and
- The MVC contains zero VTVs.

RETIRED

sets the “retired” status of the MVC. There is no default; you must specify a value.

ON

MVC is retired.

OFF

MVC is not retired but is still in “expired warranty” state (still selectable for output).

WARRANTY

sets the “expired warranty” status of the MVC. There is no default; you must specify a value.

ON

MVC’s warranty has expired.

OFF

MVC’s warranty has not expired.

Note – RETIRED and WARRANTY are mutually exclusive. That is, you cannot specify RETIRED and WARRANTY on the same MVCMAINT command.

INVLDMIR

sets the invalid MIR status of the MVC. There is no default; you must specify a value.

ON

MIR is invalid.

OFF

MIR is not invalid.

REPLACED

Resets or clears various fields in the MVC record after physically replacing an MVC. There is no default; you must specify a value.

NONE

Makes no updates to the MVC record.

MEDIA

Update the MVC record. The MVC must be empty (contain zero VTVs) and unmounted or this parameter will fail. `REPLACED(MEDIA)` updates the following MVC fields as follows:

- Media type of the MVC is cleared
- Count of times mounted is set to zero
- Warranty Expired Flag is turned off
- Invalid MIR Flag is turned off
- Retired Flag is turned off
- Broke/Error Flag is turned off
- Lost Flag is turned off
- Data Check Flag is turned off
- VTSS Last Written By is cleared
- Date/Time MVC was last written to is cleared
- Storage Class is cleared
- Date/Time of Last Drain/Reclaim is cleared
- Date/Time of Migrate from Empty State is cleared

Note – The REPLACED parameter is mutually exclusive with the LOST, ERROR, EJECT, RETIRED, WARRANTY, and INVLDMIR parameters.

RENVTSS

Sets the name of the owning VTSS to vtss-*name*.

vtss-name

the VTSS name.

Note – To set the VTSS name:

- The host on which the utility is run must have access to the VTSS, and
- The new VTSS name must be the name of an existing VTSS in the VTCS `CONFIG` deck.

Interfaces

SWSADMIN utility only.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

manifest file DD

DD statement for the manifest file.

SLSPRINT

specifies the destination for the report.

SLSIN

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

Note – In addition to the above, the JCL used to execute MVCMAINT:

- **May** specify the CDS to be updated if the CDS is in use by HSC/VTCS on the LPAR where `MVCMAINT` is run. If specified, the CDS datasets specified in `MVCMAINT` JCL must match the CDS datasets used by HSC/VTCS.
- **Must** specify the CDS to be updated if `MVCMAINT` is being executed in batch-only mode, that is, when there are no hosts active (on any LPAR) using the CDS.

If the CDS is specified in MVCMAINT JCL, it must be as follows:

- `SLSCNTL` specifies the primary CDS data set.
- `SLSCNTL2` specifies the secondary CDS data set, if it exists.
- `SLSSTDBY` specifies the standby CDS data set, if it exists.

Prior to this PTF, when the CDS was specified in MVCMAINT JCL, only SLSCNTL had to be specified even if secondary or standby CDS data sets exist. That is not the case when this PTF is applied.

Return Codes

TABLE 1-4 MVCMAINT Return Codes

Return Code	Meaning
0	All requested updates for the MVC completed successfully.
4	One requested update for the MVC failed and at lest one other requested update completed successfully.
8	All requested updates for the MVC failed.

Note – The **final** return code for the MVCMAINT job is the **largest** return code generated by any single MVCs being updated. For example, if 5 MVCs generate a return code 0 and one MVC generates a return code 8, the final return code is 8.

Usage

You can run MVCMAINT as follows:

- With HSC/VTCS active (on the same LPAR), or
- in batch-only mode when there are no Hosts active (on any LPAR) using the CDS that is to be updated. The CDS is specified in the MVCMAINT JCL via //SLSCNTL (and, if appropriate, //SLSCNTL2 and //SLSSTBY).

Note –

- An MVCMAINT request that specifies the RENVTSS parameter cannot be run in batch-only mode, because it requires HSC/VTCS services.
- MVCMAINT MVC(volsers) REPLACED(MEDIA) now clears the date and time the MVC was last mounted.

MVCMAINT Reports

FIGURE 1-55 shows an example of an MVCMAINT report for the following command:

MVCMAINT MVC(022577-022579) READONLY=OFF

SWSADMIN (6.2.0)	STORAGETEK VTCS SYTEM UTILITY	PAGE 0002
TIME 09:26:54	MVC MAINTENANCE	DATE 2006-05-14

MVCMAINT SUMMARY REPORT

MVC	RC
022577	00
022578	08
022578	08

MVCMAINT EXCEPTION REPORT

*SLS6737I MVC 022578 ALREADY HAS READONLY(OFF); REQUEST IGNORED

*SLS6737I MVC 022579 ALREADY HAS READONLY(OFF); REQUEST IGNORED

SLS1315I SWS500.V5.CDS WAS SELECTED AS THE PRIMARY CONTROL DATA SET

SWSADMIN (6.2.0)	STORAGETEK VTCS SYTEM UTILITY	PAGE 0002
TIME 09:26:54	VTCS MVC SUMMARY REPORT	DATE 2006-05-14

MVC	NUMBER	%USED	%AVAIL	%FRAG	MEDIA	TIMES	STATUS	<-----LAST MOUNTED---->			A C S	OWNER/
VOLSER	OF VTVS				SIZE (MB)	MOUNTE D	I B L D R U T M	DATE	TIME	VTSS	ID	CONSOLID ATE TIME
022577	0	0.00	99.96	0.04	400	142	I - - - - C - -	2006MAY14	06:23:23		00	2006MAY14 06:09:23
022578	0	0.00	99.96	0.04	400	197	I - - - - U - -	2006MAY14	06:23:23	VTSS16	00	VTSS16
022579	0	0.00	99.96	0.04	400	142	I - - - - U - -	2006MAY14	16:23:23		00	2006MAY14 16:09:23

3	INITIALIZED MVCS PROCESSED
0	NON-INITIALIZED MVCS PROCESSED
0	NON-LIBRARY MVCS PROCESSED

FIGURE 1-55 Example MVCMAINT report

As shown in FIGURE 1-55, the MVCMAINT report shows:

- Status of MVCs processed - volser and return code (0 - all updates completed, 4 - some updates completed, 8 - no updates completed).
- An exception report of the reason for all uncompleted updates.
- An MVC summary report.

MVCPLRPT

The MVCPLRPT reports the status of a Named MVC Pool or MVC Storage Class.

Syntax

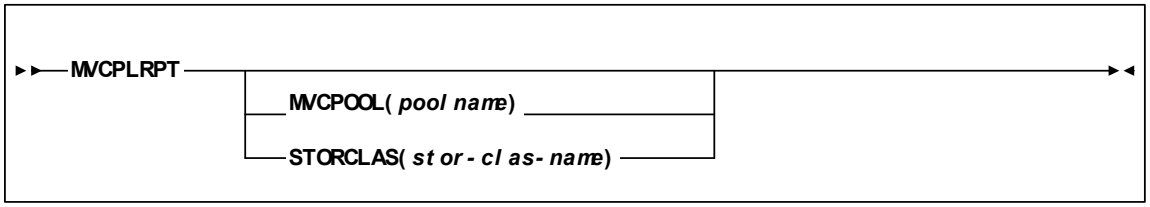


FIGURE 1-56 MVCPLRPT syntax

Parameters

MVCPOOL

report on the MVCs in the specified Named MVC Pool.

poolname

the name of an Named MVC Pool that you defined on the MVCPool control statement; for more information, see "MVCPOOL Control Statement" on page 249.

To report on all Named MVC Pools (including DEFAULTPOOL), specify ALL or omit the MVCPOOL parameter.

STORCLAS

report on the MVCs in the specified Storage Class. The report is produced as if the associated MVC pool name had been specified.

stor-class-name

the name of a Storage Class that you defined on the STORclas control statement; for more information, see "STORCLAS Control Statement" on page 252.

Interfaces

SWSADMIN utility only.

JCL Requirements

The following are the required and optional statements for the MVCPLRPT JCL:

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the report.

SLSIN

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

Named MVC Pool Report

FIGURE 1-57 and Figure 57 on page 102 show an example of a report for Named MVC Pool CUST1POOL.

SWSADMIN (6.2.0) STORAGETEK VTCS SYTEM UTILITY PAGE 0002

TIME 09:26:54 VTCS MVC SUMMARY REPORT - MVCPOOL= CUST1POOL DATE 2006-04-13

MVC VOLSER	NUMBER OF VTVS	%USED	%AVAIL	%FRAG	MEDIA SIZE (MB)	TIMES MOUNTED	STATUS I B L D R U T M	<-----LAST MOUNTED----> DATE	TIME	VTSS	ACS ID	OWNER/ CONSOLIDATE TIME
EVS99	200	10.80	84.57	4.63	2000	310	I - - - - U - M	2006MAR15	03:20:23	VTSS8	00	S1
EVS100	0	0.00	100.00	0.00	UNKNOWN	206	- - L - - U - -	2006MAR10	05:24:04	VTSS8	--	
EVS101	1009	99.00	0.00	1.00	400	306	I - - - - U - -	2006MAR15	03:20:23	VTSS8	00	S1
EVS102	5	8.25	91.75	0.00	400	6	I - - - - U - -	2006MAR15	04:23:04	VTSS8	00	S3
EVS103	EXPVTV	0.12	99.88	0.00	400	194	I - - - - J - -	2006MAR15	03:20:28	VTSS10	00	VTSS10
EVS104	0	0.00	100.00	0.00	400	5	I - - - R C - -	2006MAR18	03:49:14	VTSS8	00	2006APR12 03:49:14
EVS105	200	10.80	84.57	4.63	102040	254	I - - - R U T -	2006MAR18	04110:09	VTSS8	00	
EVS106	0	0.00	100.00	0.00	400	202	I - - - - C - -	2006MAR18	03:49:20	VTSS8	00	
EVS107	0	0.00	100.00	0.00	400	171	I - - - R E - -	2006MAR18	04:13:00	VTSS8	00	

SUMMARY FOR MVCPOOL=CUST1POOL

ACS	MEDIA	FREE-MVCS VOLS	FREE-MVCS GB	RECLAIM-MVCS VOLS	RECLAIM-MVCS GB	USED-MVCS VOLS	USED-MVCS GB
00	ECART	120	96	2	0.5	90	45
00	STK1R	30	600	1	3.5	25	350

FIGURE 1-57 Example MVCPLRPT report (Part 1)

```
SUMMARY OF MVCS BY
USAGE:


137 TOTAL MVCS PROCESSED

135 INITIALIZED MVCS
PROCESSED

  2 UN-INITIALIZED MVCS
PROCESSE

 41 FREE MVCS AVAILABLE

  0 MVCS WITH STATUS AUDIT

  6 MVCS WITH STATUS DRAIN

  4 MVCS WITH STATUS EXPORT

  0 MVCS MARKED EJECTED

 60 MVCS MARKED FULL

  0 MVCS WITH MAXIMUM
VTVS

 82 MVCS MARKED READ-ONLY

  3 MVCS WITH STATUS
BROKEN

  7 MVCS WITH STATUS LOST

  0 MVCS MARKED RETIRED

  0 MVCS WITH EXPIRED
WARRANTY

  0 MVCS HAVE INVALID MIRS

  1 MVCS HAVE DATACHECKS

  5 MVCS WITH STATUS
CONSOLIDATE
```

FIGURE 1-58 Example MVCPLRPT report (Part 2)

MVCPLRPT Fields

The following list describes the Named MVC Pool report fields. The Summary fields are either for a Storage Class or a Named MVC Pool, depending on which was specified on the report JCL. If a Storage Class specifies a Named MVC Pool, the report gives information for that subpool.

MVC Volser

the MVC volser.

Number of VTVS

the number of current VTVs on the MVC. If the MVC has been used for VTV export, this field reports **EXPVTV**.

%Used

the percentage of the MVC used by current VTVs.

%Avail

the percentage of the MVC that is physically available for use.

%Frag

the percentage of the MVC that contains non-current VTVs. This space is not usable until it is reclaimed or the MVC is drained.

Media Size (MB)

the size of the MVC (MB). This will only be determined after VTCS has used an MVC. “UNKNOWN” appears in this field until VTCS migrates a VTV to the MVC.

Times Mounted

the number of times that the MVC has been mounted for writing or reading since it was added to the MVC inventory.

STATUS

one or more of the following statuses:

I

The MVC has been initialized.

B

The MVC has an error that should be investigated. The error may not make the MVC unusable, but VTCS will not select the MVC for migration for 12 hours after it is marked “B”. After the 12 hour period, the MVC will be least preferred for subsequent migrations, and recalls from the MVC may cause VTCS to drain it. This error condition may be accompanied by messages SLS6686, SLS6687, SLS6688, SLS6690, and/or SLS6693.

Any of the following conditions can cause this MVC error:

- MVC corrupted by another job (other than VTCS/VTSS).
- Attempt to use a read-only MVC for migration.
- A DDR swap failure.
- An RTD failure.

L

The MVC was not mounted in response to the last mount request. The MVC can still be used for migration, but will not select the MVC for migration for 12 hours after it is marked “L”. After the 12 hour period, the MVC will be least preferred. This condition will clear itself the next time that the MVC is mounted.

D

A data check was reported for this MVC. VSM will not use this MVC again for migration.

R

the MVC has been marked read-only.

U

one of the following usage statuses:

U

the MVC is available for output (migration, reclamation, export, or consolidation).

-

the MVC is not available for output (migration, reclamation, export, or consolidation).

A

The MVC is either being audited or the audit failed. If the audit failed, VTCS will not use the MVC for migration. To clear this condition, rerun the AUDIT against this MVC.

C

The MVC is a consolidation MVC.

E

The MVC is an export MVC.

F

There is no space available on the MVC.

J

Either you issued a MVCDRain EJect for the MVC or the MVC was ejected for update by a RACROUTE call. The MVC will not be used again for migration or recall. To clear this condition, use MVCDRain against MVC without the EJect option.

N

Either:

- The MVC is being drained because of:
  - An automatic drain or demand reclaim.
  - An explicit MVCDRain command.

OR

- The previous DRAIN request failed, in which case VTCS will not use the MVC for migration. To clear this condition, enter MVCDRain against MVC without the EJect option.

X

The MVC has reached the maximum VTVs per MVC.

T

One of the following statuses:

T

The MVC is retired.

W

The MVC's warranty has expired.

M

The MVC has an invalid MIR.

Last Mounted

the date and time that the MVC was last mounted and the VTSS where the MVC was last used.

ACS ID

the ACS where the MVC resides.

Owner/Consolidate Time

If the MVC is empty, this field is null. If the MVC is a consolidation MVC, this field displays the time of the consolidation. If the MVC is a migration MVC and contains current VTVs, this field displays the MVC's Storage Class. If no Storage Class was explicitly assigned via the MGMTclas statement, the default Storage Class is the name of the last VTSS that wrote to the MVC for reclamation or migration.

If VTCS receives a request to migrate a VTV that is assigned to an invalid Management Class, VTCS will dynamically create the !ERROR Storage Class and migrate the VTVs defined by the invalid Management Class to the !ERROR Storage Class. Use this Storage Class to identify and correct invalid Management Classes, drain the affected MVCs, and resubmit the request.

Summary for Storage Class or Named MVC Pool

This section shows number of MVCs (**Vols**) and total storage (**Gb**) by ACS and media type for the following categories:

Free-MVCs

MVCs that have 100% usable space and do not contain any migrated VTVs. The storage shown is the total free space based on media type capacity.

Reclaim-MVCs

MVCs eligible for space reclamation. The storage shown is the total wasted space including those MVCs not yet eligible for space reclaim.

Used-MVCS

Initialized MVCs that are partially or completely full.

Total MVCs

Total MVCs for the Storage Class or Named MVC Pool with subtotals for initialized, uninitialized, and free MVCs.

Summary of MVCs by Usage

This section shows number of MVCs by the task that last used the MVC.

MVCRPT

The MVCRPT reports the status of your VSM system's MVCs.

Syntax

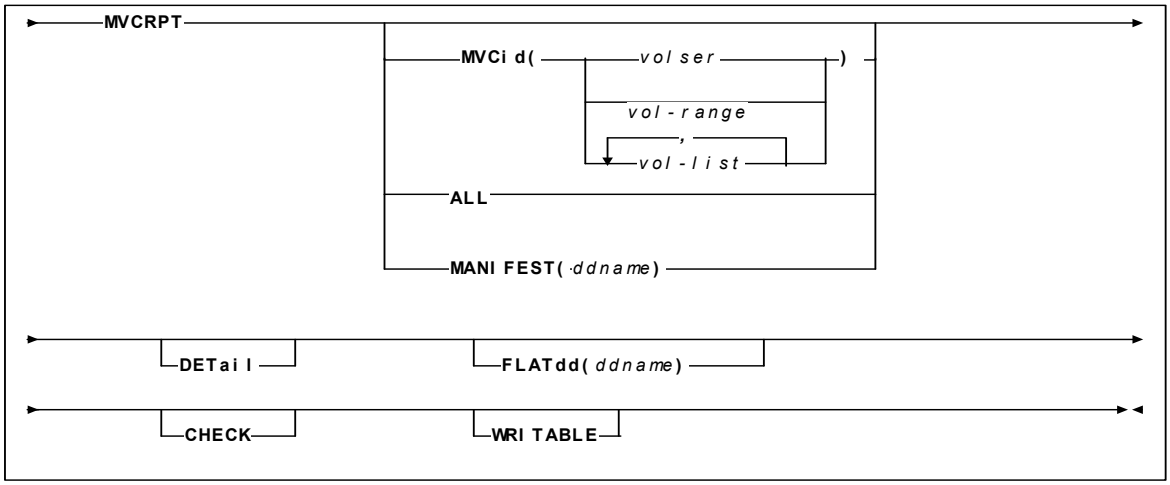


FIGURE 1-59 MVCRPT syntax

Parameters

MVCid

specifies the MVCs for the report. If you do not specify the MVCs, the report includes all MVCs in your VSM system.

volser, vol-range or *vol-list*

the volsers of one or more MVCs.

ALL

generate a report for all MVCs.

MANIFEST

specifies the input ddname of the manifest file used to generate the report.

ddname

ddname of the manifest file. **Note that** you can specify a merged manifest file or multiple manifest files.

DETail

produce a detailed MVC report; see FIGURE 1-61 on page 145 for an example. If you do not specify this option, the default is to produce a summary MVC report; see FIGURE 1-60 on page 140 for an example.

FLATdd

specifies the destination of the optional flat file output.

ddname

name of the DD in the JCL that describes the output data set if a flat file is required (must be RECFM=VB,LRECL=75).

CHECK

MVCRPT does limited MVC integrity checking, during which the CDS is reserved. Use **only** if directed by StorageTek Software Support.

WRITABLE

MVCRPT ignores readonly MVCs on MVC detailed reports and when you specify the CHECK parameter.

Interfaces

SWSADMIN utility only.

JCL Requirements

The following are the required and optional statements for the MVCRPT JCL:

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the report.

SLSIN

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

SYSOUT

specifies the output destination for SORT messages. This is only required for DETAIL MVC reports.

MVC Reports

The following sections describe the MVC summary and detailed reports that the MVCRPT produces.

MVC Summary Report

FIGURE 1-60 shows an example of an MVC summary report.

Note – The sort order on the MVC summary report the order in which the MVCs were defined to the system.

SWSADMIN (6.2.0) STORAGETEK VTCS SYTEM UTILITY PAGE 0002

TIME 09:26:54 VTCS MVC SUMMARY REPORT DATE 2006-04-13

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	CONSOLIDATE TIME
EVS99	200	10.80	84.57	4.63	20.0	310	I - - - - U - M	2006MAR15	03:20:23	VTSS8	00	S1
EVS100	0	0.00	100.00	0.00	UNKNOWN	206	- - L - - U - -	2006MAR10	05:24:04	VTSS8	--	
EVS101	1009	99.00	0.00	1.00	4.0	306	I - - - - U - -	2006MAR15	03:20:23	VTSS8	00	S1
EVS102	5	8.25	91.75	0.00	4.0	6	I - - - - U - -	2006MAR15	04:23:04	VTSS8	00	S3
EVS103	EXPVTV	0.12	99.88	0.00	4.0	194	I - - - - J - -	2006MAR15	03:20:28	VTSS10	00	VTSS10
EVS104	0	0.00	100.00	0.00	4.0	5	I - - - R C - -	2006MAR18	03:49:14	VTSS8	00	2006APR12 03:49:14
EVS104	200	10.80	84.57	4.63	1020.40	254	I - - - R U T -	2006MAR18	04110:09	VTSS8	00	
EVS105	300	15.80	54.57	4.63	1020.40	154	I - - - R U W -	2006MAR18	04110:09	VTSS8	00	
EVS106	0	0.00	100.00	0.00	4.0	202	I - - - - C - -	2006MAR18	03:49:20	VTSS8	00	
EVS107	0	0.00	100.00	0.00	4.0	171	I - - - R E - -	2006MAR18	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	.4	1	I - - - - U - -	2009MAR18	04:13:00	VTSS8	SECNDARY	
EVS480	1	10.8	84.57	4.63	2.50	10	I - - - - U - -	2009MAR18	04:13:00	VTSS8	LEPRIM	

137 Initialized MVCs processed

8 Non-Initialized MVCs processed

FIGURE 1-60 Example MVC summary report

In FIGURE 1-60 on page 140 **note that**:

- Volume media size is now reported in GB.
- Although VMVCs in a ELS 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 ELSs, 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 and to the output of `MVCPLRPT`.

The following list describes the MVC summary report fields.

MVC Volser

the MVC volser.

Number of VTVS

the number of current VTVs on the MVC. If the MVC has been used for VTV export, this field reports **EXPVTV**.

%Used

the percentage of the MVC used by current VTVs.

%Avail

the percentage of the MVC that is physically available for use.

%Frag

the percentage of the MVC that contains non-current VTVs. This space is not usable until it is reclaimed or the MVC is drained.

Media Size (MB)

the size of the MVC (MB). This will only be determined after VTCS has used an MVC. “UNKNOWN” appears in this field until VTCS migrates a VTV to the MVC.

Times Mounted

the number of times that the MVC has been mounted for writing or reading since it was added to the MVC inventory.

STATUS

one or more of the following statuses:

I

The MVC has been initialized.

B

This is a generic error that indicates the MVC, drive, or combination of the two has a problem. VTCS attempts to de-preference MVCs with this state. To clear this state:

If the MVC caused the problem, use a DRAIN(EJECT) command to remove the MVC from service.

If the RTD caused the problem, use the MVCMAINT utility to reset the MVC state.

Note also that one or more of the following messages is issued for BROKEN status: SLS6686, SLS6687, SLS6688, SLS6690. For detailed recovery procedures for these messages, see *VTCS Messages and Codes*.

L

VTCS attempted to mount an MVC and the mount did not complete within a 15-minute time-out period. VTCS is attempting to recover from a situation that may be caused by hardware problems, HSC problems, or by the MVC being removed from the ACS. VTCS attempts to de-preference MVCs with this state.

If VTCS does perform a subsequent successful mount of an MVC with LOST(ON) state, VTCS sets the state to LOST(OFF).

Determine the cause of the error and fix it. You can also use the VTCS MVCMAINT utility to set LOST(OFF) for the following events:

LOST(ON) was set due to LSM failures or drive errors that have been resolved

LOST(ON) was set because the MVC was outside the ACS and has been reentered.

D

A data check condition has been reported against this MVC. VTCS attempts to de-preference MVCs with this state. To clear this state:

If all VTVs on the MVC are duplexed, use MVCDRain on the MVC without the Eject option. This recovers all VTVs and removes the MVC from service.

If all VTVs on the MVC are not duplexed, VTCS AUDIT the MVC. The audit will probably fail. After the audit, do an MVCDRAIN (no eject). This recalls the VTVs before the data-check area in ascending block-id order and the VTVs after the data-check area in a descending block-id order. Processing the VTVs in this sequence ensures that VTCS recovers as many VTVs as possible from the media. You then need to recreate the data for any VTVs still on the MVC.

R

The MVC has been marked read-only because of one of the following conditions:

The MVC being the target of an export or consolidation process. The read-only state protects the MVC from further updates.

The MVC media is set to file protect. Correct the error and use the MVCMAINT utility to set READONLY(OFF).

The MVC does not having the appropriate SAF rules set to enable VTCS to update the MVC. Correct the error (for more information, see “Defining a Security System User ID for HSC, SMC, and VTCS” in *Installing ELS*) and use the MVCMAINT utility to set READONLY(OFF).

U

one of the following usage statuses:

U

the MVC is available for output (migration, reclamation, export, or consolidation).

-

the MVC is not available for output (migration, reclamation, export, or consolidation).

A

The MVC is either being audited or the audit failed. If the audit failed, VTCS will not use the MVC for migration. To clear this condition, rerun the AUDIT against this MVC.

C

The MVC is a consolidation MVC.

E

The MVC is an export MVC.

F

There is no space available on the MVC.

J

Either you issued MVCDRain EJect for the MVC or the MVC was ejected for update by a RACROUTE call. The MVC will not be used again for migration or recall. To clear this condition, use MVCDRain against MVC without the EJect option.

N

Either:

- The MVC is being drained because of:
  - An automatic drain or demand reclaim.
  - An explicit MVCDRain command.

OR

- The previous DRAIN request failed, in which case VTCS will not use the MVC for migration. To clear this condition, enter MVCDRain against MVC without the EJect option.

X

The MVC has reached the maximum VTVs per MVC.

T

One of the following statuses:

T

The MVC is retired. VTCS will recall from, but not migrate to, the MVC. Replace the MVC as soon as possible.

W

The MVC's warranty has expired. VTCS continues to use the MVC. You should start making plans to replace the MVC when it reaches Retired state.

M

VTCS has received status from an RTD to indicate the MIR (media information record) for a 9x40 media is invalid. An invalid MIR does not prevent access to data but may cause significant performance problems while accessing records on the tape. The MVC is not capable of high-speed searches on areas of the tape that do not have a valid MIR entry.

VTCS attempts to de-preference MVCs with this condition. For recalls, if the VTV resides on multiple MVCs, VTCS selects MVCs with valid MIRs ahead of MVCs with invalid MIRs. VTCS avoids using MVCs with invalid MIRs for migration, unless the migration is at the beginning of the tape. Migrating from the beginning of tape will correct the MIR.

VTCS detects the invalid MIR condition at either mount time or dismount time. If detected at mount time and the operation can be completed with another MVC, VTCS dismounts the first MVC and selects the alternate MVC. **Note that** VTCS has only a limited ability to switch to an alternate MVC. That is, it is mainly used for migrate and virtual mount.

For MVCs with invalid MIRs, determine the cause of the error, which may be caused by media or drive problems, and fix the error.

To recover an MVC with an invalid MIR, you simply need to read the MVC to the end of the tape, which can be done via a VTCS audit. If the media is the problem, run an MVCDRAIN EJECT to recall the VTVs and cause the MVC to be removed from the MVC pool.

Last Mounted

the date and time that the MVC was last mounted and the VTSS where the MVC was last used.

ACS ID

the ACS where the MVC resides.

Owner/Consolidate Time

If the MVC is empty, this field is null. If the MVC is a consolidation MVC, this field displays the time of the consolidation. If the MVC is a migration MVC and contains current VTVs, this field displays the MVC's Storage Class. If no Storage Class was explicitly assigned via the MGMTclas statement, the default Storage Class is the name of the last VTSS that wrote to the MVC for reclamation or migration.

If VTCS receives a request to migrate a VTV that is assigned to an invalid Management Class, VTCS will dynamically create the !ERROR Storage Class and migrate the VTVs defined by the invalid Management Class to the !ERROR Storage Class. Use this Storage Class to identify and correct invalid Management Classes, drain the affected MVCs, and resubmit the request.

MVC Detailed Report

The MVC detailed report provides all the fields from the MVC summary report and a separate section that lists additional fields. FIGURE 1-61 shows an example of these additional fields from an MVC detailed report.

SWSADMIN (6.2.0) STORAGETEK VTCS SYTEM UTILITY PAGE 0003

TIME 11:28:30 MVC EVS102 DETAIL REPORT DATE 2006-06-03

VTV VOLSER	SIZE (MB)	BLOCK ID	MANAGEMENT CLASS	MIGRATION DATE	BLOCK COUNT	MESSAGE
X20041	76.00	00000000	M5	2006JAN08	10	
X20043	76.00	134009C7	M5		9	
X20044	76.00	2A40138D	M5	2006JAN08	9	
X20045	76.00	C6401D53	M3		10	
X20047	76.00	A5402719	M3		10	

5 VTVS FOUND FOR MVC:EVS102

WARNING VTV COUNT:5 DOES NOT MATCH MVC SUMMARY RECORD VTV COUNT:22 FOR MVC:EVS102

FIGURE 1-61 Example MVC detailed report (additional fields)

MVC Detailed Report Additional Fields

The following list describes the additional fields for the MVC detailed report.

VTV Volser

the volsers of the VTVs on the MVC.

Size (MB)

the uncompressed size of the VTV (MB).

Block ID

the logical block ID of the beginning of the VTV on the MVC.

Management Class

the VTV's Management Class.

Migration Date

for migrates done by a 6.0 system or above **only**, the approximate date that the VTV copy was migrated. This date is approximate because it is recorded in the CDS as the number of days since VTV creation, and time zone adjustments can cause the value to slip a day.

Block Count

the decimal number of blocks of data that the VTV occupies on the MVC.

Message

reports the results of MVC integrity checking. Any message **not** listed below should be reported to StorageTek, as it may indicate a serious problem with the CDS **except** messages that can appear if the MVC is being audited or has failed an audit while the MVC report is being run. An MVC report returns a minimum return code of 4 if any MVCs in audit state are encountered; the audit state also prevents some other errors from being reported.

nnnn empty space to previous

Explanation. Informational message, indicating that there are *nnnn* (decimal) blocks of free space before this VTV on the MVC.

Migrated size was not as expected

Explanation. If output for VTV v on MVC m, it indicates that the (compressed) size of VTV v migrated to MVC m was different to the (compressed) size of VTV v recorded in the CDS.

This is normal if VTV v had previously been recalled with error. Otherwise this indicates a possible error which should be reported to StorageTek software support

Flat File Record Format

TABLE 1-5TABLE 1-5 shows the record format of the flat file produced by MVCRPT.

TABLE 1-5 MVCRPT Flat File

Decimal Offset	Hexadecimal Offset	Type	Length	Description
0	0	start of record		start of MVC flat file record
0	0	integer	4	record length
4	4	character	1	character set type of text fields
		X'61'		ASCII
		X'6E'		EBCDIC
5		character	1	record type'M' (indicates an MVC report)
6	5	character	6	MVC volser
12	C	integer	4	number of current VTVs on the MVC
16	10	integer	4	percentage of the MVC used by current VTVs
20	14	integer	4	percentage of the MVC that is available for use
24	18	integer	4	percentage of the MVC that contains non-current VTVs, which is not available for use until it is reclaimed or the MVC is drained
28	1C	integer	4	number of times that the MVC has been mounted for writing or reading since it was added to the MVC inventory
32	20	time_t	4	TOD MVC was last Mounted (starting from 01/01/1970)
36	24	integer	4	size of the MVC (MB)
40	28	time_t	4	Consolidation date/time (time_t format) or X'00'
44	2C	character	1	MVC exported (Y or N)
45	2D	character	1	MVC initialized (I or -)
46	2E	character	1	MVC broken (B or -)
47	2F	character	1	MVC lost (L or -)
48	30	character	1	MVC has data check (D or -)
49	31	character	1	MVC readonly (R or -)
50	32	character	1	MVC Usage status: - Not usable A AUDIT status CSet CONSOLIDATE status E EXPORT status FFULL status JEJECT status NDRAIN status UUsable
51	33	character	1	MVC Retired (T or -)
52	34	character	1	MVC has invalid MIR (M or -)
53	35	character	2	ACS location of MVC
55	37	character	8	MVC was last mounted on this VTSS
63	3F	character	8	Owning VTSS name or Storage Class name

RECALL

RECALL does demand recalls of VTVs to a VTSS.

Syntax

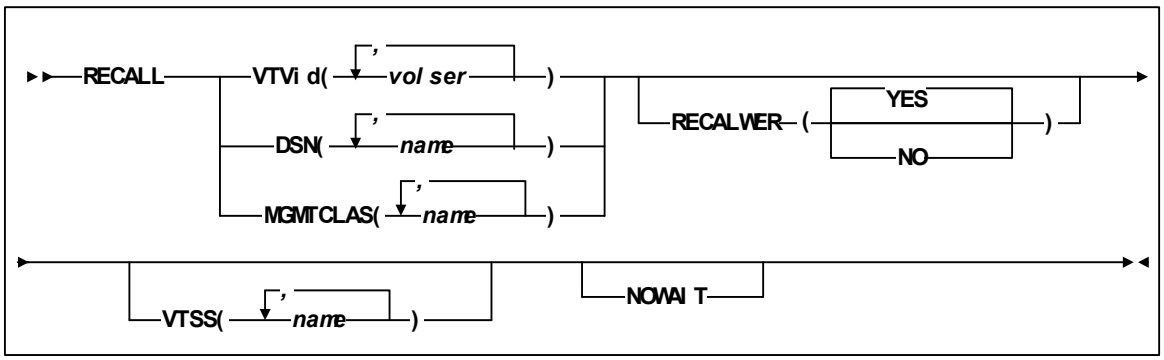


FIGURE 1-62 *RECALL* syntax

Parameters

VTVid

specifies the VTVs that VSM recalls.

volser, vol-range, or *vol-list*

the volsers of one or more VTVs. You can also specify one or more ranges.

DSN

specifies data sets used to select VTVs to recall.

name

the data set name. Table 3 on page 90 describes the valid wildcards for data set names.

MGMTCLAS

specifies one or more Management Classes that determine one or more VTVs to recall.

mgmt-class-name | *mgmt-class-list*

the names of one or more Management Classes that you defined on the MGMTclas control statement; for more information, see "MGMTCLAS Control Statement" on page 234.

Note – The VTVid, DSN, and MGMTCLAS parameters are mutually exclusive.

VTSS

specifies where the VTVs are recalled as follows:

- If you do not specify a VTSS (the default), VTCS attempts to recall the VTVs to the VTSS of creation if it is accessible. Otherwise VTCS recalls the VTVs to the VTSS with the lowest DBU.
- If you specify a single VTSS, VTCS attempts to recall the VTVs to the specified VTSS if it is accessible. Otherwise, VTCS recalls the VTVs to the VTSS with the lowest DBU.
- If you specify a list of VTVs, VTCS attempts to recall the VTVs to the VTSS of creation if it is on the list and accessible, otherwise VTCS recalls the VTVs to the VTSS with the lowest DBU on the list.

vtss-name

the names of one or more VTSSs.

RECALWER

specifies whether VTCS recalls VTVs with read data checks.

YES

recall VTVs with read data checks (the default).

NO

Do not recall VTVs with read data checks.

NOWAIT

specifies that the utility does not wait for the operation to complete and returns after the request is submitted.

Interfaces

SWSADMIN utility and VT command.

JCL Requirements

The following are the required and optional statements for the RECALL JCL:

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the report.

SLSIN

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

RECLAIM

RECLAIM does demand MVC space reclamation. The RECLAIM can also override the CONFIG RECLAIM settings for the THRESHLD, MAXMVC, and CONMVC parameters.

Syntax

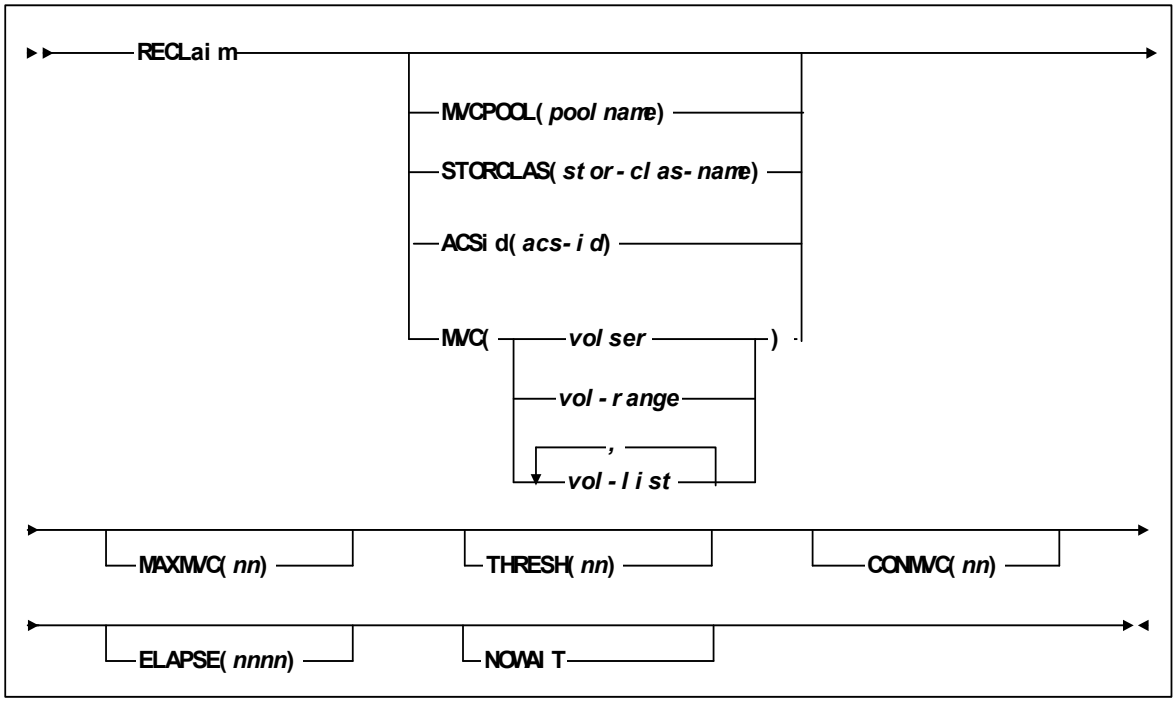


FIGURE 1-63 RECLAIM syntax

Parameters

MVCPOOL

reclaim the MVCs in the specified Named MVC Pool.

poolname

the name of a Named MVC Pool that you defined on the MVCPool control statement; for more information, see "MVCPOOL Control Statement" on page 249.

STORCLAS

reclaim the MVCs in the specified Storage Class.

stor-class-name

the name of a Storage Class that you defined on the STORclas control statement; for more information, see "STORCLAS Control Statement" on page 252.

ACSid

reclaim the eligible MVCs in the specified ACS.

acs-id

The specified ACS.

MVC

reclaim the specified MVC(s).

volser, vol-range or *vol-list*

the volsers of one or more MVCs.You can also specify individual MVCs and ranges in a list. For example:

(MVC000-MVC005,MVC010,MVC015)

MAXMVC(*nn)*

specifies the maximum number of MVCs that will be processed by a single space reclamation task. Valid values are 1 to 98. There is no default; if not specified, the CONFIG RECLAIM value (or default) is used.

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

THRESH(*nn)*

specifies the percentage of fragmented space that makes an MVC eligible for demand or automatic reclamation. Valid values are 4 to 98. If not specified, the CONFIG RECLAIM value (or default) is used.

NOWAIT

specifies that the utility does not wait for the operation to complete and returns after the request is submitted.

CONMVC(*nn)*

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

Valid values are 1 to 99. If not specified, the default is the CONMVC value specified on the CONFIG RECLAIM statement.

ELAPSE(*nnnn)*

specifies the maximum time for the reclaim in minutes. If the maximum time expires, VTCS issues message SLS6682I. If there are no MVCs currently mounted, reclaim stops when the ELAPSE value is reached. If any MVCs are currently mounted when the ELAPSE value is reached, reclaim processes the mounted MVCs and then stops.

Valid values are 1 to 1440. If not specified, there is no time limit on the reclaim process.

Interfaces

SWSADMIN utility and VT command.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the report.

SLSIN

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

RECONcil

RECONcil reconciles VTV media and location (moves VTVs from one Storage Class to another).

Note – RECONcil is valid only if FEATures VSM(ADVMGMT) is specified.

Syntax

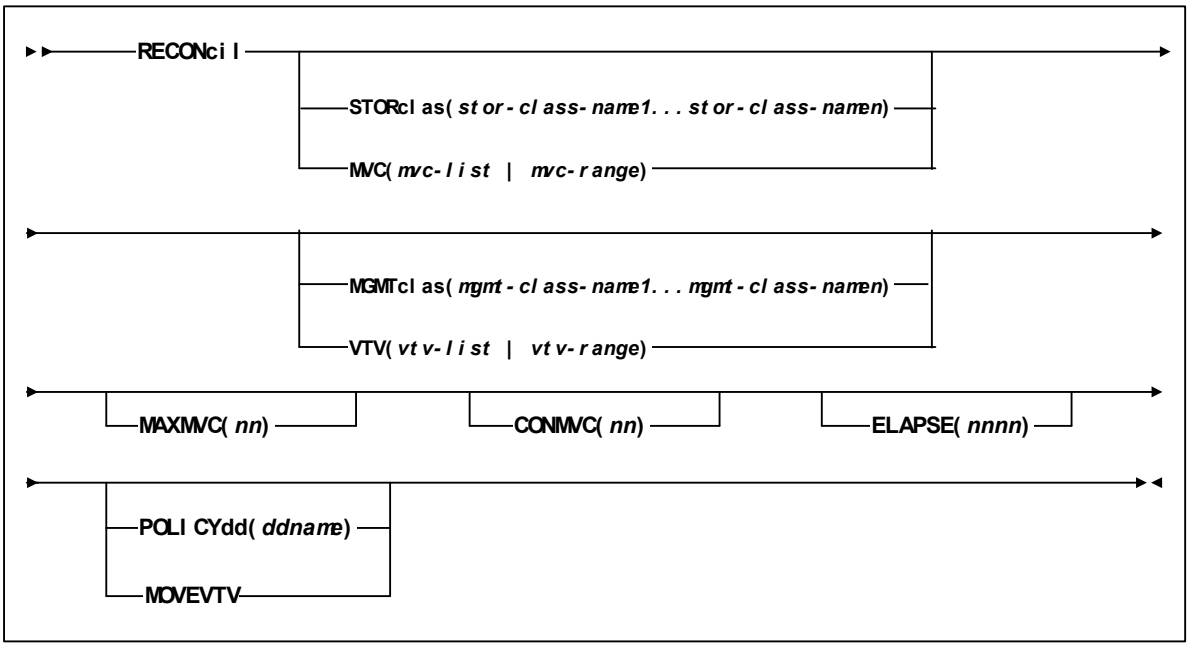


FIGURE 1-64 RECONcil syntax

Parameters

STORclas

Specifies one or more Storage Classes.

stor-class-name1...stor-class-namen

One or more Storage Classes.

MVC

Specifies a list or range of MVCs.

mvc-list | *mvc-range*

A list or range of MVCs.

MGMTclas

Specifies one or more Management Classes.

mgmt-class-name1...mgmt-class-namen

One or more Management Classes.

VTV

Specifies a list or range of VTVs.

vtv-list | vtv-range

A list or range of VTVs.

MOVEVTV

Move VTVs per the currently active Management Policies as specified by the MGMTclas statements that apply to the VTVs. If you do not specify MOVEVTV, only a report is generated.

POLICYdd(*ddname)*

Specifies the *ddname* of a file containing an alternate MGMTclas statement.This option forces the report only mode and is mutually exclusive with MOVEVTV.

MAXMVC(*nn)*

specifies the maximum number of MVCs that will be processed by a single reconciliation task.

Valid values are 1 to 98. If not specified the CONFIG RECLAIM value (or default) is used. This parameter is ignored if MOVEVTV is not specified.

CONMVC(*nn)*

specifies the maximum number of MVCs that VTCS concurrently processes during subsequent recall and migrate operations.

Valid values are 1 to 99. If not specified the CONFIG RECLAIM value (or default) is used. This parameter is ignored if MOVEVTV is not specified.

ELAPSE(*nnnn)*

specifies the maximum time for the reconciliation in minutes. If the maximum time expires, VTCS issues message SLS6682I. If there are no MVCs currently mounted, reconciliation stops when the ELAPSE value is reached. If any MVCs are currently mounted when the ELAPSE value is reached, reconciliation processes the mounted MVCs and then stops.

Valid values are 1 to 1440. If not specified, there is no time limit on the reconciliation process. This parameter is ignored if MOVEVTV is not specified.

Usage Notes

The RECONCIL command does a series of cross-checks between the MVCs, VTVs, STORCLAS and MGMTCLAS definitions.

For VTVs, this process involves checking the placement and number of MVC copies. If there is a mismatch, then all of the correct copies for a VTV are established. For MVCs, this process involves checking their membership of the storage class is correct. If there is a mismatch, then the MVC is emptied. Both analysis phases result in a list of VTV copies to be processed.

The MVC/STORCLAS parameters merely act as filters to the inspection of the MVCs to Storage Class membership. Likewise the VTV/MGMTCLAS parameters act as filters to the inspection of VTVs and comparing them to their Management Class definitions. Both or either parameter sets can be used.

RECONCIL has two phases. The first phase selects MVC & Storage Classes that are going to be scanned and for this part the MVCs are checked to ensure they meet the Storage Class definitions and ACS definitions. If there are inconsistencies then the VTVs on those MVCs are added to the list of VTVs to be recalled and remigrated to give correct MVC, ACS & Storage Class copies for that VTV.

The second phase selects the VTVs and Management Classes so that the correct number and location of the VTV copies are established. If any VTVs show as inconsistent then these are added to the list of VTVs to be recalled and remigrated.

The RECONCIL parameters STORCLAS & MVC limit the first phase of checking to the specified Storage Classes or MVCs. If either parameter is not specified then the default is all MVCs are checked.

The RECONCIL parameters MGMTCLAS & VTV limit the second phase of checking to the specified Management Classes or VTVs. If either parameter is not specified the default is all VTVs are checked.

Interfaces

SWSADMIN utility only.

JCL Requirements

The following are the required and optional statements for the RECONcil JCL:

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the report.

SLSIN

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

RECONcil Reports

RECONCIL Report (MOVEVTV Not Specified)

FIGURE 1-65 shows an example of an RECONcil report.

SWSADMIN (6.2.0) STORAGETEK VTCS SYTEM UTILITY PAGE 0001

TIME 10:07:10 RECONCILE VTV REPORT DATE 2006-11-20

VTV	SIZE	COMP%	<----------CREATION---------->		MGMT					
VOLSER	(MB)		DATE	TIME	CLASS	MVC1	MVC2	MVC3	MVC4	REASON
Y00001	1.62	39	2006JAN15	04:11:18	MJ4	02250	02251	02252	02253	LOCATION
Y00002	1.62	39	2006JAN15	04:03:57	MJ4	02250	02251	02252	02253	LOCATION
Y00003	1.62	39	2006JAN15	03:50:59	MJ3	02254	02255			MEDIA TYPE
Y00004	1.62	39	2006JAN15	03:45:04	MJ2	02256	02257	022568		MEDIA TYPE
Y00005	0.01	0	2005DEC17	23:56:00	MJ1	02259				ARCHIVE
Y00006	0.01	0	2005DEC17	23:41:37	MJ1	02259				ARCHIVE
Y00007	1.62	39	2006JAN06	06:15:46	MJ4	02250	02251	02252	02253	LOCATION

TIME 10:07:10 RECONCILE MVC REPORT DATE 2006-111-20

MVC	MEDIA	MEDIA	STORAGE CLASS/		LOCATIO N		CANDID ATE		TOTAL VTV
VOLSER	TYPE	SIZE (MB)	CONSOLIDATE TIME		(ACS ID)		VTVS		SIZE (MB)
022550	9840	20000	SC1		00		3		4.86
Y00001	Y00002	Y00007							
022551	9840C	40000	SC3				3		4.86
Y00001	Y00002	Y00007							
022559	9940A	60000	2006Aug01 17:59:58				17		18.50
Y00005	Y00006	Y00015	Y00027	Y00042		Y00048		Y00053	
Y00059	Y00061	Y00067	Y00073	Y00078		Y00084	Y00088	Y00101	Y00123

TOTAL MVCS=8 TOTAL SIZE=29.32MB

FIGURE 1-65 Example output from RECONcil

RECONcil Report Fields

The following list describes the RECONcil report fields.

VTV Volser

the VTV volser.

Size (MB)

the uncompressed size of the VTV (MB). **<MOUNT>** indicates that the VTV was mounted when the report ran. **<FENCED>** indicates that the VTV's state is unknown. If **<FENCED>** appears, contact StorageTek software support.

Comp %

the VTV compression percentage achieved. This is the difference between the uncompressed and compressed VTV size expressed as a percentage of the uncompressed VTV size. For example if a 100MB VTV compresses to 40MB then the compression% will be given as 60%. A compression of 0% indicates that no compression was possible on the VTV.

Creation Date and Time

the date and time that the VTV was created.

MGMT Class

the name of the Management Class for the VTV specified.

MVC1, MVC2, MVC3, MVC4

the MVC(s) that contain the VTV. If all of these fields are empty, the VTV has not been migrated or consolidated. If 1 or more of these fields list an MVC volser, the VTV was migrated to each of these MVCs.

Reason

the reason for the VTV selection: archive, media type, or media location.

MVC Report Fields

The following list describes the RECONcil MVC report fields. The data for each MVC is followed by one or more VTV volsers with copies on the MVC. This section of the report is followed by a total line showing the number of candidate MVCs and the size in MB to be recalled and remigrated.

MVC Volser

the MVC volser.

MVC Media Type

the MVC type.

Media Size (MB)

the size of the MVC (MB). This will only be determined after VTCS has used an MVC. "UNKNOWN" appears in this field until VTCS migrates a VTV to the MVC.

Storage Class/Consolidate time

the MVC's Storage Class or time the VTVs on it were consolidated.

Candidate VTVs

the number of candidate VTVs on the MVC.

Total VTV Size (MB)

the size of all candidate VTVs on the MVC in MBs.

Location (ACS ID)

the ACS where the MVC resides. If blank, the MVC is not currently in an ACS.

MOVEVTV Report

FIGURE 1-66 shows an example of an RECONcil report with the MOVEVTV option.

```
SWSADMIN (6.2.0) STORAGETEK VTCS SYSTEM UTILITY PAGE 0001
 TIME 10:07:10               RECONCILE MOVEVTV REPORT      DATE 2006-01-16


Move VTV - MVC 022705 ignored, MAXMVC reached
Move VTV - VTV X04898 ignored, all MVC copies rejected
Move VTV - 4 MVCs selected for processing
Move VTV - 5 VTVs selected for processing
Move VTV - 5 VTV copies to be processed
Move VTV - 0 VTV copies not matched to request
Move VTV - 1 VTV copies rejected by MAXMVC limit
Move VTVs - MVC 023484 selected and contains 1 VTVs
Move VTVs - MVC 022628 selected and contains 1 VTVs
Move VTVs - MVC 022631 selected and contains 2 VTVs
Move VTVs - MVC 022608 selected and contains 1 VTVs

Recall from MVC 022628 to VTSS HBVTSS17
SLS6683I Bulk recall of 1 VTVs issued to MVC 022628
SLS6644I VTV X99909 recalled from MVC:022628 Block:25401431
SLS6637I Recall from MVC 022628 completed

Recall from MVC 023484 to VTSS HBVTSS17
SLS6683I Bulk recall of 1 VTVs issued to MVC 023484
SLS6644I VTV X04897 recalled from MVC:023484 Block:02402581
SLS6637I Recall from MVC 023484 completed

Recall from MVC 022608 to VTSS HBVTSS16
SLS6683I Bulk recall of 1 VTVs issued to MVC 022608
SLS6637I Recall from MVC 022608 completed

Migrate to storage class HBVTSS16 from VTSS HBVTSS17
SLS6681I VTV X99909 migrated to MVC:022589 Block:01400025 StorCl:HBVTSS17 MgmtCl:SIMPLEX
SLS6636I Demand migration to MVC 022589 completed

Recall from MVC 022631 to VTSS HBVTSS16
SLS6683I Bulk recall of 2 VTVs issued to MVC 022631
SLS6644I VTV X99910 recalled from MVC:022631 Block:03400141
SLS6644I VTV X99908 recalled from MVC:022631 Block:05400281
SLS6637I Recall from MVC 022631 completed
```

FIGURE 1-66 Example output from RECONcil MOVEVTV

RTV Utility

The RTV utility converts VTVs contained on MVCs to data sets on Nearline volumes (real tape volumes).

Syntax

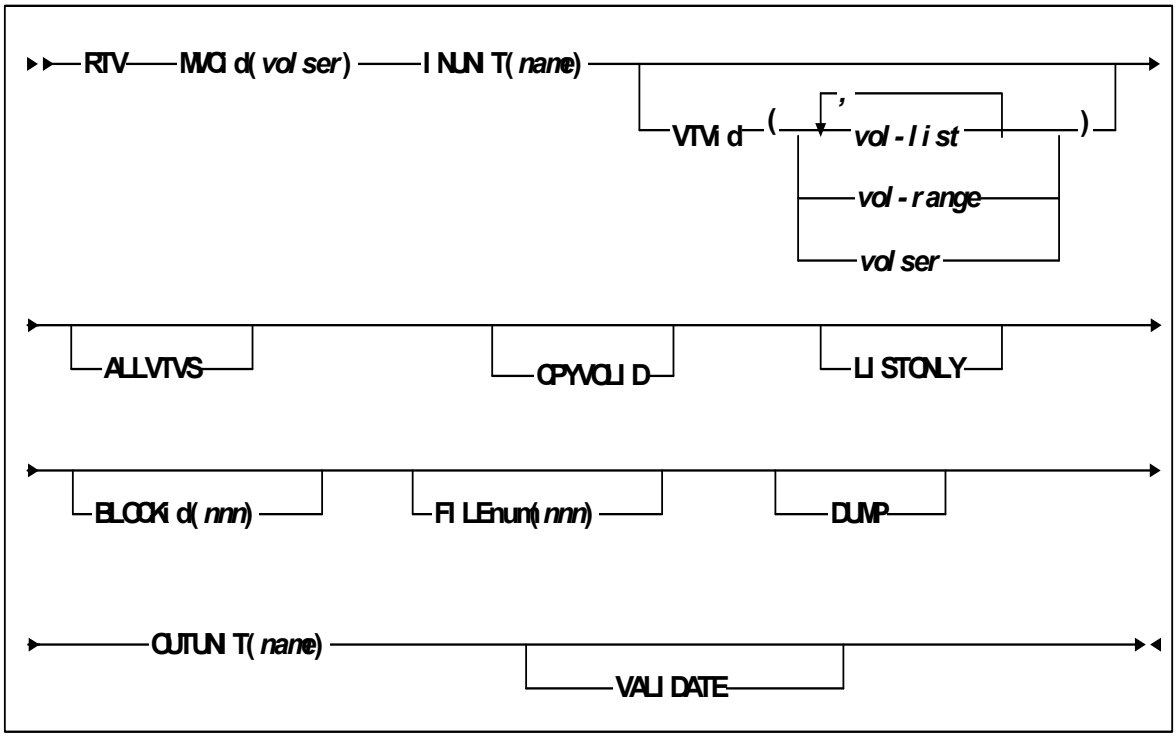


FIGURE 1-67 RTV utility syntax

Parameters

MVCid

specifies the MVC that contains the VTVs that RTV converts to Nearline volume(s) which become real tape versions of the VTVs.

volser

the MVC volser.

INUNIT

the name to use to allocate the input tape unit. You can specify an MVS unit address, an esoteric name, or a generic name. The valid values are the same as for the UNIT= JCL parameter.

name

the unit name.

VTVid

specifies one or more VTVs to convert.

volser, vol-range, or *vol-list*

the volsers of one or more VTVs.

ALLVTVS

convert the most current copy of all VTVs on the specified MVC. That is, if there are multiple copies of a VTV on the specified MVC, RTV only converts the most current copy of the VTV.

Note – The VTVid and ALLVTVS parameters are mutually exclusive.

CPYVOLID

copy the VTV internal volser from the VTV to the output volume VOL1 record. The default is to not copy the VTV VOLID.

Caution – Use the CPYVOLID parameter carefully! The volser of the output tape will be changed to the volser of the VTV. If the output tape has an external label or if the output is directed to another VTV, this will cause label mismatches and can cause unpredictable and undesirable results.

Note –

- If the output tape is non-labelled or has a non-standard tape label, CPYVOLID will be automatically specified for this VTV decompression, and a standard label tape will be created on the output device.
- If the input VTV contains standard or ANSI labels and CPYVOLID is not specified, RTV processes these label types as described in TABLE 1-6. Note that this only applies to the VOL1 record. The HDR1/HDR2 labels (if they exist) are always copied from the VTV by RTV for every VTV processed.
- If the input VTV is non-labeled and CPYVOLID is not specified, RTV processes these label types as described in TABLE 1-6.

TABLE 1-6 RTV Label Types

VTV Label Type	Output Standard Label	Output ANSI Label	Output Non-Label
Standard label	VOL1 label not copied	WTOR issued	VOL1 label is copied
ANSI label	WTOR issued	VOL1 label not copied	VOL1 label is copied
Non-Label	WTOR issued	WTOR issued	N/A – no VOL1 label

In TABLE 1-6, the WTOR is as follows:

SWSRTV - Label mismatch - Reply RELABEL, RETRY, or CANCEL

The operator responses produce the following results:

RELABEL

Decompress the RTV and overwrite the volser on the output volume.

RETRY

Mount another output volume and retry the operation.

CANCEL

Do not decompress the RTV.

LISTONLY

lists (but does not convert) the VTVs on the specified MVC. For more information, see "RTV LISTONLY Listing" on page 169.

BLOCKID

the logical block ID where the VTV begins on the MVC.

nnn

the logical block ID (8 hexadecimal characters).

Tip – The LISTONLY parameter listing on "RTV LISTONLY Listing" on page 169 supplies a Block ID value that you can use as input to the RTV utility to convert a VTV to a Nearline volume.

FILEnum

the logical data set number of VTV on the MVC.

nnnnn

the logical data set number (1 to 5 decimal characters).

Note –

- The LISTONLY, BLOCKid, and FILEnum parameters are mutually exclusive. In addition, if you specify the ALLVTVs parameter, or if a list or range of VTVs is specified, the FILEnum and BLOCKid parameters are ignored.
- Not specifying a BLOCKid value can cause very long execution times with high capacity media. The BLOCKid is used for high speed positioning to a VTV on the MVC.

DUMP

produce a S0C3 abend dump if RTV cannot decompress a VTV. If you specify DUMP, create a SYSMDUMP DD JCL statement to capture the dump.

OUTUNIT

the name to use to allocate the output tape unit. You can specify an MVS unit address, an esoteric name, or a generic name. The valid values are the same as for the UNIT= JCL parameter. This parameter is required if you do not specify LISTONLY.

name

the unit name.

VALIDATE

Validates that the RTV utility can successfully process the specified VTV(s) without creating an output tape.

Interfaces

SWSRTV only.

JCL Requirements

The following are required or optional statements for the RTV utility JCL:

STEPLIB

specifies the link library (SLSLINK) that contains the RTV modules.

SLSPRINT

specifies the destination of the RTV utility report.

SLSIN

specifies the input to the SWSRTV program (RTV utility name and parameters).

SYSMDUMP

optional DD to capture dump.

Note – Because the RTV utility must be capable of rewriting the tape standard labels on the output unit and positioning over label information on the input unit, Dynamic Allocation is used to invoke bypass label processing (BLP) on the tape volumes. This requires that the library that contains the SWSRTV executable code be APF authorized.

RTV Utility Report Messages

The RTV report displays the following messages:

Block number too large in compressed data

Explanation. An error was found in a compressed data record while processing a VTV. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Chunked record logic error

Explanation. An error was found while processing a chunked data record for a VTV. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Decompress invalid length parameter

Explanation. This indicates a program logic error. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Decompress invalid parameter list

Explanation. This indicates a program logic error. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Decompress logic error

Explanation. This indicates a program logic error. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Decompress pointer to work area is zero

Explanation. This indicates a program logic error. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Dynamic allocation error. Reason Code = xxxx-xxxx

Explanation. An error was encountered while attempting to dynamically allocate the INUNIT or OUTUNIT device. Refer to the IBM manual *MVS Authorized Assembler Services Guide* for a description of the dynamic allocation reason codes.

FILEnum of zero is invalid

Explanation. A FILEnum() value of 0 is invalid. The utility terminates with return code 12.

I/O error on input MVC

Explanation. An I/O error was encountered while reading a MVC. Further processing is stopped. The utility terminates with return code 12.

I/O error on output volume

Explanation. An I/O error was encountered while writing the output VTV. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Invalid compressed data block read

Explanation. This indicates that an invalid data record was found while processing this VTV. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Invalid VTV page number encountered

Explanation. A record sequence error was found in a compressed data record while processing a VTV. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Invalid VTV record encountered

Explanation. An error was found in a compressed data record while processing a VTV. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

MVC volser # does not match requested volser #

Explanation. The volume mounted as the input MVC did not match that requested by the MVCid() parameter. The utility terminates with return code 12.

MVC record length error

Explanation. A length error was found in a compressed data record while processing a VTV. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Incorrect algo byte

Explanation. An error was found in a compressed data record while processing a VTV. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

No HDR1 record found for requested VTV

Explanation. Following positioning by a BLOCKid() statement, there was no HDR1 record located at the desired position. Remove the BLOCKid statement and rerun the utility. The utility terminates with return code 12.

No HDR1 record found on input MVC

Explanation. The volume mounted as a MVC contained no HDR1 record. The utility terminates with return code 12.

No UHL1 record found on input MVC

Explanation. The volume mounted as a MVC contained no UHL1 record. The utility terminates with return code 12.

No VOL1 record found on input MVC

Explanation. The volume mounted as a MVC contained no VOL1 record. The utility terminates with return code 12.

NULL input buffer pointer

Explanation. This indicates a program logic error. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

NULL output buffer pointer

Explanation. This indicates a program logic error. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Requested VTV not found on MVC

Explanation. The volser requested by the VTVid() parameter was not found on the MVC. The utility terminates with return code 12.

Spanned length final error

Explanation. An error was found while processing a spanned data record for a VTV. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Spanned length intermediate error

Explanation. An error was found while processing a spanned data record for a VTV. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Unexpected request on input I/O

Explanation. This indicates a program logic error. Further processing is stopped. The utility terminates with return code 12.

Unexpected end of tape on output volume

Explanation. While writing the output VTV, an end of tape indication was encountered. The VTV must be completely contained on a single output volume. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Unexpected request on output I/O

Explanation. This indicates a program logic error. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

Unexpected tape mark on input MVC

Explanation. An unexpected tape mark was found on a MVC. Further processing is stopped. The utility terminates with return code 12.

VTVid range parameter is invalid

An invalid range value was found in the VTVid() specification. The utility terminates with return code 12.

VTV logical data check encountered

Explanation. A data check indicator was found in a compressed data record while processing a VTV. Further processing of this VTV is stopped. The VTV will be marked in the RTV Decompress Report output. The utility will continue to process other VTVs as necessary.

VTV volser # does not match requested volser #

Explanation. Following positioning by a BLOCKid() or FILEnum() statement, the VTV volser did not match that requested by the VTVid() parameter. The utility terminates with return code 12. Remove the BLOCKid or FILEnum() statement and rerun the utility.

RTV LISTONLY Listing

FIGURE 1-68 shows an example of the listing that RTV produces when you specify the LISTONLY parameter.

SWSRTV (6.2.0) StorageTek VTCS RTV Utility

PAGE 0001

TIME 14:23:33 Control Card Image Listing

DATE 2011-1-11

RTV MVC(C83107) INUNIT(/2A07) LISTONLY

SWSRTV (6.2.0) StorageTek VTCS RTV Utility

PAGE 0002

TIME 14:23:33 MVC C83107 Contents Report

DATE 12/01/00

VTV	File	Block	<---Created--->		<---Migrated--->		Data Set	VTVPAGE	VTV
Volser	#	ID	Date	Time	Date	Time	Name	Size	Status
VV6825	1	00000000	20110Nov30	12:07:56	20110Nov30	12:15:56	1933.MJS.VTV02.SL	Large	
VV6863	2	92005F0F	20110Sep27	12:57:54	20110Sep27	12:59:54		Large	
VV6893	3	92005F18	20110Aug18	08:57:26	20110Aug18	08:59:26		Large	
VV0403	4	92005F21	20110Aug18	08:57:26	20110Aug18	08:59:26	1933.MJS.VTV01.SL	Large	

FIGURE 1-68 Example RTV LISTONLY listing

This report lists the VTV's:

- Volser
- Logical file number on the MVC
- Block ID on the MVC
- Time created and migrated
- Data set name
- VTVPAGE size - STANDARD or LARGE
- Status - Not Current, or if blank, the VTV is current

RTV Decompress Listing

FIGURE 1-69 shows an example of the listing that RTV produces when you do not specify the LISTONLY parameter (that is, you run RTV to convert VTVs to Nearline volumes).

SWSRTV (6.2.0) StorageTek VTCS RTV Utility

PAGE 0001

TIME 14:28:33 Control Card Image Listing DATE 2011-1-18

RTV MVC(C8228) VTV(VV6800-VV6900) INUNIT(/2A07) OUTUNIT(/A820)

SWSRTV (6.2.0) StorageTek VTCS RTV Utility

PAGE 0002

TIME 14:28:33 MVC C83223 Contents Report DATE 2006-1-18

VTV	File	Block	<---Created--->		<---Migrated--->		Data Set	VTVPAGE	VTV
Volser	#	ID	Date	Time	Date	Time	Name	Size	Status
VV6070	1	00000000	20110Nov30	12:07:56	20110Nov30	12:15:56		Large	
VV0874	2	2B001384	20110Sep27	12:57:54	20110Sep27	12:59:54	1933.MJS.VTV02.SL	Large	
VV0772	3	A3002707	20110Aug18	08:57:26	20110Aug18	08:59:26		Large	
VV6828	4	9B002AB9	20110Aug18	08:57:26	20110Aug18	08:59:26		Large	Not current
VV6828	5	9B002AC2	20110Aug18	08:57:26	20110Aug18	08:59:26	1933.MJS.VTV021SL	Large	
VV6826	6	9B002ACB	20110Aug18	08:57:26	20110Aug18	08:59:26	1933.MJS.VTV01.AL	Large	

SWSRTV (1.0.0) StorageTek VTCS RTV Utility

PAGE 0003

TIME 14:28:33 MVC C83223 Decompress Report DATE 2006-1-18

VTV	VTV	Mounted	Mounted	Final	Final	VTVPAGE	Decompress
Volser	Label	Volser	Label	Volser	Label	Size	Status
VV6826	AL	XX0772	AL	VV6826	AL		Successful
VV6828	SL	XX0773	SL	VV6828	SL		Successful

FIGURE 1-69 Example RTV Decompress Listing

In addition to the contents fields shown in FIGURE 1-68 on page 169, the decompress listing shown inFIGURE 1-69 on page 170 lists the VTV's:

- Volser
- VTV Label type:
  - SL – Standard label
  - NL – Non-label
  - AL – ANSI label
- Mounted Volser (see note)
- Mounted Label type:
  - SL – Standard label
  - NL – Non-label
  - AL – ANSI label
- Final Volser (see note)
- Final Label type:
  - SL – Standard label
  - NL – Non-label
  - AL – ANSI label

Note –

- If the Mounted VTV contains standard or ANSI labels and CPYVOLID is not specified, RTV processes these label types as described in TABLE 1-6 on page 162.
- If the Mounted tape is non-labeled or has a non-standard tape label, CPYVOLID will be automatically specified for this VTV decompression and the labels from the input VTV (if they exist) will be copied to the output tape.
- If the Mounted or Final volume is non-labeled, the volser printed will be '**NL**'

SET MIGOPT

SET MIGOPT changes the following migration parameters:

- Maximum and minimum concurrent automatic migration, immediate migration, and migrate-to-threshold tasks
- High and low AMTs

Syntax

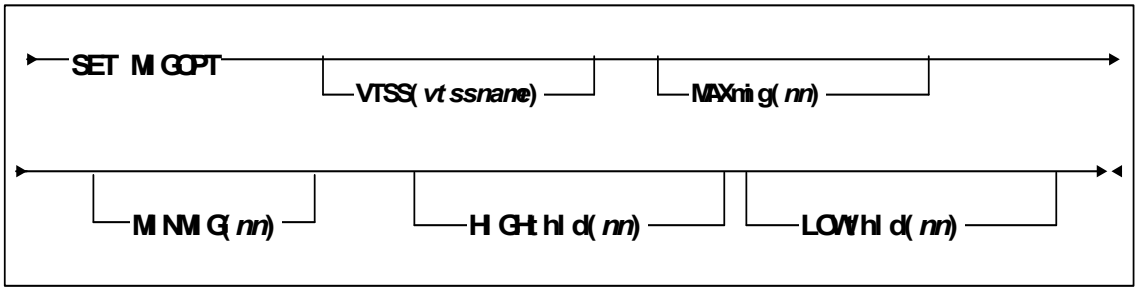


FIGURE 1-70 SET MIGOPT syntax

Parameters

VTSS

the VTSS whose migration parameters you want to change. If you do not specify a VTSS, the changes affect all VTSSs.

vtssname

the VTSS name.

MAXMIG(*nn)*

specifies the maximum number of concurrent automatic migration, immediate migration, and migrate-to-threshold tasks.

Valid values are 1 to the number of RTDs on the VTSS. See also "MAXRTDS" on page 25. There is no default; if you do not specify a value, the current value is unchanged.

MINMIG(*nn)*

specifies the minimum number of concurrent automatic migration, immediate migration, and migrate-to-threshold tasks.

Valid values are 1 to the MAXMIG setting. There is no default; if you do not specify a value, the current value is unchanged.

HIGHthld

specifies the new high AMT.

high-thr

the new high AMT as a percent of VTSS space. Valid values are 5 to 95 and must be greater than the LOWthld value.

LOWthld

specifies the new low AMT.

low-thr

the new low AMT as a percent of VTSS space. Valid values are 5 to 95 and must be less than the HIGHthld value.

Interfaces

SWSADMIN utility and VT command.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the report.

SLSIN

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

SET RMM

SET RMM enables or disables the VTCS to DFSMSrmm interface.

Syntax

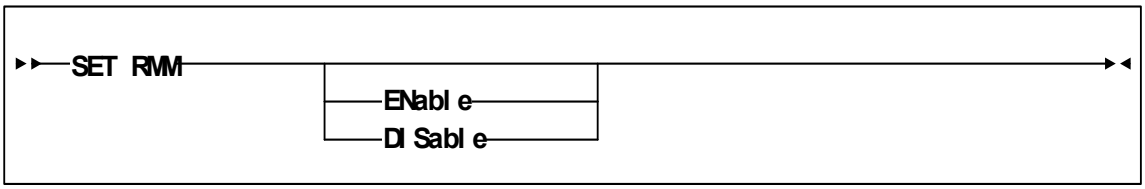


FIGURE 1-71 SET RMM syntax

Parameters

ENAble

enables the VTCS to DFSMSrmm interface. With the interface enabled, VTCS communicates with DFSMSrmm on all scratch mounts and scratch requests of VTVs whose Management Class specifies DELSCR(YES).

DISable

disables the VTCS to DFSMSrmm interface. With PTF L1H139T installed, the interface is automatically disabled at VTCS initialization.

If you do not specify a parameter, VTCS displays the current setting of the interface.

Interfaces

SWSADMIN utility and VT command.

Usage

Enabling the DFSMSrmm interface is required **only** for sites that do not have IBM APAR OA03368 applied and the following (minimum) VTSS microcode levels installed:

- VSM3: N01.00.65
- VSM4: D01.00.03
- VSM5: all microcode levels have the required change

If required, StorageTek recommends adding the VT SET RMM ENA command to the SLSSYS*xx* PARMLIB member read at HSC initialization.

Note – Enabling this interface can cause delays in scratch mount processing.

Command Example

To enable the VTCS to DFSMSrmm interface, enter:

```
.VT SET RMM EN
```

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the utility report.

SLSIN

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

JCL Example

Figure 1-72 shows example JCL enable the VTCS to DFSMSrmm interface.

```
//SETRMM  EXEC PGM=SWSADMIN,PARM='MIXED'
//STEPLIBDD DSN=hlq.SLSLINK,DISP=SHR
//SLSPRINTDD SYSOUT=*
//SLSINDD *
  SET RMM ENA
```

FIGURE 1-72 SET RMM JCL example to enable the VTCS to DFSMSrmm interface

TRACE

TRace starts or stops event tracing for specified VTCS components.

Syntax

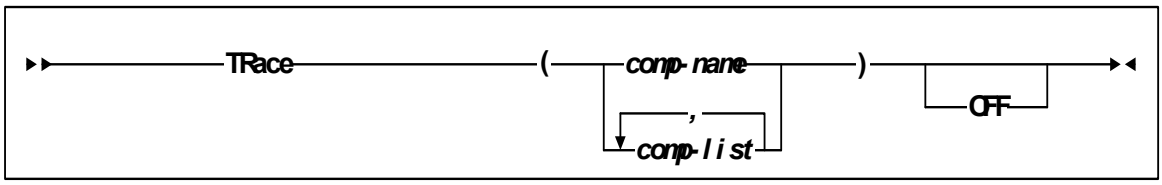


FIGURE 1-73 TRace syntax

Parameters

OFF

stops tracing for the specified components.

comp-name

specifies one of the following components:

VTCS

traces the VTCS component.

comp-list

specifies a list of components separated by commas or blanks.

Use TRace to start or stop event tracing for specified VTCS components.

Interfaces

SWSADMIN utility and VT command.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the report.

SLSIN

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

VARY CLINK

Vary CLInk changes CLINK states.

Syntax

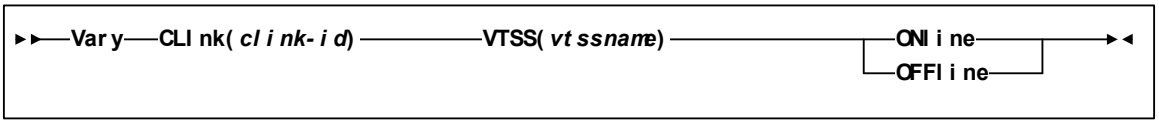


FIGURE 1-74 Vary CLINK syntax

Parameters

CLInk

the specified CLINK.

clink–id

the link ID.

VTSS

the sending VTSS in the Cluster.

vtssname

the 1 to 8 character identifier of the sending VTSS.

ONline

vary the specified CLINK online.

OFFline

vary the specified CLINK offline.

Interfaces

SWSADMIN utility and VT command.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the report.

SLSIN

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

VARY RTD

Vary RTD changes RTD states.

Syntax

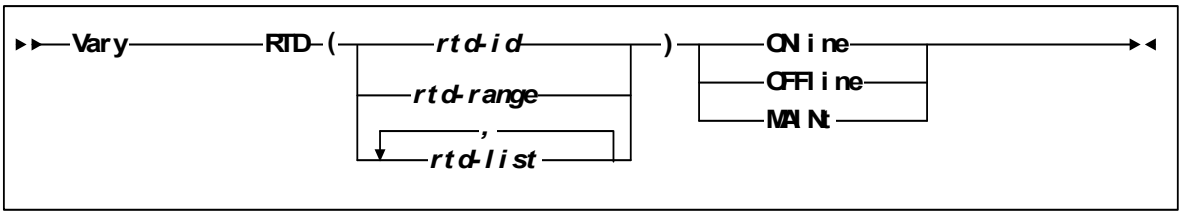


FIGURE 1-75 Vary RTD syntax

Parameters

RTD

change the state of the specified RTDs.

rtd–id, rtd-range, or *rtd-list*

the unit addresses of one or more RTDs. Lists and ranges of RTDs are limited to 64 items for VSM2s and VSM3s and 256 items for VSM4s.

ONline

vary the specified RTDs online to their connected VTSSs.

OFFline

vary the specified RTDs offline to their connected VTSSs.

MAINt

vary the specified RTDs offline (maintenance mode) to their connected VTSSs.

Interfaces

SWSADMIN utility and VT command.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the report.

SLSIN

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

VARY VTSS

Vary VTSS changes VTSS states on all hosts.

Syntax

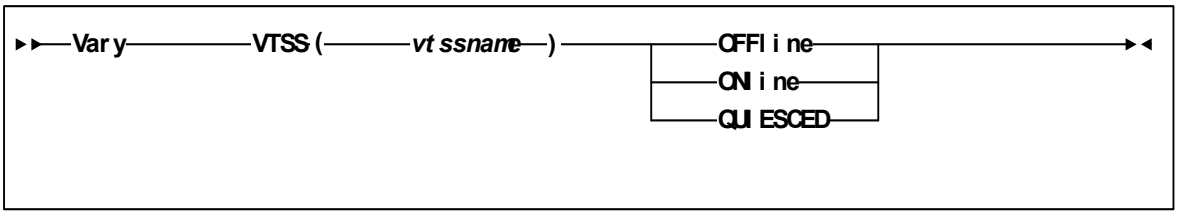


FIGURE 1-76 Vary VTSS syntax

Parameters

VTSS

change the state of the specified VTSS.

vtssname

the VTSS name.

ONline

vary the specified VTSS online.

OFFline

vary the specified VTSS offline.

QUIESCED

vary the specified VTSS to quiesced state.

Note – Vary VTSS **does not** change the state of the VTDs or RTDs associated with the specified VTSS!

Interfaces

SWSADMIN utility and VT command.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the report.

SLSIN

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

VTVMAINT

VTVMAINT does the following:

- Unlinks VTVs from MVCs,
- Sets the VTV Management Class, and
- Logically dismounts specified VTVs in an offline VTSS.

Syntax

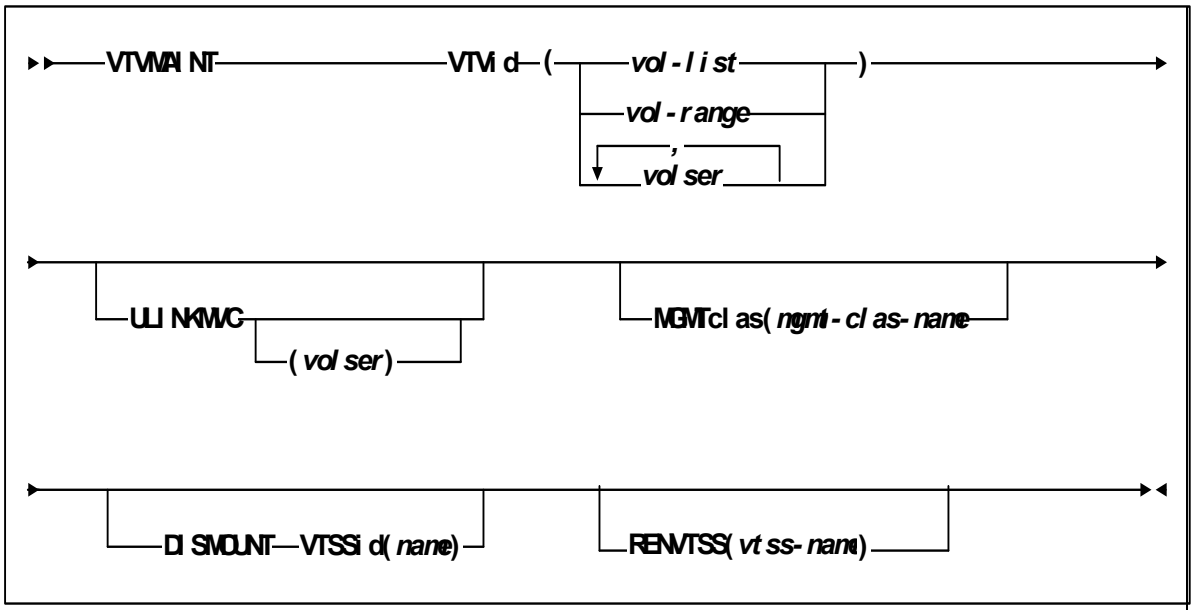


FIGURE 1-77 VTVMAINT syntax

Parameters

VTVid

specifies the VTVs.

volser, vol-range or *vol-list*

the volsers of one or more VTVs.

ULINKMVC

unlink the specified VTVs.

volser

unlink the specified VTVs from the specified MVC.

MGMTclas

set the Management Class of the VTVs.

mgmt-class-name

the Management Class name that you specified on the MGMTclas control statement. For more information, see "MGMTCLAS Control Statement" on page 234.

DISMOUNT VTSSid

logically dismount the specified VTVs in the specified VTSS.

name

the VTSS name.

RENVTSS

Sets the name of the VTSS where the VTV was last mounted to *name*.

name

the VTSS name.

Note – To set the VTSS name:

- The host on which the utility is run must have access to the VTSS, and
- The new VTSS name must be the name of an existing VTSS in the VTCS `CONFIG` deck.

Interfaces

SWSADMIN utility only.

If migrated copies of the dismounted VTVs exist that an online VTSS can access, you can now use this VTSS to access the VTVs.

Caution – If the VTV copy mounted in the offline VTSS was modified and not migrated, the MVC copy that you recall to an alternate VTSS is not current! Therefore, StorageTek strongly recommends that you do not recall these non-current MVC copies!

Tip – When the offline VTSS is ready to be brought back online, StorageTek strongly recommends that you audit the VTSS before running production jobs that use the VTSS. Also ensure that you clear any boxed VTD conditions before issuing the VTSS VARY ONLINE command.

Return Codes

TABLE 1-7 VTVMAINT Return Codes

Return Code	Meaning
0	All requested updates for the VTV completed successfully.
4	One requested update for the VTV failed and at lest one other requested update completed successfully.
8	All requested updates for the VTV failed.

Note – The **final** return code for the VTVMAINT job is the **largest** return code generated by any single VTVs being updated. For example, if 5 VTVs generate a return code 0 and one VTV generates a return code 8, the final return code is 8.

JCL Requirements

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the report.

SLSIN

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

VTVMAINT Report

FIGURE 1-78 shows an example of a VTVMAINT report for the following command:

VTVMAINT VTV(X00000-X00002) ULINKMVC MGMTCLAS(M1)

SWSADMIN (6.2.0) STORAGETEK VTCS SYTEM UTILITY PAGE 0001

TIME 06:32:03 VTV MAINTENANCE DATE 2006-04-19

VTV	RC
X00000	04
X00001	04
X00002	04

VTVMAINT EXCEPTION REPORT

VTV X00000 IS ALREADY IN MGMTCLAS M1

VTV X00001 IS ALREADY IN MGMTCLAS M1

VTV X00002 IS ALREADY IN MGMTCLAS M1

SLS1315I SWS500.V5.CDS WAS SELECTED AS THE PRIMARY CONTROL DATA SET

SWSADMIN (6.2.0) STORAGETEK VTCS SYTEM UTILITY PAGE 0002

TIME 06:32:03 VTCS VTV REPORT DATE 2006-04-19

VTV VOLSER	SIZE (MB)	COMP%	<----CREATION-----> DATE	TIME	<----LAST USED----> DATE	TIME	MIGR	SCRT	RESD	REPL	MGMT CLASS	MVC1	MVC2	MVC3	MVC4	MAX VTV	VTSS NAME
X00000	0.01	0	2006MAY19	05:02:08	2006MAY19	05:22:08	-	-	R	-	M1	022550	022551	022552	022553	800	
X00001	0.01	0	2006MAY19	05:02:08	2006MAY19	05:22:08	-	-	R	-	M1	033550	033551	033552	033553	800	
X00002	0.01	0	2006MAY19	05:02:08	2006MAY19	05:22:08	-	-	R	-	M1	044550	044551	044552	044553	800	

3 INITIALIZED VTVS PROCESSED

0 NON-INITIALIZED VTVS PROCESSED

FIGURE 1-78 VTVMAINT Report

As shown in FIGURE 1-78, the VTVMAINT report shows:

- Status of VTVs processed - volser and return code (0 - all updates completed, 4 - some updates completed, 8 - no updates completed).
- An exception report of the reason for all uncompleted updates.
- A VTV report.

VTVRPT

The VTVRPT reports the status of your VSM system's VTVs.

Syntax

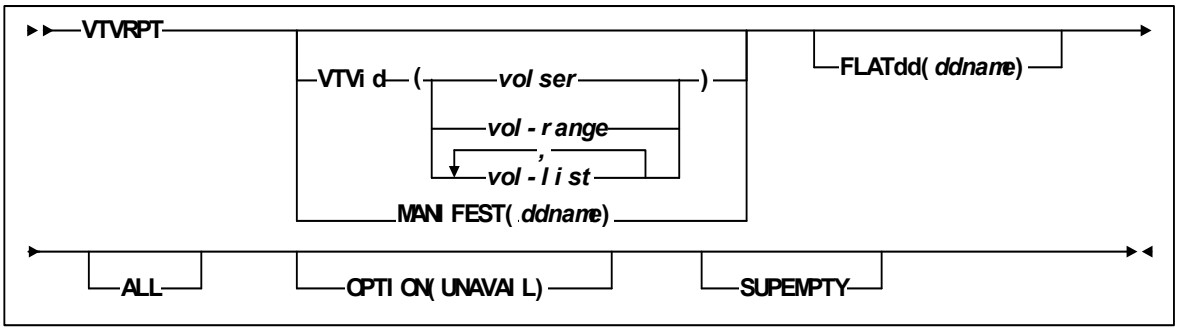


FIGURE 1-79 VTVRPT syntax

Parameters

VTVid

specifies the VTVs for the report. If you do not specify the VTVs, the report includes all initialized VTVs in your VSM system. A VTV is initialized when VTCS has used it at least once.

volser, vol-range, or *vol-list*

the volsers of one or more VTVs.

MANIFEST

specifies the input ddname of the manifest file used to generate the report.

ddname

ddname of the manifest file. **Note that** you can specify a merged manifest file or multiple manifest files.

FLATdd

specifies the output destination ddname if a flat file is required (must be RECFM= VB,LRECL=100).

ddname

the ddname of the flat file included in the JCL.

ALL

specifies to report on all VTVs (including non-initialized volumes). If you do not specify ALL, only initialized VTVs are reported.

OPTION(UNAVAIL)

specifies to report only on unavailable VTVs (VTVs in an offline VTSS).

SUPEMPTY

suppresses the printing of empty VTVs. That is, the VTV report does not contain details of VTVs that have no data on them or VTVs that are fenced. SUPEMPTY is mutually exclusive with ALL and OPTION(UNAVAIL).

Interfaces

SWSADMIN utility only.

JCL Requirements

The following are the required and optional statements for the VTVRPT JCL:

STEPLIB

specifies the link library (SLSLINK) that contains the VTCS and HSC modules.

SLSPRINT

specifies the destination for the VTV report.

SLSIN

specifies the input to the SWSADMIN program (VTVRPT utility name and parameters).

VTV Report

FIGURE 1-80 shows an example of a VTV report and Figure 78. on page 150 shows an example of a VTV report with the UNAVAIL option. This report lists only unavailable VTVs in three sections--unavailable mounted on a VTD, unavailable VTSS-resident, and unavailable VTSS-resident and fenced.

SWSADMIN (6.2.0) STORAGETEK VTCS SYTEM UTILITY PAGE 0002

TIME 06:32:03 VTCS VTV REPORT DATE 2006-03-20

VTV VOLSER	SIZE (MB)	COMP %	<----CREATION-----> DATE	TIME	<----LAST USED----> DATE	TIME	MIGR	SCRT	RESD	REPL	MGMT CLASS	MVC1	MVC2	MVC3	MVC4	MAX VTV	VTSS NAME
X00T00	0.04	84	2006JUL16	05:02:08	2006JUL19	05:41:00	M	-	R	-	VCL4	022550	022551	022552	022553	.8	VTSS16
X00002	<MOUNT>		2006JUL14	06:54:35	2006JUL19	07:43:46	M	-	R	-	VCL4	033550	033551	033552	033553	.8	VTSS17
X00003	15.60	84	2006JUL14	10:05:05	2006JUL19	05:41:28	M	-	R	-	VCL2	044550	044551			.4	VTSS16
X00004	0.36	84	2006MAY28	08:51:20	2006JUL19	05:41:30	M	S	R	-	VCL3	022550	022551	022552		.8	VTSS16
X00005	15.60	84	2006JUL14	10:05:14	2006JUL19	05:41:31	M	-	R	-	VCL1	033550				.4	VTSS16
X00006	15.60	84	2006JUL14	10:08:23	2006JUL19	08:45:31	C	-	-	-	VCL4	044550	044551	044552	044553	.8	VTSS17

FIGURE 1-80 Example output from VTVRPT

SWSADMIN (6.2.0) STORAGETEK VTCS SYTEM UTILITY PAGE 002

TIME 06:59:03 UNAVAIL MOUNTED VTV REPORT DATE 2006-03-20

VTV VOLSER	SIZE (MB)	COMP %	<----CREATION-----> DATE	TIME	<----LAST USED----> DATE	TIME	MIGR	SCRT	RESD	REPL	MGMT CLASS	MVC1	MVC2	MVC3	MVC4	MAX VTV	VTSS NAME
Y09053	<MOUNT>		2006MAR19	09:34:14	2006MAR20	05:55:44	-	-	R	-	M9					.8	HBVTSS16

SWSADMIN (6.2.0) STORAGETEK VTCS SYTEM UTILITY PAGE 003

TIME 06:59:03 UNAVAIL RESIDENT VTV REPORT DATE 2006-03-20

VTV VOLSER	SIZE (MB)	COMP %	<----CREATION-----> DATE	TIME	<----LAST USED----> DATE	TIME	MIGR	SCRT	RESD	REPL	MGMT CLASS	MVC1	MVC2	MVC3	MVC4	MAX VTV	VTSS NAME
X01007	156.24	89	2006JAN10	03:00:02	2006MAR01	04:51:47	-	S	R	-							HBVTSS16
X01010	3.90	0	2006MAR01	09:10:37	2006MAR01	09:10:37	-	-	R	-							HBVTSS16
X01014	3.90	0	2006MAR01	09:11:08	2006MAR01	09:11:08	-	-	R	-							HBVTSS16
X01021	3.90	0	2006MAR01	09:21:11	2006MAR01	09:21:11	-	-	R	-							HBVTSS16

SWSADMIN (6.2.0) STORAGETEK VTCS SYTEM UTILITY PAGE 004

TIME 06:59:03 UNAVAIL FENCED VTV REPORT DATE 2006-03-20

VTV VOLSER	SIZE (MB)	COMP %	<----CREATION-----> DATE	TIME	<----LAST USED----> DATE	TIME	MIGR	SCRT	RESD	REPL	MGMT CLASS	MVC1	MVC2	MVC3	MVC4	MAX VTV	VTSSNAME
X01280	<FENCED>						-	-	-	-							
X04762	<FENCED>						-	-	-	-							
X04776	<FENCED>						-	-	-	-							
X02019	<FENCED>						-	-	-	-							
X10066	<FENCED>						-	-	-	-							
X10068	<FENCED>						-	-	-	-							

FIGURE 1-81 Example output from VTVRPT (UNAVAIL option)

VTVRPT Report Fields

The following list describes the VTV report fields.

VTV Volser

the VTV volser.

Size (MB)

the uncompressed size of the VTV (MB). **<MOUNT>** indicates that the VTV was mounted when the report ran. **<FENCED>** indicates that the VTV's state is unknown. If **<FENCED>** appears, contact StorageTek software support.

Comp %

the VTV compression percentage achieved. This is the difference between the uncompressed and compressed VTV size expressed as a percentage of the uncompressed VTV size. For example if a 100MB VTV compresses to 40MB then the compression% will be given as 60%. A compression of 0% indicates that no compression was possible on the VTV.

Creation Date and Time

the date and time that the VTV was created.

Last Used Date and Time

the date and time that the VTV was last used. This date and time value is updated by successful completion of a VTV mount, migrate, recall, or scratch.

Migr

indicates whether the VTV has been migrated (M) or consolidated (C). If the VTV is both migrated and consolidated, a 'C' appears in this field. If all copies of the VTV are archived, an 'M' appears in this field. If the VTV has not been migrated, it is either VTSS resident or non-existent (not created or used, scratched, and deleted).

Scrt

indicates whether the VTV has been scratched.

Resd

indicates whether the VTV is resident in a VTSS.

Repl

one of the following VTV replication statuses:

-

the VTV has no replication requirements or the VTV has replication requirements but is not resident.

R

replication is required but has not started.

S

replication has started.

C

replication has completed.

Note – The **Repl** column displays R, S, or C only if the VTV is resident, otherwise a '-' appears.

MGMT Class

the name of the Management Class for the VTV specified.

MVC1, MVC2, MVC3, MVC4

the MVC(s) that contain the VTV (for both migration and consolidation). If all of these fields are empty, the VTV has not been migrated or consolidated. If 1 or more of these fields list an MVC volser, the VTV was migrated to each of these MVCs.

MAX VTV

the maximum size of the VTV in Gb (.4, .8, 2, or 4).

VTSSNAME

the VTSS where the VTV resides, or, if the VTV is migrated, the VTSS where the VTSS was last resident. If this field is empty, the VTV is non-existent (not created or used, scratched, and deleted).

VTV Flat File Record Format

TABLE 1-8 shows the record format of the flat file produced by VTVRPT.

TABLE 1-8 VTVRPT flat file record format

Decimal Offset	Hexadecimal Offset	Type	Length	Description
0	0	start of record		start of VTV flat file record
0	0	integer	4	record length
4	4	character	1	character set type of text fields
		X'61'		ASCII
		X'6E'		EBCDIC
5	5	character	1	record type 'V' (indicates VTV report)
6	6	character	6	VTV volser
12	C	character	8	VTSS where the VTV resides
20	14	integer	4	uncompressed VTV size (MB)
24	18	character	1	VTV migrated? (Y, N, or C)
25	19		1	reserved
26	1A	character	6	MVC volser (first copy)
32	20	character	6	MVC volser (second copy)
38	26	character	1	always set to N
39	27	character	1	VTV scratched? (Y or N)
40	28	time_t	4	date VTV created (time_t format)
44	2C	time_t	4	date VTV last referenced (time_t format)
48	30		8	reserved
56	38	integer	2	compression percentage for VTV
58	3A	character	1	replicate indicator: C, R, S, or - For more information, see "VTV Report" on page 189.
59	3B	character	1	fenced indicator (Y or N)
60	3C	character	1	mounted indicator (Y or N)
61	3D	character	8	Management Class name
69	45	character	6	MVC volser (third copy)
75	4B	character	6	MVC volser (fourth copy)
81	51			reserved
82	52	integer	2	maximum VTV size (400 or 800)

CHAPTER 2

SMC Support for VSM

This chapter contains reference information about the SMC features that support VSM.

Tip – SMC 6.2 also provides the SMCUUI utility, which allows you to enter VTCS commands and utilities from the SMC console running on MVS hosts other than the MVS system that is running VTCS. See the *SMC Configuration and Administration Guide* for more information.

COMMtest

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.

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

Syntax

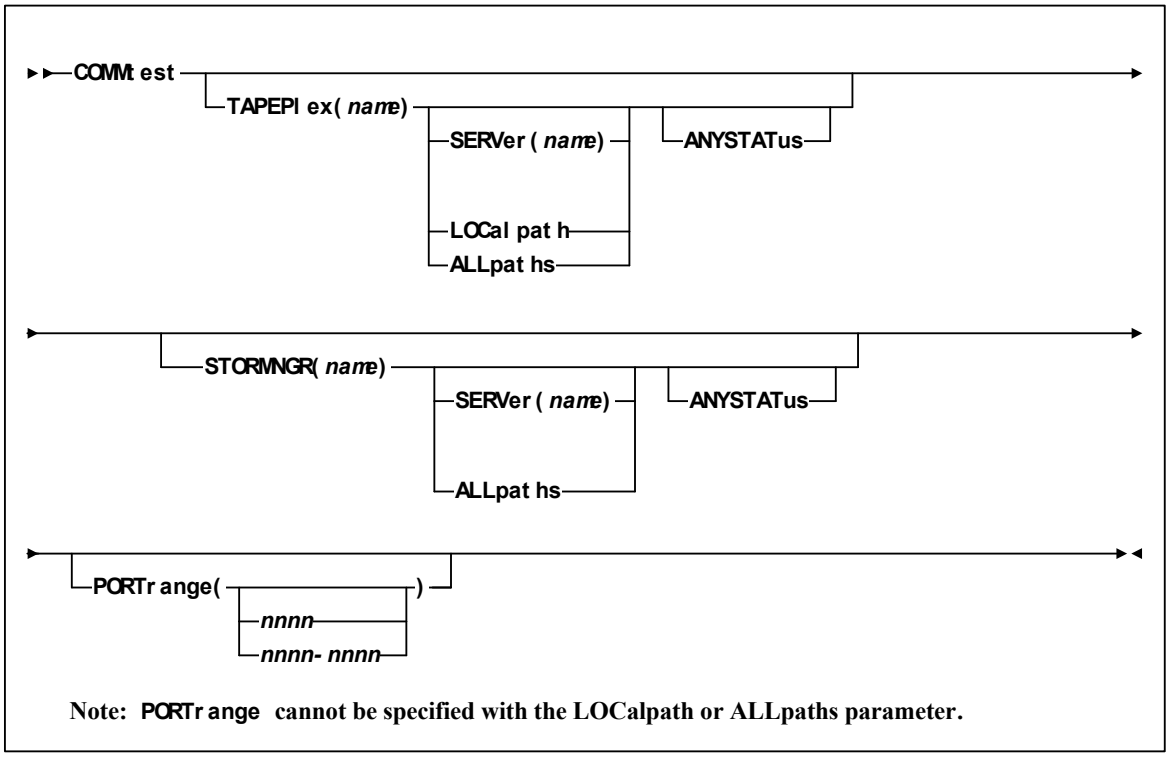


FIGURE 2-1 COMMtest syntax

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.

MOUNTDef Command DISMSCRVirt Parameter

As described in *SMC Configuration and Administrati*on Guide, the SMC MOUNTDef DISMSCRVirt parameter (which replaces the HSC SLILIBRARY DELDISP parameter) specifies how the SMC interprets the delete disposition on a dismount message for VTVs as follows:

OFF

Retain the VTV as nonscratch when MVS indicates delete disposition in the dismount message. This is the default setting.

ON

Scratch the virtual volume when MVS indicates delete disposition in the dismount message.

POLICY Command

The SMC POLicy command specifies a tape policy in a single named policy object. You can specify a policy either on a TAPEREQ statement or via the SMC DFSMS ACS routines.

Syntax

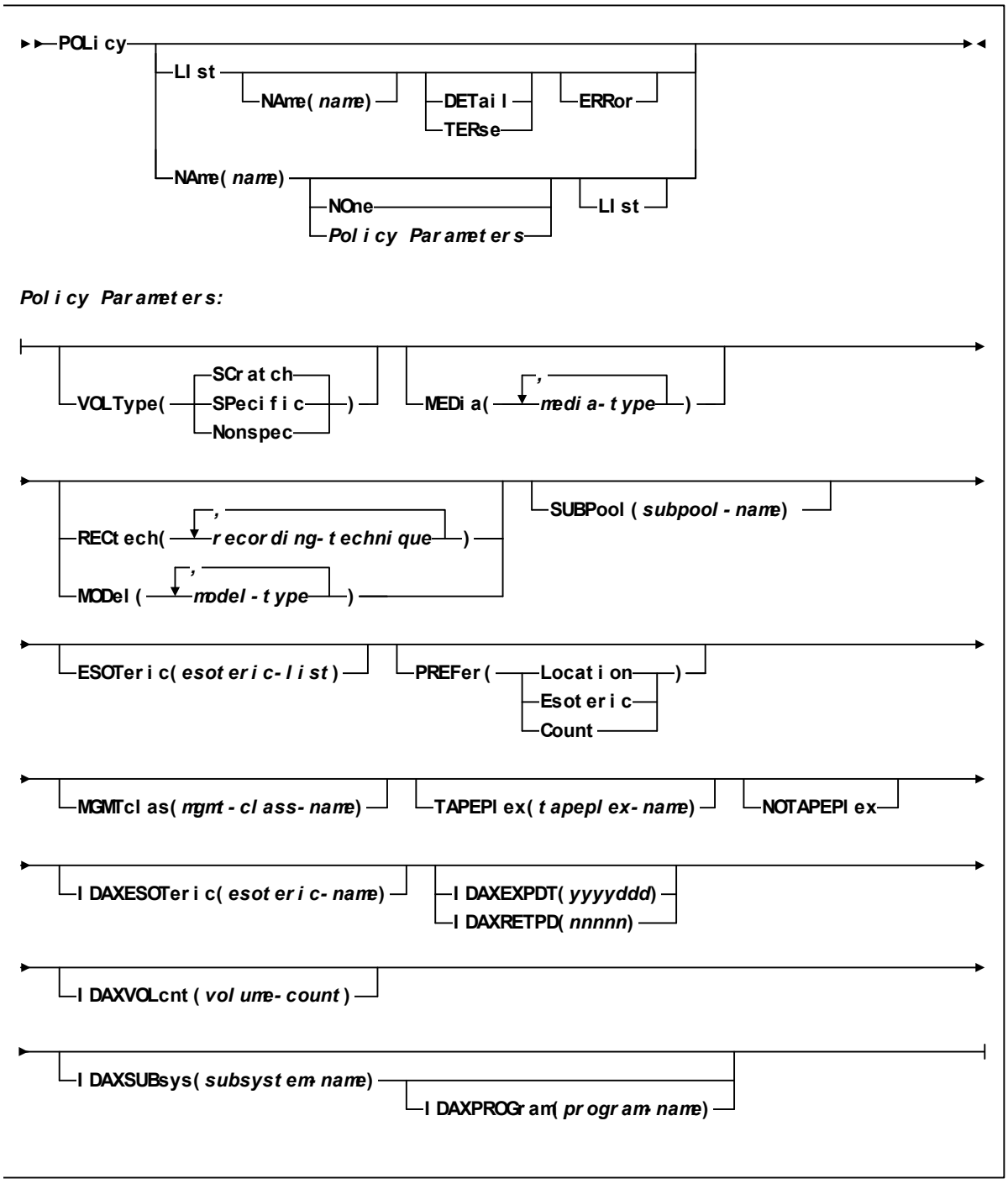


FIGURE 2-2 POLicy Syntax

Parameters

LIst

displays information about one or more policies.

NAme

specifies the name of the policy to be displayed. If the last character is an asterisk (*), all policies that begin with the prefix are displayed. For example, POL LI NA V* displays all policies that begin with V.

DETail

displays audit detail information about the policy, including last update date, time, and source.

TERse

lists only policy names and whether each policy applies to scratch, specific, or all allocations.

ERRor

displays each policy in error status.

Note –

- Matching policies are listed in alphabetic order, regardless of the order in which the they were defined to SMC. When specified with other parameters, the LIst is performed **after** the other parameters have been processed.
- DETail and TERse are mutually exclusive.

VERify

verifies that policies are logically consistent so that you do not have to wait until initialization for verification. Policies loaded at startup are automatically verified during subsystem initialization.

NAme

specifies the policy name (for display, creation, or both).

name

the policy name.

NOne

removes all policy parameters associated with the named policy (the policy still exists). Specifying NOne for a policy implies the default behavior (for example, any media/rectech, subpool 0, and so forth).

VOLType

specifies the allocation for the policy:

SCratch | Nonspecific

The policy applies only to scratch (nonspecific) allocations (the default).

SPecific

The policy applies only to specific allocations.

Note – VOLType(SPecific) is mutually exclusive with SUBPool and MEDia.

MEDia

specifies the volume media.

Virtual

specifies that VSM will route the data set to a VTV mounted on a VTD.

RECtech

specifies the recording technique.

Virtual

specifies that VSM will route the data set to a VTV mounted on a VTD.

MODel

specifies the transport model.

Virtual

specifies that VSM will route the data set to a VTV mounted on a VTD.

MGMTclas

specifies a Management Class you defined on the MGMTclas control statement.

mgmt-class-name

the Management Class name.

Note –

- The MEDia, RECtech, and MODel parameters have a value of Virtual for VSM only. Specifying Virtual on any of these three parameters will route the data set to a VTV mounted on a VTD. Virtual does not apply to NCS without VSM installed. The MGMTclas parameter does not apply to NCS without VSM installed. For more information about values for these parameters for Nearline volumes, see *SMC Configuration and Administration Guide*.
- If you do not specify MEDia, a default is chosen based on the value of the RECtech parameter.
- If both MEDia and RECtech are omitted, all available device types are eligible for allocation.
- If RECtech is not specified on any matching TAPEREQ, a default is chosen based on the value of the MEDia parameter.
- If you specify a Management Class on the MGMTclas parameter, you must specify both MEDia(V) and RECtech(V).
- VOLType(SPecific) and MEDia are mutually exclusive.
- RECtech and MODel are mutually exclusive.

SUBPool

specifies the scratch subpool that contains volumes used to satisfy nonspecific requests. For more information about scratch subpool management, see *SMC Configuration and Administration Guide*.

poolname

the subpool name.

Note – SUBPool and VOLType(SPecific) are mutually exclusive.

ESOTeric

specifies the esoteric that defines the list of eligible transports to be used to satisfy a tape request.

To route a data set to a VTD, specify one of the esoteric names that you defined during configuration. For more information on creating and using VSM esoterics, see *Installing ELS*.

For more information on using esoteric substitution to route a data set to Nearline transports that are *not* VTDs, see "User Exits" in *HSC System Programmer's Guide for MVS*.

Note – If the esoteric contains devices not currently in the Eligible Device List, SMC allocation cannot add devices to the list, but can exclude devices not in the specified esoteric.

---

esoteric-name

the esoteric name.

PREFer

optionally, specifies the relative value of the esoteric list and additional preferencing information. PREFer is **only** valid when a list of esoterics is specified.

Location

SMC prefers drives based on LSM location of a specific volume. This is the default for specific volumes. Location is **only** valid for VOLType (SP).

Esoteric

SMC prefers drives based on the order specified in the esoteric list. This is the default for scratch allocations.

Count

SMC prefers drives based on scratch counts in the LSM. Count is **only** valid for VOLType (N).

MGMTclas

specifies a Management Class you defined on the MGMTclas control statement.

mgmt-class-name

the Management Class name.

TAPEPlex

optionally, specifies the name of a TapePlex defined to SMC. SMC uses the specified TapePlex name to determine the "owner" of an allocation request. Only the specified TapePlex name is queried to determine specific volume information or scratch availability.

name

The TapePlex name.

Note –

- The named TapePlex **must** be defined using a TAPEPlex command before the POLicy command is processed.
- TAPEPlex and NOTAPEPlex are mutually exclusive.

NOTAPEPlex

optionally, specifies that SMC will not perform a TapePlex lookup to obtain volume information.

Note –

- NOTAPEPlex and TAPEPlex are mutually exclusive.
- VOLType (SPecific) is required.

IDAXESOTeric

optionally, specifies the “true” esoteric name to be substituted during IDAX (Interpreter/Dynamic Allocation Exit). Unlike the ESOTeric parameter, the IDAXESOTeric parameter changes the JCL esoteric to the value specified and does not use a subset of the original esoteric.

esoteric-name

a valid esoteric name to be used for the allocation

IDAXEXPDT

optionally, indicates the expiration date to be applied to the allocation during IDAX. This parameter is mutually exclusive with IDAXRETPD.

yyyyddd

the desired expiration date (4-digit year and Julian day).

IDAXRETPD

optionally, specifies the retention period to be applied to the allocation during IDAX. This parameter is mutually exclusive with IDAXEXPDT.

nnnnn

retention period in number of days

IDAXVOLcnt

optionally, specifies the maximum number of volumes to be applied to the allocation during IDAX. This value overrides the volume count parameter specified in the JCL.

volume-count

maximum volume count. Valid values are 1-255.

IDAXSUBsys

optionally, specifies that SMC direct any FDR/DFDSS job to the StorageTek™ ExHPDM backup and restore software. The IDAXSUBsys policy directs the output tape data set allocation to ExHPDM, effectively placing the DD SUBSYS JCL specification in the job.

IDAXPROGram

optionally, alters the program name in the job for DFDSS (ADRDSSU) to the ExHPDM User Interface Module (SOVDSSU), and in conjunction with the IDAXSUBsys parameter, directs the backup DFDSS job to ExHPDM.

Note – Refer to the *ExHPDM Operator and System Programmer’s Guide* for more information about the ExHPDM backup and restore software.

Route

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
- 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

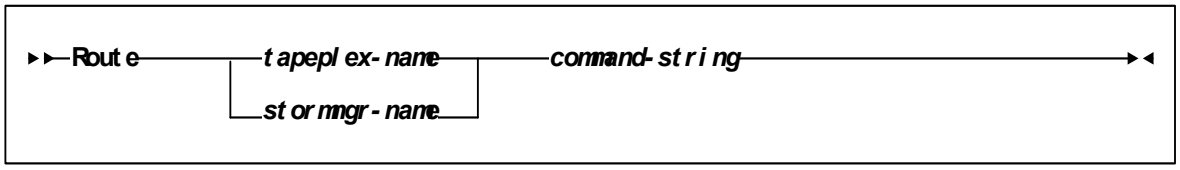


FIGURE 2-3 Route syntax

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 Command

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.
- Simlarly, 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" on page 211 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)
```

Syntax

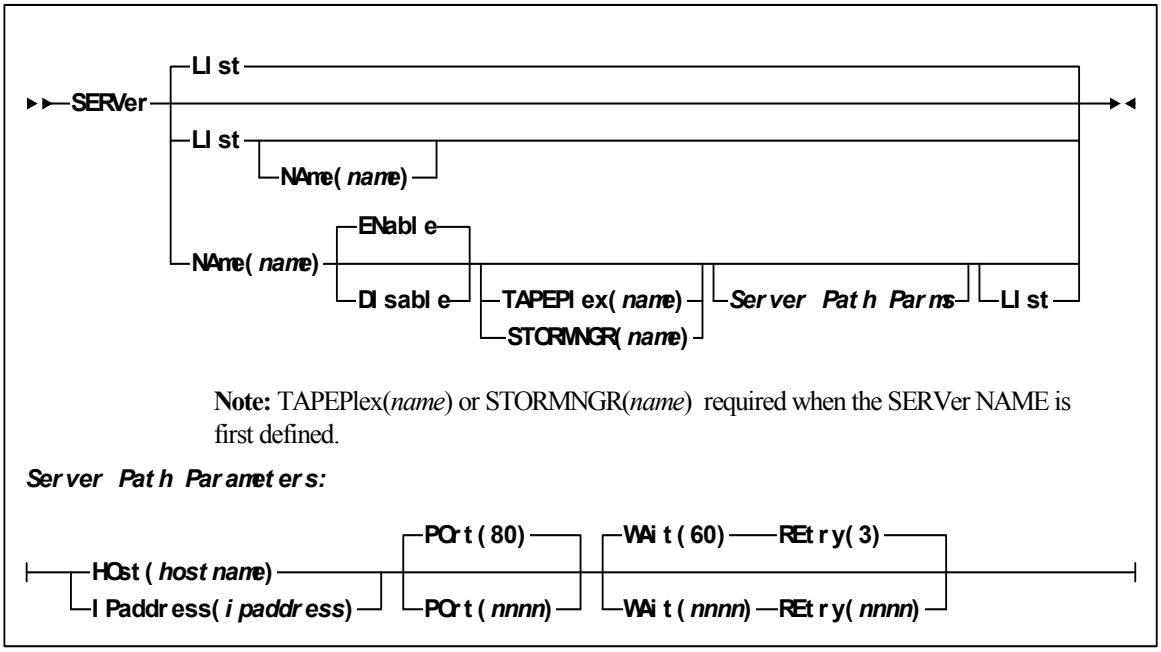


FIGURE 2-4 SERver Syntax

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 ELS appliance, 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.

STORMNGR

The SMC STORMNGR command defines a VLE appliance.

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

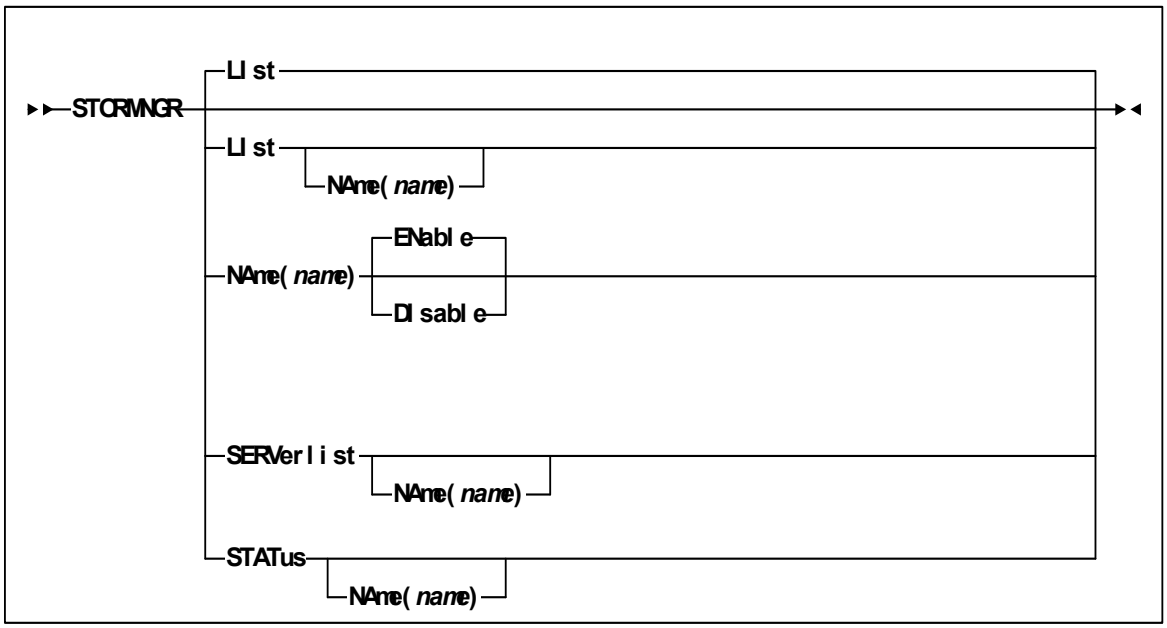


FIGURE 2-5 STORMNGR syntax

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 name, which is also defined to VTCS, and is initially defined by the ELS GUI. 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.

TAPEPLEX Command

The SMC TAPEPLEX command defines a TapePlex to a client host, where a “TapePlex” is all of the following defined a single CDS:

- ACSs, Nearline transports, and Nearline volumes.
- VTSSs, VTDs, and MVCs.

Note –

- The TAPEPLEX NAme parameter merely specifies the name of the TapePlex. One or more SERVer commands must point to the defined TapePlex and the HTTP server on the host using the TapePlex CDS, for example:

  TAPEPLEX NAME(REMHSC1)
  SERVER NAME(REMPATH) TAPEPLEX(REMHSC1) HOST(MVSB)

  For a client host to access remote HSC/VTCS libraries, therefore, you need to create a combination of TAPEPLEX and SERVer commands on the client. For more information, see “SERVER Command” on page 207.
- If you create one TAPEPLEX command, StorageTek recommends that you create sufficient TAPEPLEX commands to cover **all** the libraries in your system. Otherwise, SMC attempts to use active HSC and MVS/CSC subsystems defined in the client host’s SSCVT chain.
- You can also use the TAPEPLEX command to display information and status of one or all defined libraries.

Syntax

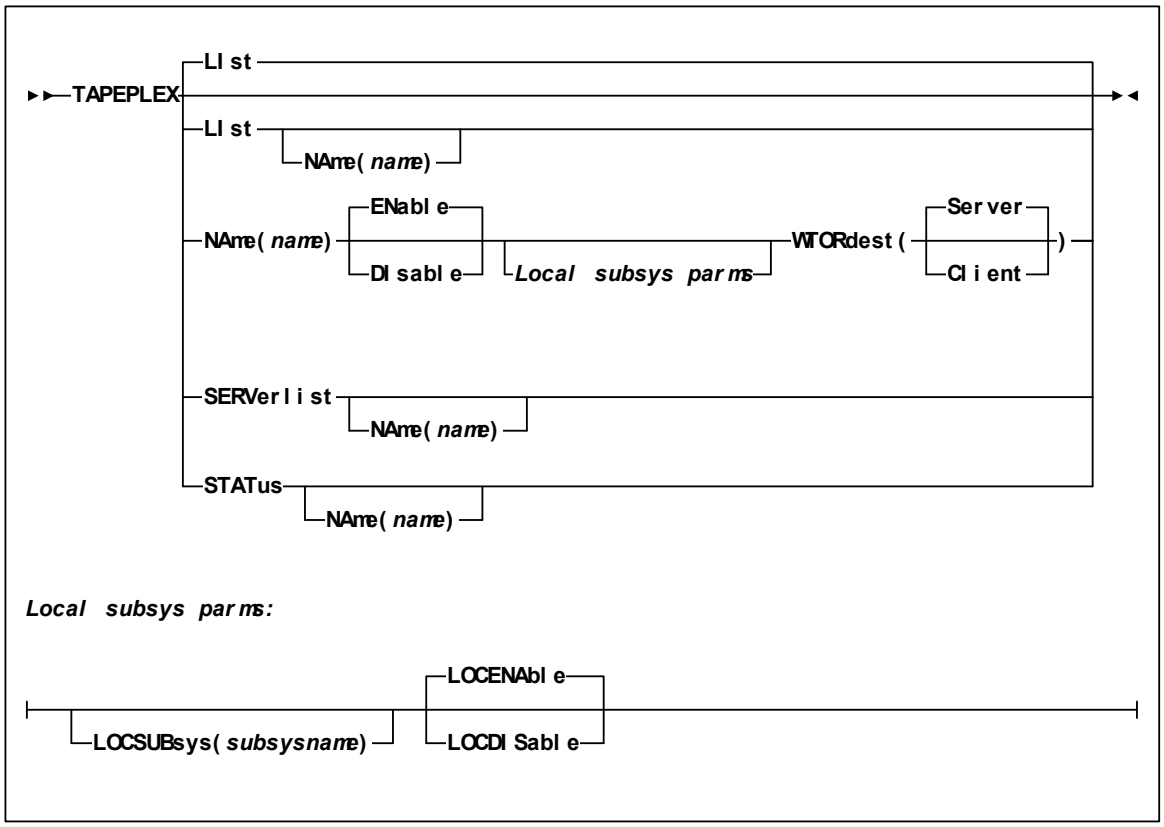


FIGURE 2-6 TAPEPLEX Syntax

Parameters

LIst

displays information about all libraries or a specified TapePlex.

NAME

The TapePlex name.

name

The TapePlex name.

Note –

- LIst is the default when **no** parameters are specified on the TAPEPLEX command. In this case, all libraries are listed.
- You can specify LIst with other parameters. In this case, the information about the specified TapePlex is displayed after the other parameters are processed.

NAME

Specifies the name of the TapePlex to be defined or modified.

name

The TapePlex name:

- The first character must be an alpha character.
- 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 | DISable

Enables or disables the specified TapePlex for selection for allocation or mount requests.

LOCSUBsys

specifies a local HSC and/or MVS/CSC subsystem.

subsysname

The subsystem name.

LOCENAble | LOCDISable

Enables or disables the specified HSC or MVS/CSC subsystem for use as a local path to the TapePlex. LOCDISable can be used to force the SMC to ignore the local subsystem and access the TapePlex via a remote server.

WTORdest

Specifies the console to which HSC WTOR messages are sent:

Client

the remote HSC sends selected WTORs to the SMC client without issuing the WTOR message. The SMC client then issues the WTOR to the operator of the system that the SMC executes on. The SMC redrives the mount or dismount request to the server supplying the WTOR response.

Server

WTOR messages are issued on the HSC server console.

SERVerlist

Displays defined libraries, their attributes and associated servers.

NAME

specifies the TapePlex name for display.

name

The TapePlex name.

STATus

Displays current status of one or more libraries. The TapePlex status is active, inactive, or disabled. For an active TapePlex, the TapePlex status lists the name of the current server or local subsystem. STATus does not perform a RESYNChronize.

NNAME

specifies the TapePlex name for status display.

name

The TapePlex name.

TAPEREQ Control Statement

The TAPEREQ control statement can route tape data sets to VSM and pass a Management Class to VSM or a policy to SMC.

Syntax

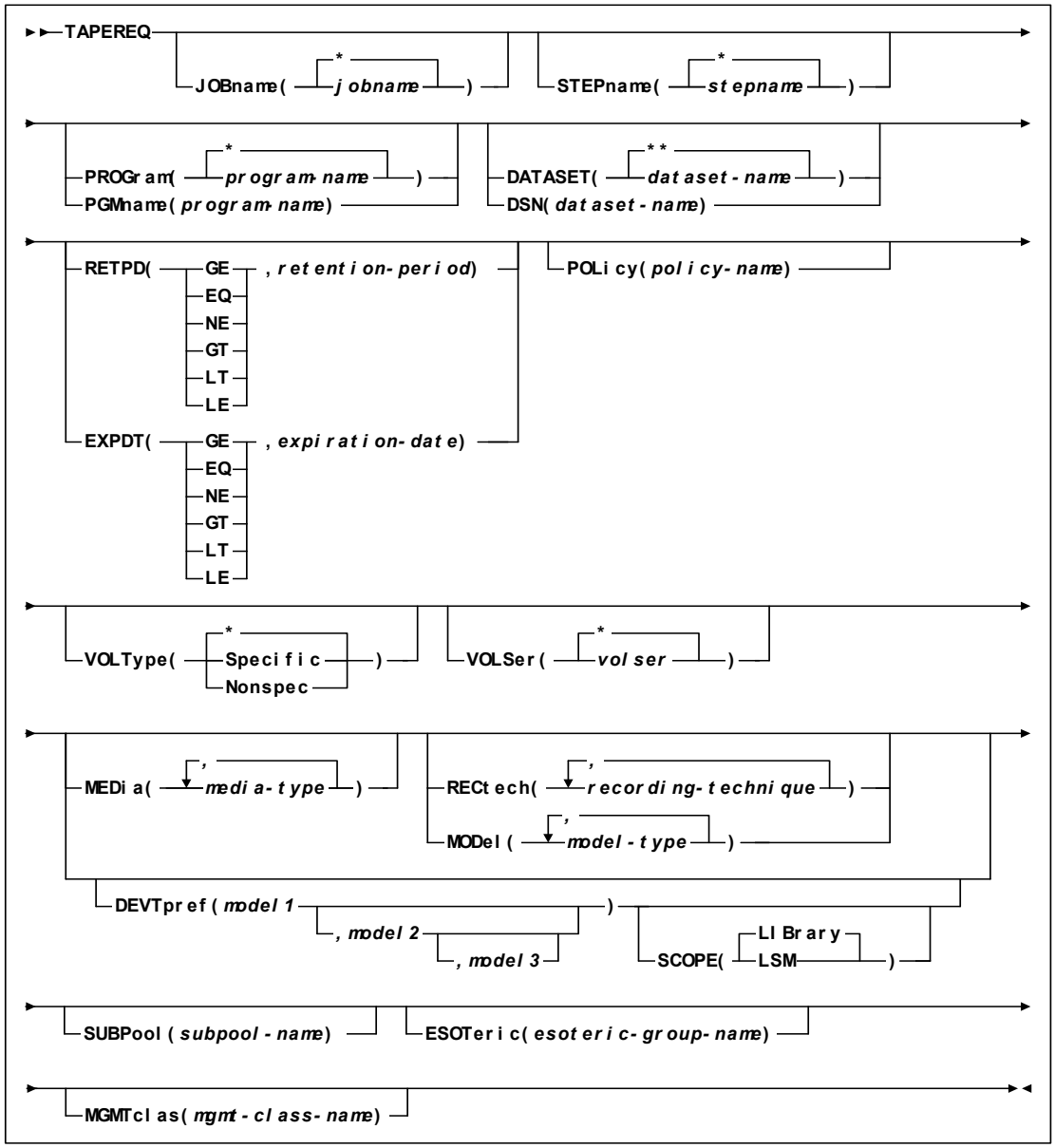


FIGURE 2-7 TAPEREQ Control Statement Syntax

Parameters

The TAPEREQ control statement is essentially an IF-THEN statement. **IF** the Input (selection criteria) parameters are met, **THEN** the Output (media and format) parameters may apply.

Input Parameters

Note – The JOBname, STEPname, PROGram, and DATASET parameter values can include the wild card characters described in TABLE 2-1.

TABLE 2-1 TAPEREQ Wild Card Characters

% or ?	any single nonblank character
*	any character string (length zero to 8) not to exceed one qualification level. For example, A.B.* matches A.B and A.B.C, but does not match A.B.C.D.
**	used only in DATASET. Indicates any number of qualifiers (zero or more). Cannot be used with any other characters within a qualifier. For example, A.B.** matches A.B, A.B.C, A.B.C.D, A.B.C.D.E, and so forth.

JOBname

specifies the job name. If not specified, the default is *.

jobname

the job name.

Note – If you are coding a TAPEREQ statement for DFHSM, you must specify JOBname as a selection criteria. Using DATASET can cause unpredictable results.

STEPname

specifies the step name. If not specified, the default is *.

stepname

the step name.

PROGram | PGMname

specifies the program name. If not specified, the default is *.

program-name

the program name.

Note – This parameter is not supported in JES3 environments.

DATASET | DSN

specifies the data set name. If not specified, the default is **.

dataset.name

the data set name.

Note –

- If you are coding a TAPEREQ statement for DFHSM, you must specify JOBname as a selection criterion. Using DATASET can cause unpredictable results.
- The name you specify on this parameter might be different from the value coded in the DSN parameter on a DD statement. For example, the following can be coded on a DD statement if a data set is temporary:

  DSN=&&ABC

  However, the actual data set name is **not** &&ABC. Similarly, if the DSN parameter refers back to a previous DD statement, the data set name is resolved to the referred-to data set. Thus, the data set name referred to must be coded on the DATASET parameter in order for the TAPEREQ control statement to be processed.

RETPD

specifies the retention period for the data set and the logical comparison. If you do not specify either RETPD or EXPDT, the default value is RETPD(GE,0) (any expiration date will match this criterion):

EQ

equal to.

NE

not equal to.

GT

greater than.

GE

greater than or equal to.

LT

less than.

LE

less than or equal to.

retention-period

the retention period in days as a 1- to 4-digit decimal number.

Note – This parameter is not supported in JES3 environments.

EXPDT

specifies the expiration date and the logical comparison.

EQ

equal to.

NE

not equal to.

GT

greater than.

GE

greater than or equal to.

LT

less than.

LE

less than or equal to.

expiration-date

the expiration date in *YYDDD* or *YYYY/DDD* format.

Note – This parameter is not supported in JES3 environments.

POLicy

optionally, specifies the name of a policy defined by an SMC POLicy command.

POLicy is mutually exclusive with VOLtype, MEDia, RECtech, MODel, SUBPool, ESOTeric, and MGMTclas.

policy-name

the policy name.

The SMC uses this policy name to determine **all** policy information associated with an allocation or mount request.

Note – The named policy must be defined using a POLicy command before the TREQDEF command is processed.

VOLType

specifies a specific or scratch volume. If omitted, the default value is * (either specific or nonspecific matches).

Specific

a specific volume.

Nonspec

a nonspecific (scratch) volume.

Note – The TAPEREQ parameter never overrides the actual media type of the specific volume returned by the server.

VOLSer

specifies one or more volsers. If not specified, the default value is * (any volume). Using a combination of TAPEREQ statements and POLicy commands, SMC allows you to specify allocation policy based on specific volume serial numbers.

volser

the volser.

Note –

- VOLSer requires the POLicy parameter.
- VOLSer is mutually exclusive with VOLType, MEDia, RECtech, MODel, SUBPool, ESOTeric, and MGMTclas.

Output Parameters

The MEDia, RECtech, and MODel parameters have a value of Virtual for VSM only. Specifying Virtual on any of these three parameters will route the data set to a VTV mounted on a VTD. Virtual does not apply to NCS without VSM installed. The MGMTclas parameter does not apply to NCS without VSM installed.

Caution – If you specify a Management Class on the MGMTclas parameter, you must specify both MEDia(V) and RECtech(V).

MEDia

specifies the volume media.

Virtual

specifies that VSM will route the data set to a VTV mounted on a VTD.

RECtech

specifies the recording technique.

Virtual

specifies that VSM will route the data set to a VTV mounted on a VTD.

MODel

specifies the transport model.

Virtual

specifies that VSM will route the data set to a VTV mounted on a VTD.

DEVTpref

requests device preferencing for a particular allocation request. The use of device preferencing is restricted to StorageTek's 36-track class transport models: 4490, 9490, and 9490EE.

Note – The following rules apply when using the MEDia, MODel, and RECtech parameters in conjunction with DEVTpref:

- The MODel parameter must include the models specified in the DEVTpref parameter.
- The RECtech parameter must include valid values for each model specified in the DEVTpref parameter.
- The MEDia parameter must include valid values for each model specified in the DEVTpref parameter.

Note – Device preferencing is managed by SMC. If DEVTpref is specified and the list of devices includes 36-track and other devices (e.g., 18-track) SMC preferences 36-track drives first, followed by other types in the list. See the *SMC Configuration and Administration Guide* for more information.

Note –

model1

specifies the transport model number of the preferred device. These transports are given a higher selection probability than the alternate transport model.

model2

specifies the transport model number of the first alternate device. It must be separated from the *model1* by either a comma or a blank.

model3

specifies the transport model number of the second alternate device. It must be separated from *model2* by either a comma or a blank.

If *model3* is omitted, the transport model number not specified by *model1* or *model2* is assumed to be the second alternate device.

TABLE 2-2 shows the alternate device order if only *model1* is entered:.

TABLE 2-2 Alternate Device Default Value

Preferred Device (*model1*):	*First Alternate Device* (*model2*):	*Second Alternate Device* (*model3*):
4490	9490	9490EE
9490	4490	9490EE
9490EE	4490	9490

Entering DEVTpref(9490,4490) is functionally equivalent to DEVTpref(9490,4490,9490EE).

SCOPE

specifies the scope of device preferencing. You can use SCOPE(LSM) to attempt to reduce pass-thru events. SCOPE is valid only when the DEVTpref parameter is specified on the same TAPEREQ statement.

LIBrary

specifies that the scope is the entire ACS (the default).

In a multiple-ACS configuration, the way that the SMC processes SCOPE(LIBrary) depends on the following:

- The ability to automate the mount
- The type of request; specific versus scratch

A specific volume request is confined to the ACS where the volume resides so that the mount can be automated. The transports in all other ACSs, and all nonlibrary transports are marked ineligible for selection. SMC then applies SCOPE(LIBrary) to one ACS, giving all of the preferred models in that ACS a higher selection probability.

A scratch request can be directed to any ACS, whether or not it contains scratch volumes. In this case, SMC applies SCOPE(LIBrary) to all ACSs, giving all of the preferred models in all ACSs a higher selection probability.

To prevent a scratch request from being directed to an ACS with no scratch volumes, use ALLOCDef Zeroscr (ON).

LSM

specifies that the scope is the LSM level. That is, all of the preferred models within an LSM are to be given a higher selection probability than all of the alternate models within that same LSM, while maintaining the ability to automate tape mounts.

Each LSM is processed individually without regard to the models attached to other LSMs in the ACS. This means that if the preferred model is not available in the LSM where the volume resides, an alternate transport in the same LSM is allocated preventing a pass-thru of the volume to a different LSM.

Use SCOPE(LSM) if reducing pass-thru events is more important than allocating a preferred transport in a different LSM.

SUBPool

specifies the scratch subpool that contains volumes used to satisfy nonspecific requests. For more information about scratch subpool management, see *SMC Configuration and Administration Guide*.

poolname

the subpool name.

ESOTeric

specifies the esoteric that defines the list of eligible transports to be used to satisfy a tape request.

To route a data set to a VTD, specify one of the esoteric names that you defined during configuration. For VSM, esoteric definition and substitution is different in JES2 and JES3. For more information on creating and using VSM esoterics for TAPEREQ statements, see *Installing ELS*.

Note – If the esoteric contains devices not currently in the Eligible Device List, SMC allocation cannot add devices to the list, but can exclude devices not in the specified esoteric.

---

esoteric-name

the esoteric name.

MGMTclas

specifies a Management Class you defined on the MGMTclas control statement.

mgmt-class-name

the Management Class name.

Note – NCS does not support the DUPlex parameter on TAPEREQ statements, only on MGMTclas statements.

CHAPTER 3

HSC Support for VSM

This chapter contains reference information about the HSC features that support VSM.

ACTMVCGN

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.

Interfaces:	SLUADMIN utility only
Subsystem Requirements:	Active HSC required only when specifying the MVCPOOL parameter

Syntax

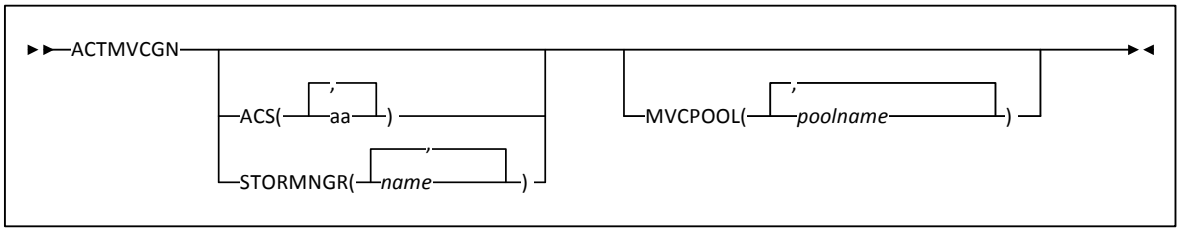


FIGURE 3-1 ACTMVCGN syntax

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.

mmmmmmmmmmmmm

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 ELS appliance that the CSV-based output is filtered against.

ssssssss

is the Subsystem Name of a VLE appliance. 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 ACTMVCGN JCL:

SLUSMVON

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

SLUSMVOF

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

DISPLAY Command

For VSM, the HSC Display command displays the data set and date and time loaded if MGMTclas and/or MVCPool control statements are active and the HSC features set by the FEATures PARMLIB control statement.

Syntax

FIGURE 3-1 Display Command

Parameters

FEATures

displays the HSC features set by the FEATures PARMLIB control statement.

MGMTDEF

displays the data set and date and time loaded if MGMTclas control statements are active.

MVCDEF

displays the data set and date and time loaded if MVCPool control statements are active.

FEATURES Control Statement

The HSC FEATures control statement specifies which VSM features are enabled.

Syntax

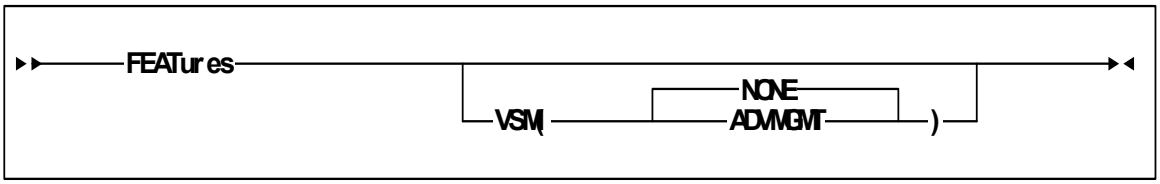


FIGURE 3-2 FEATures **Control Statement**

Parameters

VSM

specifies which VSM Management Features are enabled.

NONE

Basic Management only is enabled; the Advanced Management Feature is not enabled (the default). STORclas statements, the MGMTclas statement MIGpol, RESTIME, CONSRC, CONTGT and REPlicat parameters, and EXPORT and IMPORT are disabled.

ADVMGMT

Both Basic and the following Advanced Management Features are enabled:

- STORclas statements.
- MGMTclas statement MIGpol, RESTIME, CONSRC, CONTGT, and REPlicat parameters.
- EXPORT and IMPORT.

If the FEATures PARMLIB control statement is not specified, Basic Management only is enabled.

MERGECDS Utility

The enhanced MERGEcds utility can reconfigure a CDS or merge CDSs with VSM volume records.

Syntax

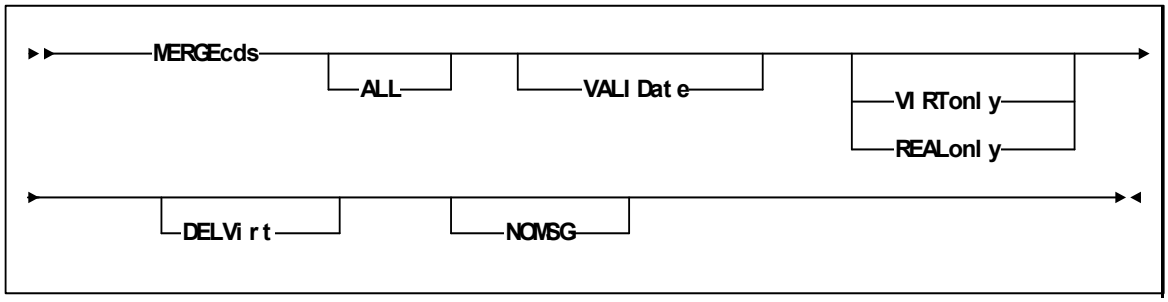


FIGURE 3-3 MERGEcds Utility Syntax

Parameters

VALIDate

specifies to only validate that the configurations to be reconfigured or merged are compatible, but not do the operation. MERGEcds VALIDate reports any duplicate, in-transit, and errant volumes.

ALL

specifies to copy volume information for all ACSs and VTSSs from the “from” CDS to the “to” CDS. For a CDS merge, the ACS ID and LSM IDs, and VTSS names must match.

If you do not specify ALL, MERGEcds reads the parameters specified in the SLSMERGE DD statement, which specify the ACSs, LSMs, and VTSSs whose volume information you want to merge or reconfigure. ALL and SLSMERGE DD are mutually exclusive.

You also specify the ALL parameter to convert a CDS to extended format.

VIRTonly

specifies to use only VSM volume records.

REALonly

specifies to use only real Nearline volume records.

As shown in FIGURE 3-3, VIRTonly and REALonly are mutually exclusive. See on page 231 for more information about the MERGEcds parameter interactions.

DELVirt

specifies that VTV and MVC volume information is not copied to the “to” CDS if **both** of the following are true:

- The VTVs and MVCs defined in the “from” CDS are either uninitialized or empty. An empty VTV is not VTSS resident and has no current MVC copies. An empty MVC contains no current VTVs and does not have an assigned STORCLAS. An MVC is empty when %USED is 0% and %AVAIL is 100% on an MVC Report or Q MVC display. Follow the procedure in *Managing VTCS* for removing MVCs from the pool.
- The uninitialized or empty VTVs and MVCs in the "from" CDS are not defined in the "to" CDS. That is, no duplicate volsers exist.

NOMSG

suppresses message SLS4245I, which displays the volser of an MVC or VTV that was not copied to the “to” CDS. NOMSG has no effect if you do not also specify DELVirt.

MERGEcds Parameter Interactions

TABLE 3-1 describes the interactions of the MERGEcds parameters.

TABLE 3-1 MERGEcds Parameter Interactions

If you specify...	The SLSMERGE DD file is...	And MERGEcds...
MERGE ALL	not read	uses both real Nearline volume records and VSM volume records but does not allow renaming the VTSS.
MERGE ALL REALonly	not read	uses only real Nearline volume records (current MERGEcds behavior).
MERGE ALL VIRTonly	not read	uses only VSM volume records but does not allow renaming the VTSS.
MERGE	read	uses both real Nearline volume records and VSM volume records and allows renaming the VTSS.
MERGE REALonly	read and MERGEcds honors the FACS/TACS and FLSM/TLSM subparameters. For more information, see “SLSMERGE” on page 232.	uses only real Nearline volume records (current MERGEcds behavior).
MERGE VIRTonly	read and MERGEcds honors the FVTSS/TVTSS subparameters. For more information, see “SLSMERGE” on page 232.	uses only VSM volume records and allows renaming the VTSS.
MERGE REALonly VIRTonly	not read	operation fails, REALonly and VIRTonly are mutually exclusive.

JCL Requirements

The following are the required and optional statements for the MERGEcds JCL:

SLSFCNTL

specifies the current primary copy of the “from” HSC CDS.

SLSFCTL2

specifies the current secondary copy of the “from” HSC CDS. This is only required if HSC has been set up to run with a secondary copy.

SLSFSTBY

specifies the current standby copy of the “from” HSC CDS. This is also only required if HSC has been set up to run with a standby copy.

SLSIN

specifies the input to the SLUADMIN program (MERGEcds utility name and parameters).

SLSMERGE

specifies the "from" and "to" ACSs, LSMs, or VTSSs to use for a merge. This parameter is optional and is mutually exclusive with the MERGEcds ALL parameter.

Syntax

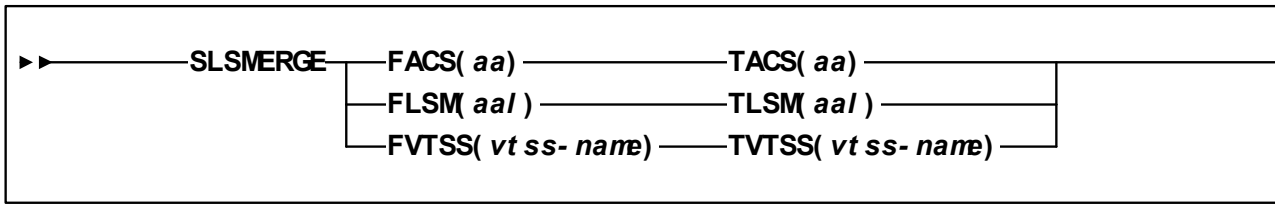


FIGURE 3-4 SLSMERGE DD Statement Syntax

ACS and LSM IDs are valid hexadecimal values for Nearline systems. The *vtss-name* is a VTSS name.

FACS=*acs-id*

specifies the “from” ACS.

TACS=*acs-id*

specifies the “to” ACS.

FLSM=*lsm-id*

specifies the “from” LSM.

TLSM=*lsm-id*

specifies the “to” LSM.

FVTSS=*vtss-name*

specifies the “from” VTSS.

TVTSS=*vtss-name*

Caution –

- Note that the “real” parameters (FACS, TACS, FLSM, TLSM) act as selection criteria, while the “virtual” parameters (FVTSS, TVTSS) do not act as selection criteria, they act only as rename criteria. For example:

```
MERGE FVTSS(VTSS18) TVTSS(VTSS17)
```

  In this example, all VTV records are copied to the new CDS, but the Resident VTSS field is changed from VTSS18 to VTSS17.

- If you specify FACS/FLSM, only volume records for real tapes in the specified ACS/LSM are copied to the new CDS. In this case, you must specify MERGECDS REALonly as described in Table 3-1 on page 231.
- If you specify FVTSS to rename a VTSS you must specify MERGECDS VIRTonly as described in Table 3-1 on page 231.
- Also note that you cannot specify "real" parameters (FACS, TACS, FLSM, TLSM) and the "virtual" parameters (FVTSS, TVTSS) in the same SLSMERGE DD statement. In this case, you must run two MERGECDS jobs.

  Similarly, you cannot specify the same TOVTSS statement multiple times in a single MERGECDS job. For example, if you want to change the Resident VTSS field to VTSS17 for VTVs where the current field is set to VTSS18 and those where the current field is set to VTSS19, you must run two separate MERGECDS and the corresponding SLICREAT(e) jobs for each rename operation.

  Example:

1. Run SLICREAT(E) to build a temporary CDS.
2. Run MERGECDS FVTSS(VTSS18) TVTSS(VTSS17) using the old CDS as input.
3. Run SLICREAT(E) to build a new permanent CDS.
4. Run MERGECDS FVTSS(VTSS19) TVTSS(VTSS17) using the temporary CDS from Step 1 as input.

MGMTCLAS Control Statement

The MGMTclas control statement defines a VSM Management Class. As shown in the following sections, the VSM feature you enable determines which MGMTclas parameters are valid.

Syntax - Basic Management Feature

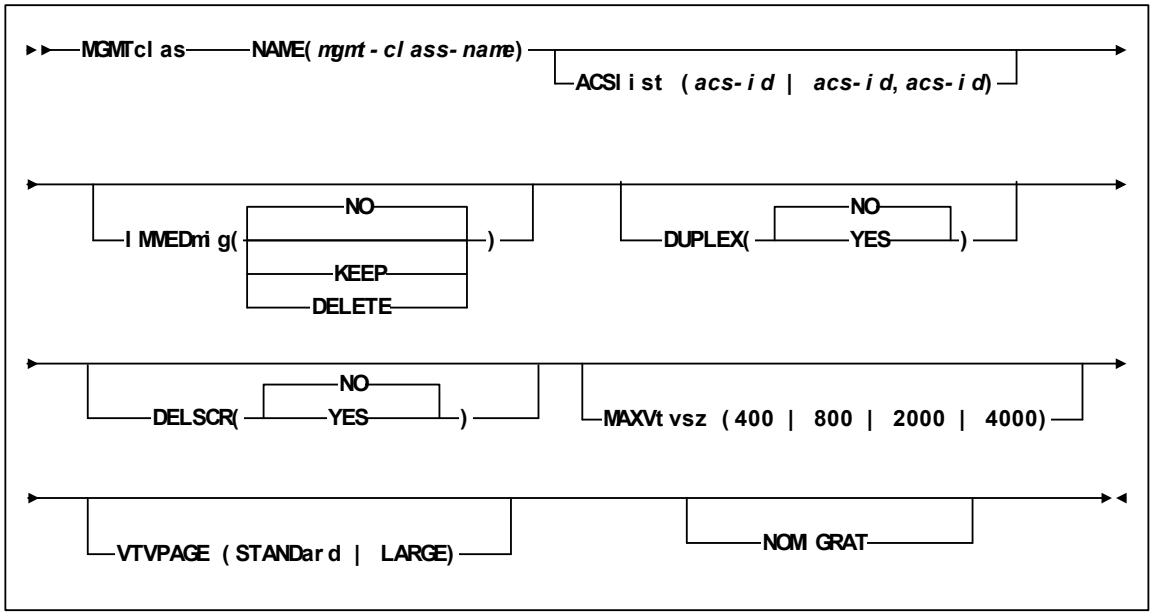


FIGURE 3-5 MGMTclas Control Statement Syntax - Basic Management Feature

Parameters - Basic Management Feature

NAME

specifies the name of the Management Class.

mgmt-class-name

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

ACSlist

specifies the ACSs from which RTDs and MVCs are selected.

ACSlist is optional; if not specified, the default is the ACS specified on the CONFIG DEFLTACS parameter.

See TABLE 3-2 on page 242 for information about using the DUPlex and ACSlist parameters.

acs-id | *acs-id,acs-id*

Specify either one or two ACS IDs. An ACS ID has a hexadecimal value from 00 through FF.

IMMEDmig

specifies whether VSM immediately migrates a VTV after dismounting it.

NO

specifies that VSM does not immediately migrate the VTV, but migrates it according to standard VSM migration criteria (the default).

KEEP

specifies that VSM immediately migrates a VTV and keeps a copy resident on the VTSS until the VTV become eligible for deletion.

DELETE

specifies that VSM immediately migrates the VTV and then deletes it from the VTSS.

Note – IMMEDmig KEEP and IMMEDmig DELETE are mutually exclusive with CONFIG HOST NOMIGRAT. If you specify both, the IMMEDmig value overrides NOMIGRAT, and VTCS does not issue a message about this override.

DUPlex

specifies whether VSM will migrate two copies of the VTV to two MVCs.

See TABLE 3-2 on page 242 for information about using the DUPlex and ACSlist parameters.

NO

Do not duplex the VTV (the default).

YES

Duplex the VTV.

DELSCR

specifies whether VSM deletes scratched VTVs.

This parameter is optional.

NO

do not delete scratched VTVs (the default).

YES

delete scratched VTVs.

Caution – When you scratch a VTV with DELSCR YES attribute, **VSM erases the VTV data at scratch synchronization time**, which eliminates the ability "unscratch" a VTV to recover data!

Also note that when using HSC to perform scratch synchronization, **it is possible that a volume that is scratch** in the TMC at the beginning of scratch synchronization run and also scratch in the CDS from the previous scratch update run (and thus is in the list for HSC to scratch in the CDS) is accessed by a job during the scratch update run and written to and **made non-scratch** by the TMS in the TMC. **In this case, it is still possible for HSC to scratch the volume** because it was in the originally extracted list of volumes to be scratched. Therefore, **StorageTek strongly recommends** that you **do not** run any jobs that use scratches during HSC scratch synchronization. For more information about HSC scratch synchronization with the Scratch Conversion Utility (SLUCONDB), see *HSC System Programmer's Guide for MVS*.

For more information about ExLM scratch synchronization with the SYNCVTV function, see "Using ExLM to Manage Nearline and VTCS Resources" in *ExLM System Administrator's Guide*.

MAXVtvsz

specifies the maximum VTV size. Valid values for this parameter depend on both the CDS level and the microcode levels of the applicable VTSSs.

400

400MB (the default, if not specified).

800

800MB. The CDS must be at a E level or above.

2000

2GB. The CDS must be at a G level or above.

4000

4GB. The CDS must be at a G level or above.

Note –

- The size of a VTV changes *only* after it goes through a scratch cycle. Therefore, if you change the Management Class and DISP=MOD, then it will still retain the original size.
- If you specify a VTV size that is not supported by the configuration, VTCS issues warning messages and `MAXVtvsz` defaults to the largest VTV size supported by the configuration.
- `MAXVtvsz` **does not** apply to VSM2s.
- `MAXVTVSZ(2000 | 4000)` requires VSM4/VSM5 microcode D02.02.00.00 or VSM3 microcode N01.00.77.00. No installed option is required.

NOMIGRAT

specifies that VTVs in the Management Class **are not** candidates for migration, consolidation or export, but **are** candidates to reside on a tapeless VTSS.

Syntax - Advanced Management Feature

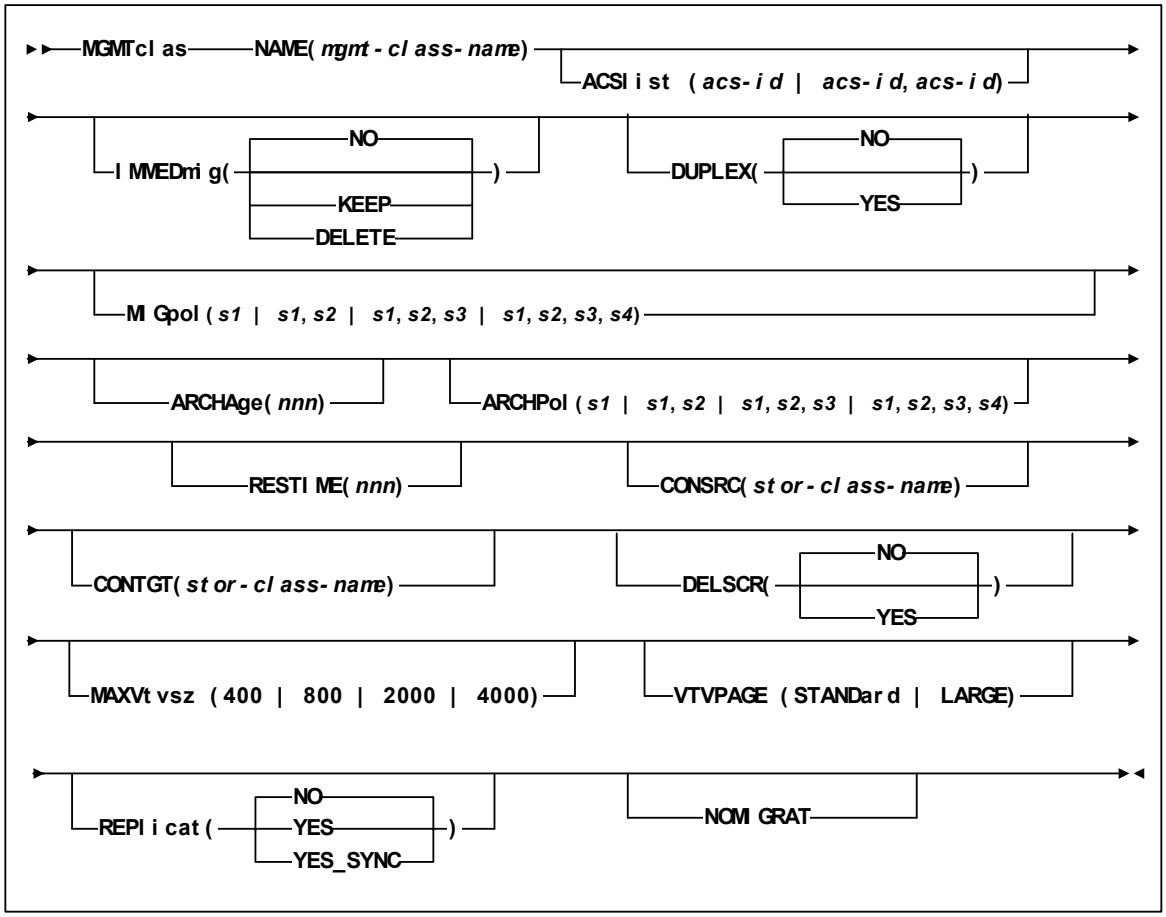


FIGURE 3-6 MGMTclas Control Statement Syntax - Advanced Management Feature

Additional Parameters - Advanced Management Feature

The following MGMTclas parameters are valid for the Advanced Management Feature in addition to the Basic Management Feature parameters described in "Parameters - Basic Management Feature" on page 235.

MIGpol

specifies up to four Storage Classes that specify the ACS and media type of migration MVCs. If you specify:

- One Storage Class, VTCS migrates one copy of a VTV.
- Multiple Storage Classes (with different ACS values, different MEDIA values, or both), VTCS makes multiple copies the VTV to different MVCs in different ACSs.
- Multiple Storage Classes with identical ACS and MEDIA values, VTCS makes multiple copies of the VTV to the same ACS and media type but to different MVCs.

Note – Multiple Storage Classes on MIGpol also affects how:

- VTV recall works.
- MVC space reclamation works.
- How VTV consolidation works

This parameter is optional; there is no default value.

s1 | s1,s2 | s1,s2,s3 | s1,s2,s3,s4

the names of up to 4 Storage Classes that you defined on the STORclas control statement. Greater than two copies requires you to specify CDSLEVEL(V6ABOVE) or greater on the CONFIG statement.

Note – The CONFIG GLOBAL REPlicat parameter specifies when to replicate a VTV (always, or only when changed while mounted).

ARCHAge

specifies the age (in days) of a VTV before it is archived as specified by ARCHPol. If you specify ARCHAge, you must specify ARCHPol.

This parameter is optional; there is no default value. Valid values are 1 to 999.

nnn

the VTV age in days.

ARCHPol

specifies up to four Storage Classes that specify the ACS and media type of the archive MVCs. If you specify:

- One Storage Class, VTCS archives one copy of a VTV.
- Multiple Storage Classes (with different ACS values, different MEDIA values, or both), VTCS archives multiple copies of the VTV to different MVCs in different ACSs.
- Multiple Storage Classes with identical ACS and MEDIA values, VTCS archives multiple copies of the VTV to the same ACS and media type but to different MVCs.

Note – Multiple Storage Classes on ARCHPol also affects how:

- VTV recall works.
- MVC space reclamation works.
- How VTV consolidation works

This parameter is optional; there is no default value. If you specify ARCHPol, you must specify ARCHage.

*stor-clas-name*1...*stor-clas-namen*

the names of one or more Storage Classes that you defined on the STORclas control statement. Greater than two copies requires you to specify CDSLEVEL(V6ABOVE) or CDSLEVEL(V6ABOVE) on the CONFIG statement.

RESTIME

specifies how long VTCS attempts to keep a VTV as VTSS-resident before becoming a preferred automatic migration candidate.

This parameter is optional; there is no default value. Valid values are 1 to 9999. Note that 9999 specifies that the VTVs in this Management Class are resident permanently unless VTSS space management requires VTCS to automigrate the VTV and then delete it from the VTSS.

The RESTIME and IMMEDmig(DELETE) parameters are mutually exclusive.

nnnn

the residency time in hours.

CONSRC

specifies the Storage Class that species a preference for the source MVC ACS and media for consolidation of VTVs that are migrated and copied to multiple different MVC locations or media types. If the MVC in the specified Storage Class is unavailable, and the specified Storage Class is not the last (in order specified in the migration policy), VTCS will use the MVC associated with the last Storage Class. If the MVC in the specified Storage Class is unavailable and the specified Storage Class is the last (in order specified in the MIGpol parameter), VTCS will use the MVC associated with the previous Storage Class (in order specified in the MIGpol parameter).

This parameter is optional; there is no default value.

stor-class-name

the name of a Storage Class that you defined on the STORclas control statement.

CONTGT

specifies the Storage Class that determines the output MVC ACS and media for VTV consolidation. Note that the media preferencing is in the opposite order of the list of media types specified on the Storage Class.

This parameter is optional; there is no default value. If you do not specify a value for CONTGT, VTCS selects the output MVC as follows:

- For single-ACS and dual-ACS configurations, the media selection order for VTV consolidation.
- For multiple ACS systems, VTCS selects MVCs from the default ACS specified by the CONFIG DEFLTACS parameter.

stor-class-name

the name of a Storage Class that you defined on the STORclas control statement.

REPlicat

specifies whether VSM replicates the VTV.

NO

Do not replicate the VTV (the default).

YES

Asynchronously replicate the VTV.

YES_SYNC

Synchronously replicate the VTV.

Note – Synchronous replication requires enablement via the CONFIG GLOBAL SYNCHREP parameter. For more information, see "CONFIG Utility GLOBAL Statement" on page 22.

Usage Notes

Note the following:

- `ACSLIST` is mutually exclusive with `MIGPOL`, `ARCHAGE` and `ARCHPOL`.
- `IMMEDMIG` is mutually exclusive with `IMMDELAY`.
- `DUPLEX` is mutually exclusive with `MIGPOL`, `ARCHAGE` and `ARCHPOL`.
- `MIGPOL` is mutually exclusive with `ACSLIST` and `DUPLEX`.
- `CONSRC` has a corequisite of `MIGPOL`.
- `ARCHAGE` has a corequisite of `ARCHPOL` and is mutually exclusive with `ACSLIST` and `DUPLEX`.
- `ARCHPOL` has a corequisite of `ARCHAGE` and is mutually exclusive with `ACSLIST` and `DUPLEX`.
- `IMMDELAY` is mutually exclusive with `IMMEDMIG`.

Using the DUPlex parameter

TABLE 3-2 describes possible scenarios using the DUPlex and ACSlist parameters.

TABLE 3-2 MGMTclas ACSlist/DUPlex Scenarios

If DUPlex is set to...	And ACSlist specifies...	Then VSM...
YES	two ACSs	migrates the VTVs to two MVCs, one in each ACS. (This scenario is the normal one for duplexing to two ACSs.)
YES	one ACS	migrates the VTVs to two MVCs in the ACS specified
NO	two ACSs	ignores the DUPlex policy and migrates the VTVs to two MVCs, one in each ACS.
NO	one ACS	migrates the VTVs to one MVC in the ACS specified

MGMTDEF Command

The MGMTDEF command loads the MGMTclas, STORclas, VTSSLST, VTSSSEL, STORLST, and STORSEL statements from a specified definition data set.

Syntax

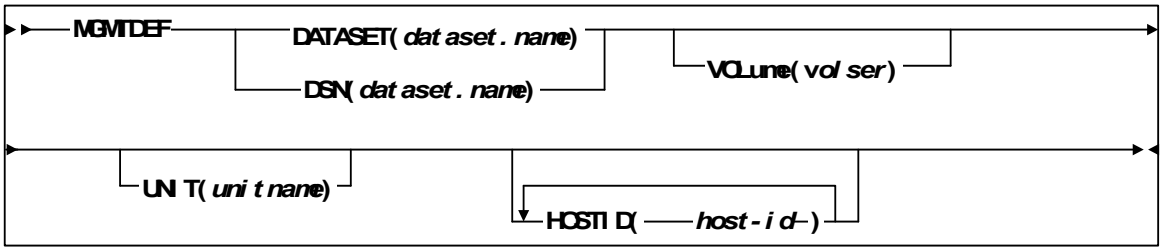


FIGURE 3-7 MGMTDEF Command

Parameters

DATASET or DSN

specifies the definition data set that contains the MGMTclas and STORclas statements to load.

dataset.name

the data set name.

VOLume

specifies the DASD volume where the definition data set resides. This parameter is optional, unless the data set is not cataloged, or the data set resides on a volume other than the volume indicated by the catalog.

volser

the DASD volser.

UNIT

specifies the DASD device where the definition data set resides.

unitname

the DASD unit name. If the definition data set is not cataloged and this parameter is omitted, the unit name defaults to SYSALLDA.

HOSTID

specifies the host for execution of the MGMTDEF command. This parameter is only valid when MGMTDEF is specified as a PARMLIB control statement.

host-id

specifies the name of one or more hosts from which to execute the MGMTDEF command. Multiple hosts must be separated by commas.

MIGRSEL Control Statement

The MIGRSEL control statement defines migration settings for a Storage Class, and/or VTSS, and/or host. It is loaded by the MGMTDEF command.

Syntax

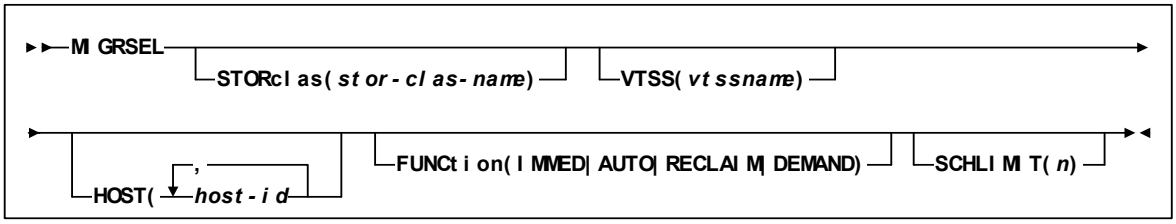


FIGURE 3-2 MIGRSEL control statement syntax

Parameters

STORclas

optionally, specifies a Storage Class to which the migration settings apply. If you do not specify a Storage Class, the statement applies to all Storage Classes.

stor-clas-name

the name of the Storage Class you defined on the STORclas control statement.

VTSS

optionally, specifies a VTSS to which the migration settings apply. If you do not specify a VTSS, the statement applies to all VTSSs.

vtssname

the VTSS name

HOST

optionally, specifies one or more hosts to which the migration settings apply. Any hosts not specified on this parameter ignore the settings. If you do not specify one or more hosts, the settings apply to all hosts.

host-id

a host identifier (maximum 8 characters)

FUNCTION

optionally, specifies the type of migration to which the settings apply:

IMMED

migrations resulting from either of the following:

- MGMTclas IMMEDmig(KEEP)
- MGMTclas IMMEDmig(DELETE)

AUTO

automatic migrate to threshold migration processing

RECLAIM

migrations resulting from MVC DRAIN or RECLAIM requests

DEMAND

migrations resulting from a MIGRATE command or utility (demand migrations)

SCHLIMIT

optionally, de-preferences migration per Storage Class.

n

the preferencing value. Valid values are 0 to 99. The default is 99, which indicates no limit, up to the VTSS MAXMIG value.

Lower values de-preference migration, and you can specify automatic, immediate, demand, and reclaim migrates. Lower values can do the following:

- Optimize MVC usage.
- Preference migration to other Storage Classes.
- Limit migration to keep RTDs available for auto recalls.
- Reduce MVC swapping when workloads change.

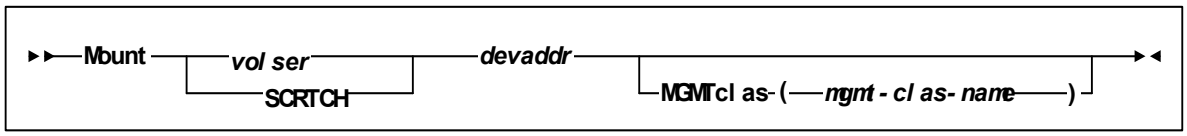
For auto and immediate migration processing, MIGRSEL SCHLIMIT de-preferences migration for the VTSS to storage class relationship. This comparison is not global and only effects requests driven by the individual VTCS host

For demand migration requests, MIGRSEL SCHLIMIT will cause the request to be held if the scheduling of it would cause the number of globally active migration requests on the VTSS that satisfy the same FUNCTION and STORCLAS selection criteria to be exceeded. The migration requests will be released and an MVC picked once the constraint subsides.

MOUNT Command

The enhanced MOUNT command mounts a scratch or specific VTV on a VTD and optionally assigns a Management Class to the VTV.

Syntax



Parameters

volser | SCRTCH

specifies a specific VTV volser or the scratch VTV attribute (SCRTCH).

volser

the volser of a specific VTV.

devaddr

specifies the MVS device address of the VTD to use to mount the VTV.

MGMTclas

specifies a Management Class you defined on the MGMTclas control statement.

mgmt-class-name

the Management Class name.

MVCATTR Control Statement

The MVCATTR control statement assigns attributes to an MVC media name. The MVCATTR control statement is loaded by the MGMTDEF command.

Syntax

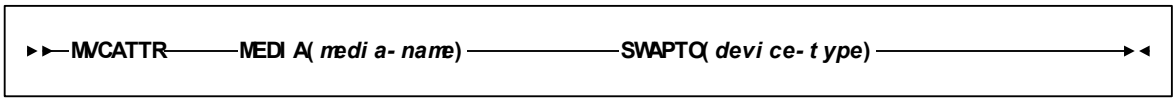


FIGURE 3-3 MVCATTR control statement syntax

Parameters

MEDIA

Specifies the MVC media name to which the attribute is assigned.

Note – Only one MVCATTR should be coded for each MEDIA name.

---

media-name

the MVC media name; see TABLE 3-3.

SWAPTO

Defines the RTD device type the MVC is swapped to (if possible).

device-type

the RTD device type; see TABLE 3-3.

TABLE 3-3 Valid MVC Media Names and Compatible SWAPTO RTD Device Types

Valid MEDIA Names	Compatible SWAPTO Device Types
STK1R	STK1RA34,STK1RB34,STK1RD34,STK1RDE4 **Note: STK1RA34 and STK1RB34 devices are equivalent to VTCS, so you cannot ensure a swap to these specific device types.**
STK1RC	STK1RC34,STK1RD34,STK1RDE4
STK1RD	STK1RD34,STK1RDE4
STK1RDE	STK1RDE4
STK2P	STK2PA34,STK2PB34
STK2PB	STK2PB34
T10000T1	T1A34,T1AE34,T1B34,T1BE34
T10000TS	T1A34,T1AE34,T1B34,T1BE34
T10000E1	T1AE34,T1BE34

TABLE 3-3 Valid MVC Media Names and Compatible SWAPTO RTD Device Types

Valid MEDIA Names	Compatible SWAPTO Device Types
T10000ES	T1AE34,T1BE34
T1B000T1	T1B34,T1BE34
T1B000TS	T1B34,T1BE34
T1B000E1	T1BE34
T1B000ES	T1BE34

Usage

When an error occurs while reading an MVC on an RTD, VTCS may swap the MVC to another RTD to retry the operation. If the swap is for an MVC audit then the selection of the "swap to" device type **excludes** the device type that the error occurred on. Otherwise, the "swap to" device type is restricted to the native device type (device type MVC was created on). However, **note that**:

- T9840D devices are the **preferred** device type for reading an MVC created on **any** T9840 device type (T9840A/T9840B/T9840C/T9840D).
- T10KB devices are the **preferred** device type for reading an MVC created on **any** T10K device type (T10KA/T10KB).

This enhancement provides the MVCATTR control statement, which lets you specify the "swap to" RTD device type for each MVC media name.

Note – It is possible to get the SLS6949I message on a write swap. This will only occur if the MVCATTR requests a swap to the native device type. For example, if you have an MVCATTR for MEDIA(STK1RC) and request SWAPTO(STK1RC34,) then it will look like the swap was influenced because it matched the MVCATTR but in reality it did this by default.

MVCPOOL Control Statement

The MVCPool control statement defines your system's MVC pool and, optionally, Named MVC Pools within that pool.

Syntax

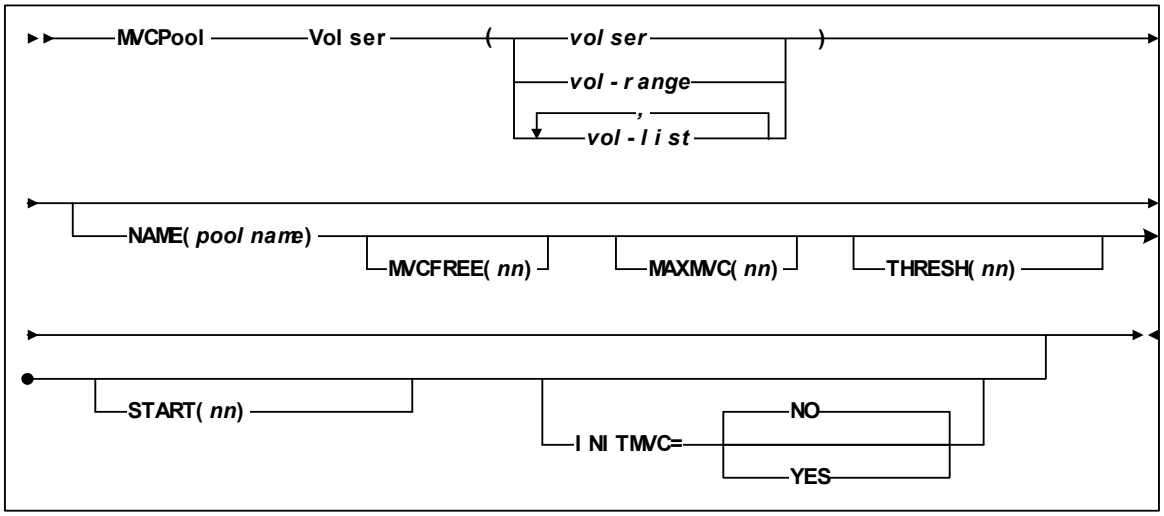


FIGURE 3-8 MVCPool Control Statement Syntax

Parameters

Volser

defines the MVCs.

volser, vol-range, or *vol-list*

the volsers of one or more MVCs. If you specify multiple volume ranges, do not overlap them.

NAME

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

poolname

the MVC Pool name (up to 13 characters).

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

MVCFREE(*nnn)*

specifies the minimum number of free MVCs in the MVC pool. A free MVC has 100% usable space and does not contain any migrated VTVs. Valid values are 0 to 255. If not specified, the CONFIG GLOBAL value (or default) is used.

If free MVCs is equal or less than this value, VTCS issues message SLS6616I and starts an automatic space reclamation.

MAXMVC(*nn)*

specifies the maximum number of MVCs that will be processed in a single space reclamation run. Valid values are 1 to 98. There is no default; if not specified, the CONFIG RECLAIM value (or default) is used.

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

THRESH(*nn)*

specifies the percentage of fragmented space that makes an MVC eligible for demand or automatic reclamation. Valid values are 4 to 98. There is no default; if not specified, the CONFIG RECLAIM value (or default) is used.

START(*nn)*

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

(*MVCs eligible for reclamation*/*Total available MVCs*) * 100

Where:

MVCs eligible for reclamation

is the number of eligible MVCs determined by the THRESHLD parameter.

Total available MVCs

equals the number of eligible MVCs *plus* the number of free MVCs. A free MVC has 100% usable space and does not contain any migrated VTVs.

Valid values are 1 to 98. There is no default; if not specified, the CONFIG RECLAIM value (or default) is used.

INITMVC

specifies whether uninitialized MVCs in the named MVC Pool are initialized when first mounted on an RTD.

NO

Uninitialized MVCs are not initialized.

YES

Uninitialized MVCs are initialized.

Note –

- MVCPOOL INITMVC overrides GLOBAL INITMVC. There is no default for MVCPOOL INITMVC; if not specified for a named MVC Pool the CONFIG GLOBAL value (or default) is used.
- Initialization of MVCs in the DEFAULTPOOL is controlled by the GLOBAL INITMVC specification (or default).
- MVC Initialization applies only to VSM4/5 and requires microcode level D02.05.00.00 or higher. If this level of microcode is not installed on all VTSSs in the configuration, MVC initialization will be limited to the VTSSs that have it installed.

STORCLAS Control Statement

The HSC `STORclas` 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 ELS appliance.

Note – The STORclas control statement is valid only if FEATures VSM(ADVMGMT) is specified.

Syntax

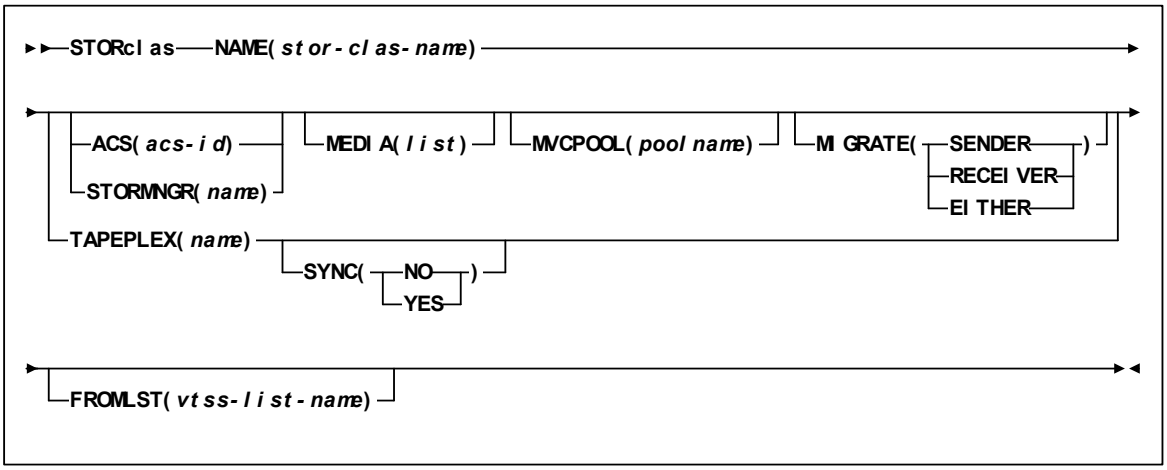


FIGURE 3-9 STORclas Control Statement Syntax

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

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

STORMNGR

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

stormngr

a Subsystem name.

MEDIA

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

list

preference list of media types

MVCPOOL

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.

FROMLST

optionally:

- for Management Classes with REPLICAT(YES) that reference this Storage Class, specifies the source VTSS (in a Cluster) for VTV migration.
- for export of a VTV to another tapeplex, specifies the source VTSS for the export. Specification of this parameter for a TAPEPLEX storage class also implies that any replication processing within the TapePlex is completed before the export is performed.

vtss-list-name

the name of the VTSSLST statement that contains a list of VTSS names. Either the migrate or export from this storage class will be directed from one of the VTSSs in this list.

- If only one of the VTSSs exists in the list, it is used as the source.
- If both VTSSs are in the list, the VTSS with the highest priority is used as the source.
- If both VTSSs have equal priority, the source VTSS is randomly selected.

STORLST Control Statement

The STORLST control statement specifies a list of Storage Classes and their corresponding preferencing.

Note – The STORLST control statement is valid only if FEATures VSM(ADVMGMT) is specified.

Syntax

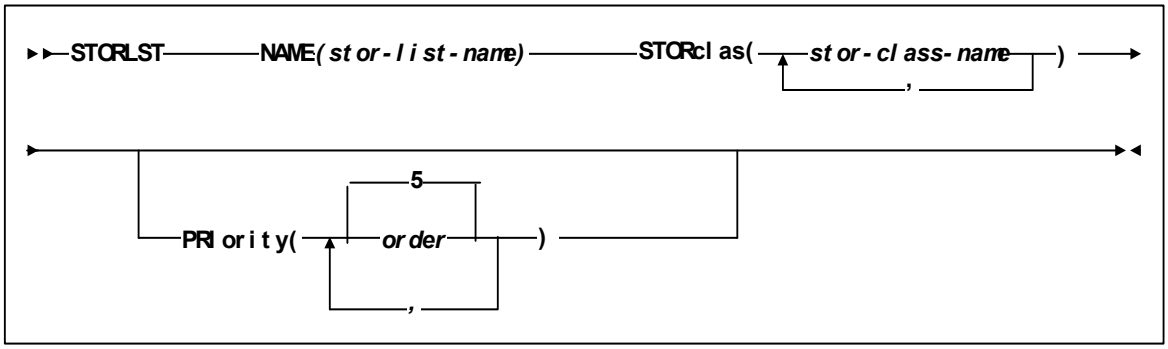


FIGURE 3-10 STORLST Control Statement Syntax

Parameters

NAME

specifies the name of the Storage Class list.

stor-list-name

the list name (a maximum of 8 alphanumeric characters).

STORclas

specifies one to four Storage Classes on the Storage Class list.

stor-clas-name

the name of a Storage Class that you defined on the STORclas control statement.

PRIority

a list of priorities corresponding to the Storage Classes specified on the STORclas parameter. Valid values are 0 to 9 (highest priority), and the default is 5. The Storage Classes are considered (or used) in the order implied by the priority. You can assign the same priority to multiple Storage Classes. For example, if two Storage Classes both have a priority of 9, VTCS randomly orders the two within their priority. A 0 (zero) priority specifies that VTCS selects the Storage Class only if all other Storage Classes are unavailable.

order

the specified priority

The Storage Class list is further qualified by the criteria specified by the MGMTclas and VTSS parameters of the STORSEL statement.

STORSEL Control Statement

The STORSEL control statement defines a Storage Class usage rule that applies to the Storage Class list and its preferencing specified on a referenced STORLST control statement.

Note – The STORSEL control statement is valid only if FEATures VSM(ADVMGMT) is specified.

Syntax

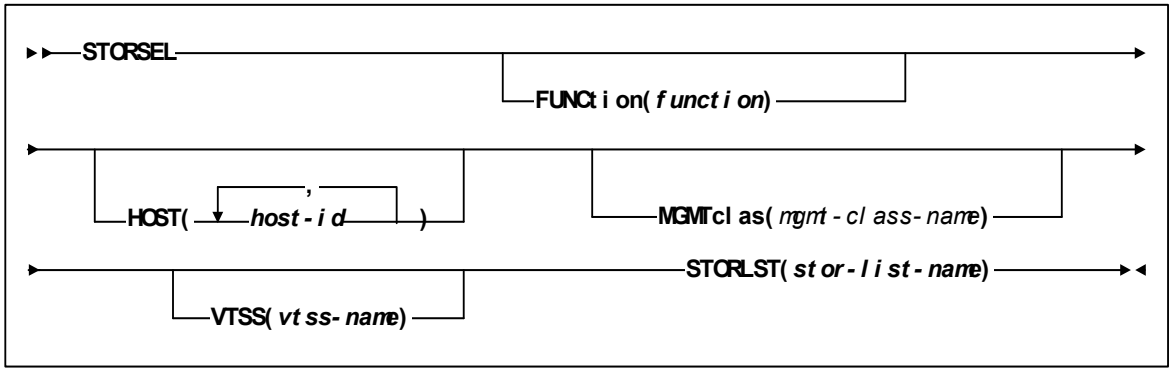


FIGURE 3-11 STORSEL Control Statement Syntax

Parameters

FUNCtion

optionally specifies the VSM function to which the rule applies. Only one function can be specified per statement. If this parm is omitted, it applies to all functions. If all functions do apply, then it makes economical sense to omit the FUNCtion parm, thus reducing the number of statements to 25% of what would otherwise need to be coded.

function

the function name, as described in TABLE 3-4.

TABLE 3-4 STORSEL Functions

Function	Explanation
SPECIFIC	Applies to automatic recall of a specific VTV for mounting. The list of Storage Classes is determined by the specified STORLST statement. This list influences the list of RTDs eligible to mount the MVC in the Storage Class to recall the VTV.
RECALL	Applies to demand recall of a specific VTV for mounting. The list of Storage Classes is determined by the specified STORLST statement. This list influences the list of MVC copies of a VTV to select the optimal MVC for recall of the VTV.
EXPORT	Applies to export. The list of Storage Classes is determined by the specified STORLST statement. This list influences the list of MVC copies of a VTV to select the optimal MVC for export of the VTV.
CONSOLID	Applies to consolidate. The list of Storage Classes is determined by the specified STORLST statement. This list influences the list of MVC copies of a VTV to select the optimal MVC for consolidation of the VTV.

Note – The HOST, MGMTclas, and VTSS parameters are optional and have no default values.

HOST

optionally specifies one or more hosts to which the rule applies. If this parameter is used, any hosts not specified on this parameter ignore the rule. If the parameters is not used the statement applies to all hosts.

host-id

a host identifier (maximum 8 characters).

Note – The Storage Class list specified on the STORLST parameter is further qualified by the criteria specified by the MGMTclas and VTSS parameters.

MGMTclas

specifies a Management Class.

mgmt-class-name

the name of a Management Class that you defined on the MGMTclas control statement.

VTSS

specifies a VTSS as follows:

- For automatic recalls, the VTSS where the recall is performed.
- For all other functions, the VTSS where the VTV previously resided. That can be determined from the VTSS value shown on the `VT QU VTV` display.

STORLST

specifies a list of Storage Classes and their corresponding preferencing.

stor-list-name

the name of a Storage Class list that you defined on the STORLST control statement.

VOLATTR Control Statement

For VSM, the enhanced HSC VOLATTR control statement specifies VTV attributes, including the volser and media type (virtual).

Syntax

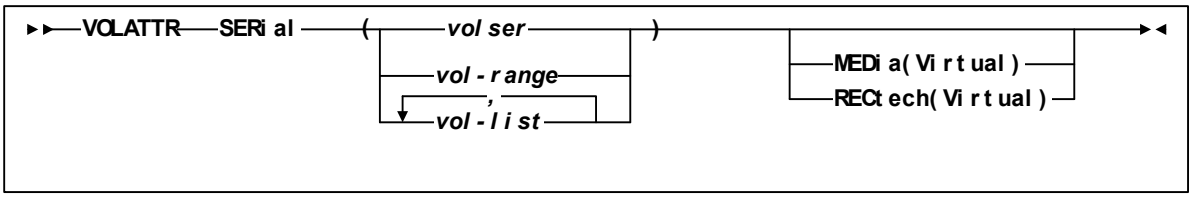


FIGURE 3-12 VOLATTR Control Statement Syntax

Parameters

Unchanged VOLATTR Parameters

The SERial VOLATTR parameter is unchanged but applies to VSM. Figure 1 shows valid values for this parameter; see “Control Statements and Start Procedure” in *HSC System Programmer’s Guide for MVS* for more information.

Tip – When you create VOLATTR statements for VTVs, you use the SERial parameter to specify the VTV volsers.

Caution – On VOLATTR statements for VTVs, do *not* specify duplicate volsers or overlapping volser ranges.

In addition, after you define an initial set of VTV volsers, you can add more volsers but you should not change your initial set of VTV volsers, which wastes HSC CDS space. For example, if you initially define VTVs V00000 - V99999, you can later add VTVs W00000 - W99999 by specifying both volser ranges when you update the VOLATTR statement that specifies your system’s VTVs. If you update the VOLATTR statement to change the volser range from V00000 - V99999 to W00000 - W99999, hosts can still access the original range (V00000 - V99999). If a host scratches a VTV in the original range, however, the VTV cannot be reused, but continues to take up space in the CDS.

VOLATTR Parameters Enhanced for VSM

The following VOLATTR parameters have a new *required* value of Virtual for VTVs only. Virtual does *not* apply to physical HSC volumes.

MEDia

specifies the volume media.

Virtual

specifies that VSM will route data sets to a VTV mounted on a VTD.

RECtech

specifies the recording technique.

Virtual

specifies that VSM will route the data set to a VTV mounted on a VTD.

To define a volume as virtual, you must specify the Virtual keyword for either the MEDia or RECtech. You can also specify both MEDia or RECtech; the keyword must be Virtual for both parameters to define the volume as virtual.

VTSSLST Control Statement

The VTSSLST control statement specifies a list of VTSSs and their corresponding

preferencing. VTCS first determines a system priority for each VTSS, based on various factors. For example, whether the VTSS can service the request, whether the required resources are online/available or whether the VTSS is in a compromised state (high DBU). When more than one VTSS has the highest system priority, VTSSLST priorities can be used to influence which VTSS is used. The VTSSLST is only considered when there is an obvious choice of VTSSs with equal abilities to service a request.

Note – The VTSSLST control statement is valid only if FEATures VSM(ADVMGMT) is specified.

Syntax

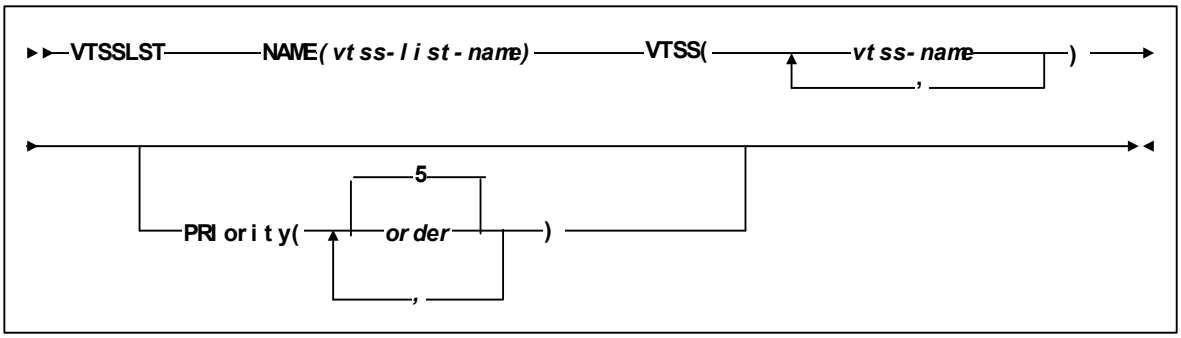


FIGURE 3-13 VTSSLST Control Statement Syntax

Parameters

NAME

specifies the name of the VTSS list.

vtss-list-name

the list name (a maximum of 8 alphanumeric characters).

VTSS

specifies one to eight VTSSs on the VTSS list.

vtss-name

a VTSS name.

PRIority

a list of priorities corresponding to the VTSS names specified on the VTSS parameter.

order

the specified priority. Valid values are 0 to 9 (highest priority), and the default is 5. You can assign the same priority to multiple VTSSs. Within the set of VTSSs with the highest system priority:

- If two or more VTSSs have the same highest VTSSLST priority, VTCS selects randomly from the two (according to factors such as DBU and VSM model).
- A 0 (zero) priority specifies that VTCS selects the VTSS only if all other VTSSs are unavailable (for example, unavailable due to DBU > 95%, VTSS offline, or RTDs offline).

Note – The VTSS list specified on the VTSSLST parameter is further qualified by:

- The function specified on VTSSSEL statement.
- The criteria specified by the MGMTclas, VTSS, STORclas, and MVCpool parameters of the VTSSSEL statement.
- Other factors such as RTD connectivity.

For example, in scratch allocation, the list of VTSSs is reduced to the VTSSs that can meet Management Class policies (such as REPlicat(YES)). If the list of VTSSs is reduced to zero, the request fails.

VTSSSEL Control Statement

The VTSSSEL control statement defines a VTSS usage rule that applies to the VTSS list and its preferencing specified on a referenced VTSSLST control statement.

Note –

- The VTSSSEL control statement is valid only if FEATures VSM(ADVMGMT) is specified.
- VTSSSEL statements are only honored if the VTVs on the MVCs being processed are **not** resident in a VTSS. If the VTVs are resident, then VTCS will ignore the VTSSSEL and migrate the VTV from the VTSS where it is resident.

Syntax

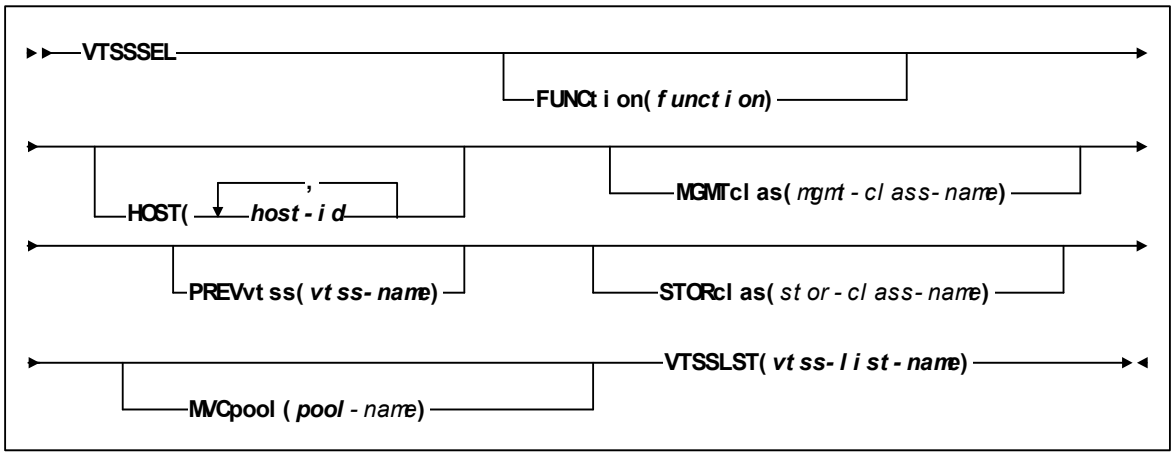


FIGURE 3-14 VTSSSEL Control Statement Syntax

Parameters

FUNCtion

optionally specifies the VSM function to which the rule applies. Only one function can be specified per statement. If this parm is omitted, it applies to all functions. If all functions do apply, then it makes economical sense to omit the FUNCtion parm, thus reducing the number of statements to 25% of what would otherwise need to be coded.

function

the function name, as described in TABLE 3-5.

TABLE 3-5 VTSSSEL Functions

Function	Explanation
SCRATCH	Applies to non-specific (scratch) VTV allocation. The list of eligible VTDs is determined by the specified VTSSLST statement. The PREVVTSS, STORclas and MVCpool parameters do not apply.
SPECIFIC	Applies to specific VTV allocation. The list of eligible VTDs is determined by the specified VTSSLST statement. The STORclas and MVCpool parameters do not apply.
RECALL	Applies to demand recall. The list of eligible VTSSs for recall is determined by the specified VTSSLST statement. The VTSS list also determines the search order for an RTD to service the MVCs selected for recall (derived from the VTVs selected). The MGMTclas parameter does not apply.
RECLAIM	Applies to reclaim. The list of eligible VTSSs for reclaim is determined by the specified VTSSLST statement. The VTSS list also determines the search order for an RTD to service the MVCs selected for reclaim. The MGMTclas parameter does not apply.
DRAIN	Applies to drain. The list of eligible VTSSs for drain is determined by the specified VTSSLST statement. The VTSS list also determines the search order for an RTD to service the MVCs selected for drain. The MGMTclas parameter does not apply.
MOVEVTV	Applies when a ARCHIVE or RECONCIL command that has the MOVEVTVS parameter specified. The list of eligible VTSSs for moving the VTVs via is determined by the specified VTSSLST statement. The VTSS list also determines the search order for an RTD to service the MVCs selected for processing. The MGMTclas parameter does not apply.
AUDIT	Applies to MVC audit. The list of eligible VTSSs for audit is determined by the specified VTSSLST statement. The VTSS list also determines the search order for an RTD to service the MVCs selected for audit. The MGMTclas parameter does not apply.
EXPORT	Applies to export. The list of eligible VTSSs for export is determined by the specified VTSSLST statement. VTSS list also determines the search order for an RTD to service the MVCs selected for export. The MGMTclas parameter does not apply.
CONSOLID	Applies to consolidate. The list of eligible VTSSs for consolidation is determined by the specified VTSSLST statement. VTSS list also determines the search order for an RTD to service the MVCs selected for consolidation. The MGMTclas parameter does not apply.

HOST

optionally specifies one or more hosts to which the rule applies. If this parameter is used, any hosts not specified on this parameter ignore the rule. If the parameters is not used the statement applies to all hosts.

host-id

a host identifier (maximum 8 characters).

Note – The VTSS list specified on the VTSSLST parameter is further qualified by the criteria specified by the MGMTclas, VTSS, STORclas, and MVCpool parameters.

MGMTclas

specifies a Management Class.

mgmt-class-name

the names of a Management Class that you defined on the MGMTclas control statement.

PREVvtss

specifies a VTSS where a VTV:

- Is or resident.
- Was migrated from.

STORclas

specifies a Storage Class and applies only when MVCs are used to select VTSSs.

stor-clas-name

the name of a Storage Class that you defined on the STORclas control statement.

MVCpool

specifies a Named MVC Pool and applies only when MVCs are used to select VTSSs.

poolname

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

VTSSLST

specifies a list of VTSSs and their corresponding preferencing.

vtss-list-name

the name of a VTSS list that you defined on the VTSSLST control statement.

HSC Programmatic Interface Enhancements

The MOUNT, QDRLIST, and SELSCR requests support an additional value of VIRTUAL (to specify a VTV) for the MEDIA and RECtech parameters.

For these requests for VSM:

- For scratch requests, within a VTSS, VSM selects the first available VTD with the lowest device address.

  In a multi-VTSS system, VSM will determine which VTSS is optimal for the request, then select the first available VTD with the lowest device address in that VTSS. For a given request, VSM limits VTD selection to the first 8 VTSSs, but will select from all VTSSs for a series of requests.
- For specific requests, depending on the level of VTCS, VSM will either return a list of all VTDs or preferred VTDs.
- The SCRPOOL parameter is invalid; specify SUBPOOL instead.
- The MOUNT request supports an additional parameter of MGMTclas that can assign a VSM Management Class to the VTV.
- The volume information element returned for a QVOLUME request or a MOUNT request for a virtual volume includes a value of VIRTUAL for media type (SLXVMED) and an x'01' for volume status (SLXVSTA).

  For more information on these requests, see "Programmatic Interface" in the *HSC System Programmer's Guide for MVS*.

HSC Batch API Enhancements

The HSC Batch API supports bulk reading of CDS VTV and MVC records. For more information about the Batch API, see "Batch Application Interface (API)" of *HSC System Programmer's Guide for MVS*.

Tip – "Batch Application Interface (API)" in *HSC System Programmer's Guide for MVS* provides an example of a QCDS request that retrieves VTV records.

For more information about the HSC Batch API enhancements for VSM, see the following sections:

- "Batch API Mapping Macros" on page 269
- "SLSUREQ QCDS Request" on page 272
- "Library Element Mapping" on page 272

The HSC Batch API provides additional data in the following records returning from a Batch API Query CDS request:

- The VTV record now provides the compressed and uncompressed size of the VTV. For more information, see "SLUVTDAT Macro Record Format" on page 271.
- The MVC record now provides an MVC status indicator. For more information, see "SLUVMDAT Macro Record Format" on page 269.

Batch API Mapping Macros

The following sections described the macros to support VSM:

- "SLUVMDAT Macro Record Format" on page 269"
- "SLUVTDAT Macro Record Format" on page 271

SLUVMDAT Macro Record Format

TABLE 3-6 describes the SLUVMDAT macro record format.

TABLE 3-6 SLUVMDAT Macro Record Format

Dec	Hex	Type	Length	Label	Description
SLUVMDAT - FLAT FILE MVC DATA DSECT FUNCTION: DESCRIBES THE MVC DATA WHICH IS GENERATED TO THE FLAT FILE BY THE BATCH API					
0	(0)	STRUCTURE		MDREC	FLAT FILE RECORD
0	(0)	SIGNED-FWORD	4	MDRECRDW	RECORD DESCRIPTOR WORD
4	(4)	SIGNED-FWORD	4	MDRECL	LENGTH
8	(8)	CHARACTER	1	MDRECC	CHARACTER EBCDIC/ASCII
9	(9)	CHARACTER	1	MDRECT	TYPE M - MVC
10	(A)	CHARACTER	6	MDRECM	MVC VOLSER
16	(10)	SIGNED-FWORD	4	MDRECVC	VTV COUNT
20	(14)	SIGNED-FWORD	4	MDRECPU	PERCENT USED
24	(18)	SIGNED-FWORD	4	MDRECPA	PERCENT AVAILABLE
28	(1C)	SIGNED-FWORD	4	MDRECPW	PERCENT WASTED
32	(20)	SIGNED-FWORD	4	MDRECMC	MOUNTED COUNT
36	(24)	SIGNED-FWORD	4	MDRECTL	TIME LAST USED HIGH ORDER WORD FROM STCK INSTRUCTION
40	(28)	SIGNED-FWORD	4	MDRECMS	MEDIA SIZE
44	(2C)	LENGTH		MDRECLEN	LENGTH OF RECORD WHEN USING VERSIONS 1 AND 2 OF SLSUREQM
44	(2C)	BITSTRING	1	MDRECERR	MVC STATUS INDICATOR
		X'80'		MDINITD	MVC INITIALIZED FROM A MIGRATE
		X'40'		MDMOUNT	MVC IS MOUNTED ON AN RTD
		X'20'		MDBROKE	MVC HAS AN ERROR
		X'10'		MDFULL	MVC CANNOT CONTAIN ANY MORE VTVS
		X'08'		MDDRAIN	MVC IS BEING DRAINED
		X'04'		MDLOST	MVC IS LOST (LAST MOUNT TIMED OUT)

TABLE 3-6 SLUVMDAT Macro Record Format

Dec	Hex	Type	Length	Label	Description
		X'02'		MDDATCK	MVC SWAPPED (NOT IN RECOVERY)
		X'01'		MDREADO	MVC IS READONLY
45	(2D)	RESERVED	3		
48	(30)	LENGTH		MDRECLEN3	LENGTH OF RECORD LENGTH OF RECORD WHEN USING VERSIONS 3 OF SLSUREQM

SLUVTDAT Macro Record Format

TABLE 3-7 describes the SLUVTDAT macro record format.

TABLE 3-7 SLUVTDAT Macro Record Format

Dec	Hex	Type	Length	Label	Description
SLUVTDAT - FLAT FILE VTV DATA DSECT FUNCTION: DESCRIBES THE VTV DATA WHICH IS GENERATED TO THE FLAT FILE BY THE BATCH API					
0	(0)	STRUCTURE		VDREC	FLAT FILE RECORD
0	(0)	SIGNED-FWORD	4	VDRECRDW	RECORD DESCRIPTOR WORD
4	(4)	SIGNED-FWORD	4	VDRECL	LENGTH
8	(8)	CHARACTER	1	VDRECC	CHARACTER EBCDIC/ASCII
9	(9)	CHARACTER	1	VDRECT	TYPE V - VTV
10	(A)	CHARACTER	6	VDRECV	VTV VOLSER
16	(10)	CHARACTER	8	VDRECVT	VTSS
24	(18)	SIGNED-FWORD	4	VDRECSZ	SIZE (MB)
28	(1C)	CHARACTER	1	VDRECM	MIGRATED Y/N
29	(1D)	CHARACTER	1	VDRECD	DUPLEX Y/N
30	(1E)	CHARACTER	6	VDRECM1	MVC VOLSER OF FIRST/ONLY COPY
36	(24)	CHARACTER	6	VDRECM2	MVC VOLSER OF SECOND COPY
42	(2A)	CHARACTER	1	VDRECI	INVALID Y/N
43	(2B)	CHARACTER	1	VDRECS	SCRATCH Y/N
44	(2C)	SIGNED-FWORD	4	VDRECTC	HIGH ORDER WORD OF TOD CLOCK (GMT) CREATION DATE/TIME RETURNED BY STCK INSTRUCTION
48	(30)	SIGNED-FWORD	4	VDRECTL	HIGH ORDER WORD OF TOD CLOCK (GMT) LAST USED DATE/TIME RETURNED BY STCK INSTRUCTION
52	(34)	CHARACTER	8	VDRECMC	MANAGEMENT CLASS
60	(3C)	LENGTH		VDRECLEN	LENGTH OF RECORD WHEN USING SLSUREQM VERSIONS 1 AND 2
60	(3C)	SIGNED-FWORD	4	VDUCMPSZ	VTV UNCOMPRESSED SIZE (BYTES)
64	(40)	SIGNED-FWORD	4	VDCOMPSZ	VTV COMPRESSED SIZE (BYTES)
68	(44)	CHARACTER	6		RESERVED
74	(4A)	CHARACTER	6		RESERVED
80	(50)	LENGTH		VDRECLEN	LENGTH OF RECORD WHEN USING SLSUREQM VERSION 3

SLSUREQ QCDS Request

The following TYPE= values are valid on the SLSUREQ QCDS request:

MVC

specifies the VTCS MVC record area of the CDS.

VTV

specifies the VTCS VTV record area of the CDS.

Library Element Mapping

TABLE 3-8 describes the Library Element Record Mapping additions.

TABLE 3-8 Library Element Record Mapping Additions

Request	Records Returned
READ MVC	VTCS MVC records mapped by the SLUVMDAT macro
READ VTV	VTV records mapped by the SLUVTDAT macro

HSC Operator Command Enhancements

With HSC:

- You can dynamically reload SCRPOOL statements via the SCRPDEF command. For more information, see “Scratch Subpool Definition (SCRPDEF) Command and Control Statement” in *HSC System Programmer’s Guide for MVS*.
- The Warn SCRatch, Display SCRatch, and Display THReshld commands are enhanced to let you manage and monitor scratch VTVs. For more information, see:
  - “HSC WARN SCRATCH Command Enhancements” on page 274
  - “HSC DISPLAY THRESHLD Command Enhancements” on page 275
  - “HSC DISPLAY SCRATCH Command Enhancements” on page 276
- You can expand the CDS using the CDS EXPAND command.
- You can use the TRACELKP command to trace HSC definition data sets, including the following:
  - VOLATTR
  - LMUPDEF
  - MVCPOOL
  - MGMTCLAS
  - STORCLAS

  For more information, see “Commands, Control Statements, and Utilities,” in *HSC Operator’s Guide for MVS*.

HSC WARN SCRATCH Command Enhancements

For VSM, the enhanced HSC Warn SCRatch command sets scratch warning thresholds for HSC subpools that contain VTVs.

Syntax

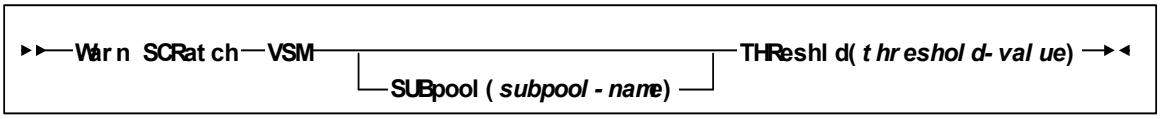


FIGURE 3-15 Warn SCRatch VSM Command Syntax

Parameters

VSM

Specifies that the command applies to VTVs only.

SUBpool

Specifies an HSC subpool that contains VTVs. If not specified, the command applies to the total VSM scratch count.

subpool-name

The specified subpool.

THReshld

Specifies the scratch warning threshold. If the number of scratches goes below the specified threshold, HSC issues a warning message. Valid values are 0 to 9999.

threshold-value

The specified threshold.

HSC DISPLAY THRESHLD Command Enhancements

For VSM, the enhanced HSC Display THReshld command displays scratch thresholds for HSC subpools that contain VTVs.

Syntax

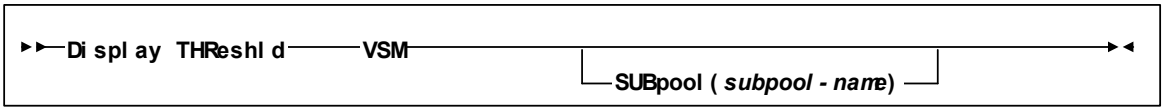


FIGURE 3-16 Display THReshld VSM Command Syntax

Parameters

VSM

Specifies that the command applies to VTVs only.

SUBpool

Specifies an HSC subpool that contains VTVs. If not specified, the command applies to the total VSM scratch count.

subpool-name

The specified subpool.

HSC DISPLAY SCRATCH Command Enhancements

For VSM, the enhanced HSC Display SCRatch command displays scratch counts for HSC subpools that contain VTVs.

Syntax

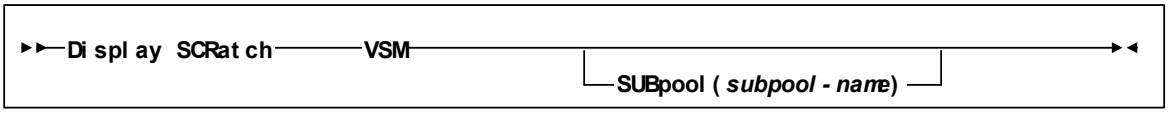


FIGURE 3-17 Warn SCRatch VSM Command Syntax

Parameters

VSM

Specifies that the command applies to VTVs only.

SUBpool

Specifies an HSC subpool that contains VTVs. If not specified, the command applies to the total VSM scratch count.

subpool-name

The specified subpool.

CHAPTER 4

MVS/CSC Support for VSM

This chapter contains reference information about the MVS/CSC features that support VSM.

Caution – If you are using MVS/CSC to request mounts of scratch VTVs from an HSC scratch subpool, if the subpool is empty, MVS/CSC will not issue a message indicating that the subpool is empty. Instead, VTCS issues a message SLS6671E and continues to retry the scratch mount until you make scratch VTVs available!

MVS/CSC Startup Parameter Enhancements

The following sections describe the MVS/CSC startup parameter enhancements for VSM. For more information about MVS/CSC startup parameters, see “Defining MVS/CSC Startup Parameters” in *MVS/CSC Configuration Guide*.

DEFER

In the JES2 environment, regardless of the value you specify, the DEFER parameter is always set to ON for VTVs. That is, for VTVs, deferred mount processing is enabled, which overrides the mount processing specified in the user’s JCL. The VTV mount is deferred until the JCL job step opens a data set on the VTV. This value helps minimize VTV recalls. If a data set resides on a migrated VTV, VSM does not recall the VTV until the job actually opens the data set on the VTV.

In the JES3 environment, regardless of the value you specify, the DEFER parameter is always set to JES3 for VTVs, which causes all mounts to be JES3 deferred. A volume is not mounted until a step begins execution.

Note that if a unit affinity chain includes a mixture of incompatible drives (including VTDs), NCS SMC device exclusion always ensures that the chain will be broken.

FETCH

FETCH specifies whether JES3 operator fetch message IAT5110 is issued during VTD allocation.

MVS/CSC DISPLAY Command Enhancements

In a VSM configuration, the DISPLAY LIBUNITS command displays VIRTUAL for VTDs in the Model column. For more information about the DISPLAY LIBUNITS command, see "Issuing MVS/CSC Operator Commands" of *MVS/CSC Operator's Guide*. Note that you can use VTCS commands and reports on another host to produce additional VSM information.

MVS/CSC User Exit Enhancements

MVS/CSC User Exit SCSUX02 (JES2 and JES3 without TAPE setup environments), which you use to control transport allocation for scratch mounts, now supports return code UX02VIRT in register 15. SCSUX04 (JES3 with TAPE setup environment) also supports return code UX04VIRT in register 15. To satisfy a scratch mount request, these return codes cause VSM to select an available VTD in your system and route the data set to a VTV mounted on that VTD. In a multi-VTSS environment, therefore, these return codes do *not* direct the VTD allocation to a specific VTSS, but let the allocation occur in any VTSS in the configuration.

Information returned from SCSUX09 (JES2 and JES3 without TAPE setup environments) and SCSUX11 (JES3 with TAPE setup environment) applies to real transports only and is ignored for VTDs. VTD mounts are automatically deferred. For more information about MVS/CSC User Exits, see *SMC User Exits Guide*.

MVS/CSC Programmatic Interface Enhancements

The SCSXREQM macro mappings are updated to support VSM as follows:

- The SCXVMED field can now display a value of VIRTUAL for VTVs.
- In the Volume Information Element, the formerly reserved field at decimal offset 24 is now an 8 byte character field with label SCXVTSSN. If SCXVMED is VIRTUAL, the volume is VTSS-resident, and MVS/ CSC controls the VTD in the VTSS in which the VTV resides, SCXVTSSN displays the VTSS name. If the VTV is migrated, SCXVTSSN is blank.
- The field SCXVLC is hexadecimal zero for a VTV.

For more information about the SCSXREQM macro, see "SCSXREQM Macro Mappings" of *MVS/CSC System Programmer's Guide*.

MVS/CSC DELDISP Parameter Enhancements

MVS/CSC provides two new settings for the DELDISP startup parameter which is specified in a sequential file (usually a PDS member) at initialization:

ASCRTCH

(All scratch). Both real tape volumes and VTVs are made scratch if they were mounted scratch and the disposition on the dismount message is delete ('D').

VSCRTCH

(Virtual scratch). Only VTVs are made scratch if they were mounted scratch and the delete disposition on the dismount message is delete ('D').

The current DELDISP settings (SCRTCH and NOSCRTCH) define scratch handling at dismount **only** for real volumes. In an MVS/CSC system, if DELDISP is set to either of these values, VTVs are **never** scratched at dismount.

Each MVS/CSC system can define its own startup parameter file and can have different settings for DELDISP. A recycle of an MVS/CSC system is not necessary to change the DELDISP setting. The MVS/CSC ALTER command can change the setting for DELDISP; it accepts the two new settings. When changing the DELDISP setting via the ALTER command, it goes into affect immediately for that MVS/CSC system. However, if the MVS/CSC is recycled, the DELDISP setting is set to the value defined in the startup parameter file; if omitted it defaults to NOSCRTCH.

CHAPTER **5**

LibraryStation Support for VSM

This chapter contains reference information about the LibraryStation features that support VSM.

SPNUM Statement

The enhanced SPNUM statement, which defines a LibraryStation subpool that corresponds to an HSC subpool, lets MVS/CSC and non-MVS/CSC clients request VTV mounts and pass a Management Class to VSM.

Syntax

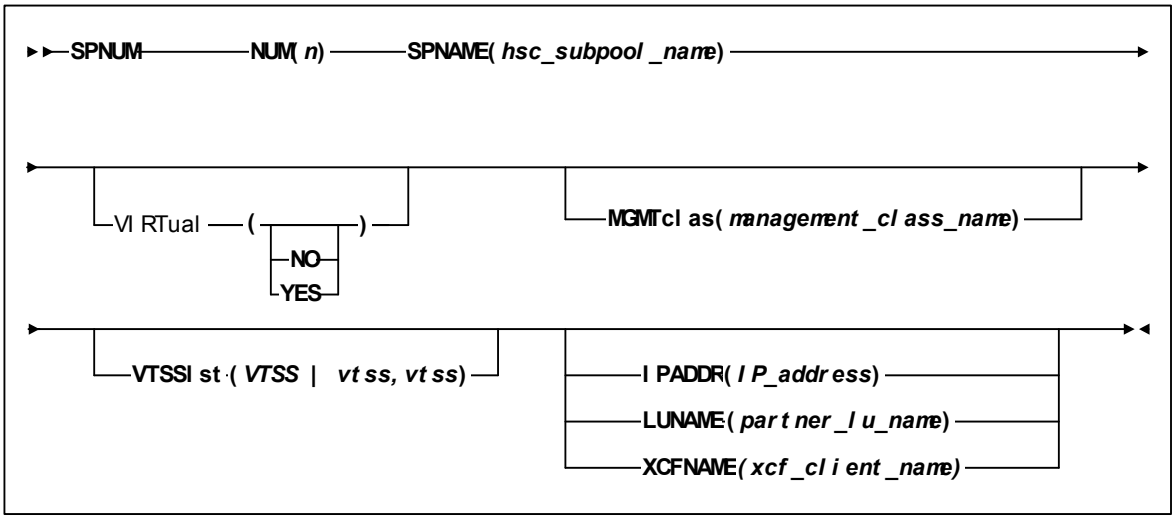


FIGURE 5-1 SPNUM Statement Syntax

Parameters

Unchanged SPNUM Parameters

The following SPNUM parameters are unchanged but apply to VSM. FIGURE 5-1 shows valid values for these parameters; see "Configuring the LSDEF Data Set" in *LibraryStation Configuration Guide* for more information.

- NUM
- SPNAME
- IPADDR
- LUNAME
- XCFNAME

SPNUM Parameters for VSM

The following new SPNUM parameters apply to LibraryStation in VSM environments. These parameters allow MVS/CSC and non-MVS/CSC clients to request VTV mounts.

VIRTual

specifies whether the subpool contains VTVs.

NO

the subpool does not contain VTVs.

YES

the subpool contains VTVs.

Note – If VIRTual is not specified, the subpool can contain both real volumes and VTVs.

MGMTclas

specifies the name of a Management Class you defined on the HSC MGMTclas control statement.

mgmt-class-name

the Management Class name.

Caution – For MVS/CSC clients, if you specify a Management Class when routing data to VSM (on a TAPEREQ statement, for example), StorageTek recommends that you do **not** specify a Management Class on the SPNUM statement.

VTSS

specifies up to 16 VTSSs used to satisfy the mount request.

vtssnam1...vtssname16

the names of up to 16 VTSSs

Each VTSS name must correspond to a VTSS name specified on a VIRTACS statement.

VIRTACS Statement

The VIRTACS statement defines a virtual ACS that maps to a VTSS to let clients connect to VSM.

Syntax

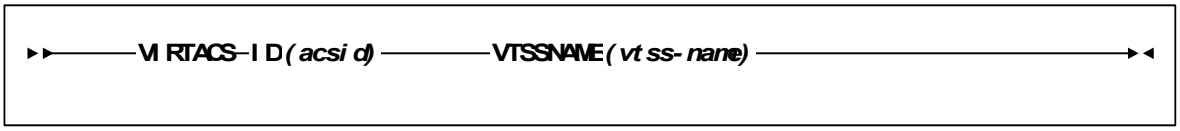


FIGURE 5-2 VIRTACS Statement Syntax

Parameters

ID

specifies a virtual ACS ID.

acsid

a decimal virtual ACS ID.

VTSSNAME

specifies the VTSS name that maps to the virtual ACS ID.

vtss-name

a VTSS name.

SLGDIAG VIRTUAL_DRIVE Parameter

The SLGDIAG utility now provides VIRTUAL_DRIVE parameter that verifies LibraryStation operation with VSM in the following format (all decimal numbers):

=VIRTUAL_DRIVE=*ascid,lsmid,panelid,driveid*

See *Installing ELS* for more information on VTD drive addresses for LibraryStation and NCS clients.

You can use SLGDIAG to verify LibraryStation operation with VSM in either of the following ways:

- To verify LibraryStation operation with only VSM (but not with HSC), specify the = VIRTUAL_DRIVE= parameter to query the specified VTD.
- To verify LibraryStation operation with VSM and HSC in the same batch job, specify the = VIRTUAL_DRIVE= parameter and also the existing =DRIVE= and =V0LUME= parameters (which request a mount/dismount on the specified Nearline transport).

For more information on the SLGDIAG utility, see “Administration and Maintenance” in *LibraryStation Operator and System Programmer’s Guide*.

APPENDIX **A**

VTCS SMF Record Formats

This appendix describes the formats of the HSC SMF record subtypes for VTCS events.

Note – In the record descriptions in this appendix, all generated timestamps, regardless of whether they are ttime or TOD values, are based on GMT time, not local time.

SLSSMF10 - VTCS SMF Subtype 10 Record

Function

Records a VTSS subsystem performance request.

TABLE A-1 SLSSMF10 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF10	VTCS SMF record subtype 10
0	0	character	8	SMF10VTS	VTSS ID
8	8	hexstring	2	SMF10BCH	base cache size (MB), where base cache is system space reserved for VTSS processing
10	A	hexstring	2	SMF10CCH	customer cache size (MB)
12	C	hexstring	4	SMF10OCH	offline cache size
16	10	hexstring	4	SMF10PCH	pinned cache size
20	14	hexstring	2	SMF10NSZ	nvs size (MB)
22	16	hexstring	8	SMF10TCT	reserved
30	1E	hexstring	8	SMF10TCP	total back end capacity
38	26	hexstring	8	SMF10FCT	reserved
46	2E	hexstring	8	SMF10FCP	total free back end capacity
54	36	hexstring	8	SMF10CFT	reserved
62	3E	hexstring	8	SMF10CFP	collected free back end capacity
70	46	hexstring	8	SMF10BRT	reserved

TABLE A-1 SLSSMF10 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
78	4E	hexstring	8	SMF10BRP	bytes read for free space collection
86	56	hexstring	8	SMF10SCT	reserved
94	5E	hexstring	8	SMF10SCP	total amount of free space collection
102	66	hexstring	2	SMF10RGC	redundancy group count
104	68	hexstring	8	SMF10CDT	reserved
112	70	hexstring	8	SMF10CDP	standard capacity defined
120	78	hexstring	4	SMF10EMP	count of ECAM-T messages processed
124	7C	hexstring	4	SMF10EBS	count of ECAM-T messages bypassed because no buffer space available
128	80	hexstring	4	SMF10EBC	count of ECAM-T messages bypassed because configuration was busy
132	84	hexstring	4	SMF10ECP	number of ECAM-T channel programs

SLSSMF11 - VTCS SMF Subtype 11 Record

Function

Records a VTSS channel interface performance request.

TABLE A-2 SLSSMF11 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF11	VTCS SMF record subtype 11
0	0	character	8	SMF11VTS	VTSS ID
8	8	hexstring	2	SMF11CNT	count of entries in this record the following fields repeat for each interface in this record
10	A	data		SMF11ENT	start of entry
10	A	character	8	SMF11INM	channel interface name
18	12	bitstring	2	SMF11CI	channel interface installed (y/n)
		X'0000'		SMF11CIN	no
		X'0001'		SMF11CIY	yes
20	14	bitstring	2	SMF11CE	channel interface enabled (y/n)
		X'0000'		SMF11CEN	no
		X'0001'		SMF11CEY	yes
22	16	hexstring	2	SMF11NAT	number of addresses trapped

TABLE A-2 SLSSMF11 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
24	18	hexstring	2	SMF11CSP	Channel type
			X'0200'	SMF11CSE	ESCON channel
			X'1000'	SMF11CSI	IP link
			X'2000'	SMF11CSF	FICON channel
26	1A	hexstring	8	SMF11NIO	number of I/Os
34	22	hexstring	8	SMF11CUB	control unit busy (in υ -seconds)
42	2A	bitstring	2	SMF11TOL	type of link
		X’0000’		SMF11TLH	host
		X’0001’		SMF11TLR	RTD
44	2C	length		SMF11ENL	length of each entry

SLSSMF13 - VTCS SMF Subtype 13 Record

Function

Records a VTV mount request.

TABLE A-3 SLSSMF13 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF13	VTCS SMF record subtype 13
0	0	character	8	SMF13VTS	VTSS ID
8	8	character	6	SMF13VID	VTV volser ID
14	E	character	2	SMF13DID	VTD device ID
16	10	bitstring	2	SMF13RWS	read/write state (thumbwheel)
		X'0001'		SMF13RRO	read only
		X'0002'		SMF13RRW	read/write
18	12	bitstring	2	SMF13VMT	virtual mount type
		X'0001'		SMF13EXT	mount existing VTV
		X'0002'		SMF13SSL	mount sl scratch VTV
		X'0003'		SMF13SNL	mount existing VTV as scratch
		X'0004'		SMF13SAL	mount ANSI label scratch VTV
20	14	hexstring	4	SMF13TIM	VTV timestamp (ttime format, seconds since 1/1/70)
24	18	bitstring	2	SMF13RCI	recall indicator
		X'0001'		SMF13MNR	mounted without a recall
		X'0002'		SMF13MRC	mounted after a recall
26	1A	bitstring	2	SMF13CTP	cartridge type
		X'0000'		SMF13SCT	S-cart (max 400MB size)
		X'0001'		SMF13ECT	E-cart (max 800MB size)
		X'0002'		SMF132GB	2000MB
		X'0003'		SMF134GB	4000MB

TABLE A-3 SLSSMF13 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
28	1C	character	10		reserved
38	26	character	8	SMF13JNM	MVS jobname
46	2E	character	8	SMF13SNM	MVS stepname
54	36	character	44	SMF13DSN	MVS data set name
98	62	hexstring	8	SMF13MST	mount start timestamp (TOD), where mount start occurs when VTCS receives a mount request from HSC (or VTCS generates the request), generates a new thread to handle the mount request, then determines whether the request is for an existing, new, or scratch VTV
106	6A	hexstring	8	SMF13MET	mount end timestamp (TOD), where mount end occurs when VTSS generates a successful response to the ECAM-T request to mount the VTV on the selected RTD
114	72	character	8	SMF13MGT	VTV Management Class
122	7A	character	8	SMF13HST	Originating host name

SLSSMF14 - VTCS SMF Subtype 14 Record

Function

Records a VTV dismount request.

TABLE A-4 SLSSMF14 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF14	VTCS SMF record subtype 14
0	0	character	8	SMF14VTS	VTSS ID
8	8	character	6	SMF14VID	VTV volser ID
14	E	bitstring	2	SMF14STA	VTV state
		X’0001’		SMF14MNT	VTV mounted
		X’0002’		SMF14DSM	VTV dismounted
		X’0003’		SMF14NON	VTV does not exist
		X’0004’		SMF14MIG	VTV is being migrated
		X’0005’		SMF14REC	VTV is being recalled
		X’0006’		SMF14VTM	VTV logically dismounted by VTVMAINT
16	10	hexstring	2	SMF14DID	MVS device address
18	12	hexstring	4	SMF14VSZ	Uncompressed VTV size - number of 4k units
22	16	hexstring	4	SMF14MSZ	the number of virtual tape pages in 32K increments required to migrate the VTV to an RTD
26	1A	hexstring	4	SMF14TIM	the last time the VTV was successfully mounted on a VTD (ttime format, seconds since 1/1/70)
30	1E	hexstring	2	SMF14UL#	number of MVCs to unlink
32	20	bitstring	2	SMF14CTP	cartridge type
		X’0000’		SMF14SCT	S-cart (max 400MB size)
		X’0001’		SMF14ECT	E-cart (max 800MB size)
		X’0002’		SMF142GB	2000MB

TABLE A-4 SLSSMF14 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
		X’0003’		SMF144GB	4000MB
34	22	bitstring	2	SMF14VMT	virtual mount type
		X’0001’		SMF14EXT	mount existing VTV
		X’0002’		SMF14SSL	mount sl scratch VTV
36	24	character	8	SMF14JNM	MVS jobname
44	2C	character	8	SMF14SNM	MVS stepname
52	34	character	44	SMF14DSN	MVS data set name
96	60	character	8	SMF14MGT	VTV Management Class
104	68	character	8	SMF14HST	Originating host name
112	70	hexstring	4	SMF14N4K	UNCOMP SIZE - NO OF 4K UNITS
116	74	bitstring	2	SMF14SRS	Synchronous Replication Status
		X’0000’		SMF14SRN	SYNCH REP NOT REQUESTED
		X’0001’		SMF14SRC	SYNCH REP COMPLETE
		X’0002’		SMF14SRF	SYNCH REP FAILED
118	76	hexstring	8	SMF14RUN	Rewind/Unload received time (TOD format)

SLSSMF15 - VTCS SMF Subtype 15 Record

Function

Records a delete VTV request.

TABLE A-5 SLSSMF15 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF15	VTCS SMF record subtype 15
0	0	character	8	SMF15VTS	VTSS ID (blanks if migrated)
8	8	character	6	SMF15VID	virtual volser ID
14	E	character	4	SMF15TIM	VTV creation time (ttime format, seconds since 1/1/70)
18	12		4		reserved
22	16	character	4	SMF15LTR	time VTV last referenced (high order TOD value)
26	1A	bitstring	2	SMF15CTP	cartridge type
		X'0000'		SMF15SCT	S-cart (max 400MB size)
		X'0001'		SMF15ECT	E-cart (max 800MB size)
		X'0002'		SMF152GB	2000MB
		X'0003'		SMF154GB	4000MB
28	1C		2		reserved
30	1E	bitstring	2	SMF15RSN	VTV delete reason code
		X'0001'		SMF15NMM	VTV migrated then deleted
		X'0002'		SMF15MPR	VTV previously migrated
		X'0003'		SMF15SPR	VTV reclaimed
		X'0004'		SMF15CON	VTV consolidated
		X'0005'		SMF15OLD	invalid VTV version found
		X'0006'		SMF15DSC	VTV deleted on scratch
		X'0007'		SMF15IMP	VTV deleted by import
		X'0008'		SMF15DSU	VTV deleted by DELETSCR utility
		X'0009'		SMF15FRC	ensure VTV is not resident after failed recall
32	20	character	8	SMF15MGT	VTV Management Class

TABLE A-5 SLSSMF15 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
40	28	bitstring	2	SMF15LRI	last residency indicator
		X’0001’		SMF15LRR	last residency was for recall
		X’0002’		SMF15LRC	last residency was for create

SLSSMF16 - VTCS SMF Subtype 16 Record

Function

Records an RTD mount request.

TABLE A-6 SLSSMF16 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SMF16VTS	VTCS SMF record subtype 16
0	0	character	8	SMF16VTS	VTSS ID
8	8	hexstring	2	SMF16RID	RTD ID (0-1F)
10	A	character	6	SMF16MID	MVC volser ID
16	10	character	6	SMF16AID	actual volser from VOL1 label
22	16	bitstring	2	SMF16RWS	read/write state (thumbwheel)
		X’0001’		SMF16RRO	read only state
		X’0002’		SMF16RRW	read/write state
24	18	bitstring	2	SMF16MT	mount request type
		X’0001’		SMF16MTM	migrate
		X’0002’		SMF16MTR	recall
		X’0003’		SMF16MTL	reclaim
		X’0004’		SMF16MTD	drain
		X’0005’		SMF16MTA	audit
		X’0006’		SMF16MTC	consolidate
		X’0007’		SMF16MTX	export
		X’0008’		SMF16MTI	INVENTRY utility
26	1A	hexstring	32	SMF16SNS	RTD sense data (all zeros or all X’FF’s unless RTD errors occur)
58	3A	hexstring	8	SMF16MST	mount start timestamp (TOD), where mount start occurs when HSC receives a successful request to load the requested MVC
66	42	hexstring	8	SMF16MET	mount end timestamp (TOD), where mount end occurs when the VTSS receives a successful ECAM-T request to mount the requested MVC on an RTD

TABLE A-6 SLSSMF16 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
74	4A	character	8	SMF16SCL	MVC Storage Class (will contain hex zeros if theMVC is empty)
82	52	character	2	SMF16INF	RTD Channel Interface ID
84	54	character	4	SMF16ADR	MVS address of RTD
88	58	hexstring	2	SMF16LOC	Location name/ACS:LSM of the RTD
96	60	character	4	SMF16TYP	Device type name of the RTD

SLSSMF17 - VTCS SMF Subtype 17 Record

Function

Records an RTD dismount request.

TABLE A-7 SLSSMF17 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF17	VTCS SMF record subtype 17
0	0	character	8	SMF17VTS	VTSS ID
8	8	hexstring	2	SMF17RID	RTD ID (0-1F)
10	A	hexstring	64	SMF17BLD	RTD buffered log data
74	4A	hexstring	32	SMF17SNS	RTD sense data (all zeros or all X'FF's unless RTD errors occur)
106	6A	character	8	SMF17SCL	unused
114	72	character	6	SMF17MVC	MVC volser
120	78	hexstring	2	SMF17INF	RTD Channel Interface ID
122	7A	character	4	SMF17ADR	MVS address of RTD
126	7E	hexstring	1	SMF17DFL	Dismount flag
		X'80'		SMF17DRX	Retention period reduced
		X'40'		SMF17DAR	Auto recall
		X'20'		SMF17DMF	Full MVC
		X'10'		SMF17DVY	RTD vary command
		X'08'		SMF17DSW	MVC required on another drive or RTD required for another volume
		X'01'		SMF17DRT	Retain period applied

SLSSMF18 - VTCS SMF Subtype 18 Record

Function

Records a migrate VTV request.

TABLE A-8 SLSSMF18 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF18	VTCS SMF record subtype 18
0	0	character	8	SMF18VTS	VTSS ID
8	8	hexstring	2	SMF18RID	RTD ID (0-1F)
10	A	character	6	SMF18VID	VTV volser ID
16	10	character	6	SMF18MID	MVC volser ID
22	16	hexstring	4	SMF18VPO	VTV position on this MVC (block ID)
26	1A	character	6	SMF18AID	actual volser from VOL1 label
32	20	hexstring	4	SMF18MSZ	uncompressed size of the VTV in bytes
36	24	hexstring	4	SMF18BCM	the number of virtual tape pages in 32K increments required to migrate the VTV to an RTD
40	28	hexstring	4	SMF18TIM	the last time the VTV was successfully mounted on a VTD (ttime format, seconds since 1/1/70)
44	2C	bitstring	2	SMF18MT	migrate request type
		X’0001’		SMF18MTA	auto
		X’0002’		SMF18MTI	immediate
		X’0003’		SMF18MTD	demand
		X’0004’		SMF18MTR	reclaim
		X’0005’		SMF18MTC	consolidate
		X’0006’		SMF18MTX	export
46	2E	bitstring	2	SMF18CTP	cartridge type
		X’0000’		SMF18SCT	S-cart (max 400MB size)
		X’0001’		SMF18ECT	E-cart (max 800MB size)
		X’0002’		SMF182GB	2000MB

TABLE A-8 SLSSMF18 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
		X'0003'		SMF184GB	4000MB
48	30	hexstring	4	SMF18NPO	next MVC position (block ID)
52	34	hexstring	32	SMF18SNS	RTD sense
84	54	hexstring	8	SMF18MST	migrate start timestamp (TOD)
92	5C	hexstring	8	SMF18MET	migrate end timestamp (TOD)
100	64	character	8	SMF18MGT	VTV Management Class
108	6C	character	8	SMF18SCL	MVC Storage Class
116	74	character	2	SMF18INF	RTD Channel Interface ID
118	78	character	4	SMF18ADR	MVS address of RTD

SLSSMF19 - VTCS SMF Subtype 19 Record

Function

Records a recall VTV request.

TABLE A-9 SLSSMF19 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SMF19VTS	VTCS SMF record subtype 19
0	0	character	8	SMF19VTS	VTSS ID
8	8	hexstring	2	SMF19RTD	RTD ID (0-1F)
10	A	character	6	SMF19VID	VTV volser ID
16	10	character	6	SMF19MID	MVC volser ID
22	16	hexstring	4	SMF19VPO	VTV position on this MVC (block ID)
26	1A	bitstring	2	SMF19RE	recall with error
		X'0000'		SMF19REN	no
		X'0001'		SMF19REY	yes
28	1C	character	6	SMF19AID	actual volser from VOL1 label
34	22	hexstring	4	SMF19MSZ	VTV media size
38	26	hexstring	4	SMF19BCM	number of bytes currently recalled
42	2A	hexstring	4	SMF19TIM	the last time the VTV was successfully mounted on a VTD (ttime format, seconds since 1/1/70)
46	2E	bitstring	2	SMF19RT	recall request type
		X'0001'		SMF19RTA	auto
		X'0002'		SMF19RTN	drain
		X'0003'		SMF19RTD	demand
		X'0004'		SMF19RTR	reclaim
		X'0005'		SMF19RTC	consolidate
		X'0006'		SMF19RTX	export
48	30	bitstring	2	SMF19CTP	cartridge type
		X'0000'		SMF19SCT	S-cart (max 400MB size)
		X'0001'		SMF19ECT	E-cart (max 800MB size)
		X'0002'		SMF192GB	2000MB

TABLE A-9 SLSSMF19 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
		X'0003'		SMF194GB	4000MB
50	32	hexstring	32	SMF19SNS	RTD sense
82	52	hexstring	8	SMF19RST	recall start timestamp (TOD)
90	5A	hexstring	8	SMF19RET	recall end timestamp (TOD)
98	62	character	8	SMF19MGT	VTV Management Class
106	6A	character	8	SMF19SCL	MVC Storage Class
114	72	character	2	SMF19INF	RTD Channel Interface ID
116	74	character	4	SMF19ADR	MVS address of RTD

SLSSMF20 - VTCS SMF Subtype 20 Record

Function

Records an RTD performance request.

TABLE A-10 SLSSMF20 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF20	VTCS SMF record subtype 20
0	0	character	8	SMF20VTS	VTSS ID
8	8	hexstring	2	SMF20CNT	count of entries in this record the following fields repeat for each RTD in this record
10	A	area		SMF20ENT	start of entry
10	A	character	8	SMF20RNM	RTD name
18	12	bitstring	2	SMF20ST	RTD state
		X'0000'		SMF20STU	unconfigured
		X'0001'		SMF20STC	configured
20	14	hexstring	8	SMF20ATM	device available time (υ -seconds), which is the time the MVC is mounted on the RTD
28	1C	hexstring	8	SMF20ACT	device activity (initial selects)
36	24	hexstring	8	SMF20BTR	bytes transferred - read
44	2C	hexstring	8	SMF20BTW	bytes transferred - write
52	34	hexstring	8	SMF20DUT	device utilization time (υ -seconds), which is the accumulated time of each CCW chain to device end
60	3C	hexstring	8	SMF20DCT	device connect time (υ -seconds), which is the accumulated time of each CCW chain to device end

SLSSMF21 - VTCS SMF Subtype 21 Record

Function

Records a vary RTD.

TABLE A-11 SLSSMF21 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF21	VTCS SMF record subtype 21
0	0	character	8	SMF21VTS	VTSS ID
8	8	hexstring	2	SMF21RTD	RTD ID (0-1F)
10	A	bitstring	2	SMF21STA	new device state
		X’0001’		SMF21OFF	offline
		X’0002’		SMF21ON	online
		X’0003’		SMF21MAI	maintenance

SLSSMF25 - VTCS SMF Subtype 25 Record

Function

Records MVC status.

TABLE A-12 SLSSMF25 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF25	VTCS SMF record subtype 25
0	0	character	6	SMF25MID	MVC volser ID
6	6	hexstring	4	SMF25TFS	total free space (includes any space for invalid VTVs)
10	A	hexstring	4	SMF25UFS	usable free space (after the last valid VTV on the MVC)
14	E	hexstring	4	SMF25NAV	number of active VTVS
18	12	character	8	SMF25SCL	MVC Storage Class
26	1A	hexstring	4	SMF25TUS	space in Kb used by current VTVs
30	1E	hexstring	4	SMF25NDV	number of "holes" (deleted VTVs)
34	22	hexstring	4	SMF25LUT	top 4 bytes of the TOD clock when the MVC was last used
38	26	hexstring	4	SMF25LWT	top 4 bytes of the TOD clock when the MVC was last updated

SLSSMF26 - VTCS SMF Subtype 26 Record

Function

Records VTV movement.

TABLE A-13 SLSSMF26 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF26	VTCS SMF record subtype 26
0	0	character	8	SMF26VTS	VTSS ID
8	8	character	6	SMF26VID	VTV volser ID
14	E	character	6	SMF26OMI	old MVC volser ID
20	14	character	6	SMF26NMI	new MVC volser ID
26	1A	hexstring	4	SMF26VPO	VTV position on new MVC (block ID)
30	1E	hexstring	8	SMF26MST	move start timestamp (TOD)
38	26	hexstring	8	SMF26MET	move end timestamp (TOD)
46	2E	character	8	SMF26MGT	VTV Management Class

SLSSMF27 - VTCS SMF Subtype 27 Record

Function

Records VTV scratch status.

TABLE A-14 SLSSMF27 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF27	VTCS SMF record subtype 27
0	0	character	6	SMF27VID	VTV volser ID
6	6	character	8	SMF27MCL	VTV Management Class
14	E	bitstring	2	SMF27STP	VTV scratch type
		X’0001’		SMF27STN	no delete on scratch
		X’0002’		SMF27STD	delete on scratch
16	10	hexstring	4	SMF27MSZ	VTV media size
20	14	hexstring	4	SMF27TIM	the last time the VTV was updated (ttime format, seconds since 1/1/70)
24	18	hexstring	4	SMF27LUS	the last time the VTV was used (TOD format)
28	1C	hexstring	6	SMF27MV1	volser of MVC 1 that contains the VTV
34	22	hexstring	6	SMF27MV2	volser of MVC 2 that contains the VTV
40	28	character	8	SMF27VTS	VTSS name
48	30	bitstring	1	SMF27RES	VTV last resident indicator
			X’80’	SMF27RVT	resident on VTSS
			X’40’	SMF27RM1	resident on MVC1
			X’20’	SMF27RM2	resident on MVC2
			X’10’	SMF27RM3	resident on MVC3
			X’08’	SMF27RM4	resident on MVC4
49	31		1	SMF27SPR	reserved
50	32	bitstring	2	SMF27CTP	cartridge type
		X’0000’		SMF27SCT	S-cart (max 400MB size)
		X’0001’		SMF27ECT	E-cart (max 800MB size)
		X’0002’		SMF2732GB	2000MB
		X’0003’		SMF274GB	4000MB

TABLE A-14 SLSSMF27 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
52	34		4	SMF27SP2	reserved
56	38	character	6	SMF27MV3	volser of MVC3 that contains the VTV
62	3E	character	6	SMF27MV4	volser of MVC4 that contains the VTV

SLSSMF28 - VTCS SMF Subtype 28 Record

Function

Records a VTV replication.

TABLE A-15 SLSSMF28 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF28	VTCS SMF Record sub-type 28
0	0	character	8	SMF28VTS	Primary VTSSname
8	8	character	8	SMF28SVT	Secondary VTSSname
16	10	character	8	SMF28CLN	Cluster Name
24	18	character	6	SMF28VID	VTV Volser
30	1E	hexstring	2	SMF28AID	CLINK CHANIF
32	20	hexstring	1	SMF28DID	CLINK device-id
Note: SMF28AID and SMF28DID will contain binary zeros if the replication is synchronous.					
33	21	hexstring	1		reserved
34	22	bitstring	2	SMF28CTP	cartridge type
		X’0000’		SMF28SCT	S-cart (max 400MB size)
		X’0001’		SMF28ECT	E-cart (max 800MB size)
		X’0002’		SMF272GB	2000MB
		X’0003’		SMF274GB	4000MB
36	24	bitstring	1	SMF28FLG	SMF28FLAGBYTE
		X’0000’		SMF28SCT	S-cart (max 400 Mb size)

TABLE A-15 SLSSMF28 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
		X'0001'		SMF28ECT	E-cart (max 800 Mb size)
40	28	hexstring	4	SMF28BCR	Bytes replicated for VTV
44	2C	hexstring	4	SMF28TIM	VTV last updated timestamp (seconds since 1/1/70)
48	30	hexstring	32	SMF28SNS	Sense data from CLINK
80	50	hexstring	8	SMF28RST	Replicate Start Time (TOD format)
88	58	hexstring	8	SMF28RET	Replicate End Time (TOD format)
96	60	hexstring	8	SMF28MGT	VTV Management Class
104	68	hexstring	8	SMF28RUN	Rewind/Unload received time (TOD format)

SLSSMF29 - VTCS SMF Subtype 29 Record

Function

Records a VTV and MVC unlink event.

TABLE A-16 SLSSMF29 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF29	VTCS SMF Record sub-type 29
0	0	character	6	SMF29VID	VTV volser
6	6	character	6	SMF29MVC	MVC volser
12	C	character	2	SMF29MV#	number of remaining MVCs
14	E	bitstring	2	SMF29RSN	reason for unlink
		X’0001’		SMF29NLC	VTV no longer current (dismount)
		X’0002’		SMF29DRN	MVC drain/reclaim
		X’0003’		SMF29DOS	delete on scratch
		X’0004’		SMF29IMP	VTV import
		X’0005’		SMF29VMN	VTVMAINT utility
		X’0006’		SMF29MVC	MVC inventory
		X’0007’		SMF29VTS	VTSS inventory
		X’0008’		SMF29VAD	VTV audit
		X’0009’		SMF15DSU	DELETSCR utility
16	10		6		reserved

SLSSMF30 - VTCS SMF Subtype 30 Record

Function

Records a Vary Clink event.

TABLE A-17 SLSSMF30 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF30	VTCS SMF Record sub-type 30
0	0	character	8	SMF30CLU	Cluster name
8	8	character	8	SMF30VTS	Primary or Sending VTSS name
16	10	character	8	SMF30PAR	Partner VTSS name
24	18	bitstring	2	SMF30CLI	Clink Id
26	1A	bitstring	2	SMF30STA	New Clink status
		X’0001’		SMF30OFF	Offline
		X’0002’		SMF30ON	Online
		X’0004’		SMF30ONA	Online/AUTO (SYNC REPLICATE)

SLSSMF31 - SMF Subtype 31 Record

Function

Records information about each dynamically added or deleted transport.

TABLE 5-1 SLSSMF31 Record Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0	start of record		SLSSMF31	SMF Record sub-type 31
0	0	character	6	SMF31DRV	"DRIVE" character string
6	6	character	12	SMF31DYN	"DYNAMICALLY" character string
18	12	character	8	SMF31TYP	Type of drive modification: n ADDED - Drive has been dynamically added n DELETED - Drive has been dynamically deleted
26	1A	character	6	SMF31CH1	"UNIT:" character string
32	20	character	4	SMF31UNT	transport unit address
36	24	character	10	SMF31CH2	"LOCATION:" character string
46	2E	character	11	SMF31LOC	transport location
57	39	character	7	SMF31CH3	"MODEL:" character string
64	40	character	8	SMF31MOD	HSC model type

APPENDIX **B**

NCS/VTCS Alphabetic Volsers

NCS/VTCS supports alphabetic volser ranges for all commands and utilities. The rules for alphabetic volser ranges are as follows:

1. An alphabetic volser range consists of a pair of volsers (start volser and end volser) containing an incrementing alphabetic portion of 1 to 6 characters. For example:, 00000A-00000Z, ABCAAA-ABCZZZ, 9AA000-9CC000, A00A00-A00M00.

   a. A volser is composed of sequence of one to six numerics, (upper case) alphabetic or national characters (#, @ and the primary national currency symbol).

   b. A volser of less than six characters is left justified and blank padded. Each volser element in a range must have the same number of characters specified. For example, if the first volser element is 4 characters, the second must be exactly 4 characters.

2. The start and end volsers forming a volser range consists of the following sub-elements: an optional prefix, an incremental portion, and an optional suffix. Table 35. on page 256 shows examples of alphabetic volser ranges.

   a. The optional prefix consists of identical leading characters (if any) in the start and end volsers.

   b. The incremental portion starts at the first non-identical leading character in the start and end volsers forming a range. The incremental portion is either:

      - All numeric (contains characters 0 through 9 only).
      - All alphabetic (contains character A through Z only).

      The incremental portion of a volser range, therefore, terminates where a change of character type (numeric -> alphabetic or alphabetic -> numeric) is detected.

The incremental type is derived from the character type of the first character in the incremental part (numeric/alphabetic). TABLE B-1 shows example incremental ranges.

TABLE B-1 Example Incremental Ranges

volser Range	Incremental Portion	Data Type
00000A-00000Z	A-Z	Character
ABCAAA-ABCZZZ	AAA-ZZZ	Character
9AAZ00-9CCZ00	AAZ-CCZ	Character
A00B00-A99B00	00-99	Numeric
A00A00-A00M00	A-M	Character
A00B00-A00B99	00-99	Numeric

Note the following rules for incremental ranges:

- The expansion of an alphabetic incremental part is derived from a collating sequence of A-Z (it will not include the national character set).
- The data types of the incremental portions in the start and end volsers must be identical.
- The position of the incremental portion of the start volser must match that of the end volser.
- The length of the incremental portion of the start and end volsers must be identical.
- The incremental portion of the end volser must be greater than or equal to the start volser.

c. The optional suffix consists of the trailing characters from the end of the incremental portion onwards. TABLE B-2 shows an example range suffix.

TABLE B-2 Example Range Suffix

Volser Range	Incremental Portion	Suffix
A00B00-A00B99	00-99	none
A00B@0-A00D@0	B-D	@0
9AAZ00-9CCZ00	AAZ-CCZ	00 (not Z00)
900A@A-950A@A	900-950	A@A
ABCAAA-ABCZZZ	AAA-ZZZ	none

For a range to be valid the suffix of the start and end volsers forming the range must be identical.

3. The number of volumes generated from an alphabetic volser range is dependent on the number of elements in the incremental portion of the volser elements. For an A to Z range in each character position, the number of volumes can be calculated by 26 to the power of the number of positions that are being incremented as shown in TABLE B-3.

TABLE B-3 Size of Alphabetic Volser Ranges

Range	Calculation	Number of Volumes
A-Z	26^1	26
AA-ZZ	26^2	676
AAA-ZZZ	26^3	17,576
AAAA-ZZZZ	26^4	456,976

4. Per TABLE B-3, it is possible to define 26^4 VTVs in a single range. **Note, however, that** the more VTVs you define, the bigger your CDS has to be.

Alphabetic Volser Examples

TABLE B-4 and TABLE B-5 on page 321 describe valid and invalid alphabetic ranges.

TABLE B-4 Valid Alphabetic Ranges

Range	Subcomponents			Number of VTVs
	Prefix	Incremental Portion	Suffix	
AAA000-AAZ000	AA	A-Z	000	26
A00A00-A00A99	A00A	00-99		100
0AAAA0-0ZZZZ0	0	AAAA-ZZZZ	0	456,976
A00A00-A99A00	A	00-99	A00	100
99AA##-99ZZ##	99	AA-ZZ	##	676
A9A000-A9Z000	A9	A-Z	000	26
######-######	######			1
AA00##-ZZ00##		AA-ZZ	00##	676
AA00##-AA99##	AA	00-99	##	100
PROD00-PROD99	PROD	00-99		100
PROD00-PROZ00	PRO	D-Z	00	23
A4Z#@0-A9Z#@0	A	4-9	Z#@0	6
A4Z#@0-Z4Z#@0		A-Z	4Z#@0	26
A4Z#@0-A4Z#@6	A4Z#@	0-6		7
AAAAAA-AAACCC	AAA	AAA-CCC		1407
A3BZZ9-A3CDE9	A3	BZZ-CDE	9	84
999AM8-999CM8	999	AM-CM	8	53
111AAA-111ZZZ	111	AAA-ZZZ		17576

TABLE B-5 Invalid Alphabetic Ranges

Range	Subcomponents			Number of VTVs	Comments
	Prefix	Incremental Portion	Suffix		
0AAAAA-0BAAAA	0	AAAAA-BAAAA		456,977	Greater than 456,976 VTVs
A9A000-A9Z999					Cannot mix incremental portions
######-#####@					National characters cannot increment
AA00##-ZZ99##					Invalid range
CCNNZZ-CDNZAA		CCNNZZ-CDNZAA		464,414	Greater than 456,976 VTVs
A4Z#@0-A9Z#@9					Invalid range

APPENDIX C

Using the HSC Significant Event Notification Facility

HSC provides a Significant Event Notification Facility (SEN). The SEN has a macro interface that allows an application to request notification of specific HSC and VTCS events. The application can then process the data passed by the SEN. SEN notification requests are maintained across HSC warm starts and are purged on HSC cold starts. When HSC is cold started, therefore, all previously established notification requests must be reestablished.

The SEN macro interface is supported at HSC base service level. There is no operator interface for the SEN itself, but the HSC 5.0.0 and above Display command lists SEN notification requests.

For information on the XML format output of the supported SEN events, see "VTCS and HSC Events XML Tags" on page 348.

Caution – When HSC initializes, the SEN facility is disabled, so you must enable via the HSC OPTION SEN operator command. For example, to enable the SEN, enter the following:

```
.OPTION SEN=ON
```

Overview of the SEN Macro Interface

You invoke the SLSXSEN macro to make SEN requests. The SLSXSEN macro has two forms:

- A **list form** that generates a parameter list. The parameter list is mapped by macro SLSXSENM and must be included in routines invoking SLSXSEN.
- The **execute form** that populates the parameter list and calls the SEN request module SLSXSENR. Modules using the SLSXSEN macro must have an authorization code of 1 and reside in an APF authorized library.

The SLSXSEN macro supports the following four requests:

LISTEN

Request notification for SEN defined events, which requires providing a routine to be invoked when the event occurs.

DELETE

Remove a specific event notification request.

DISABLE

Disable the LISTEN routine for event notification request.

ENABLE

Enable the LISTEN routine for event notification request.

Programs invoking SLSXSEN LISTEN requests must supply the entry points of two routines:

- The SEN request module SLSXSENR.
- The listener routine to get control when an event occurs.

SLSXSENR must be loaded from an APF authorized library. The listener routine must reside in common storage. The invoking program is responsible for managing this storage.

The listener routine is validated during LISTEN request processing by invoking the routine with general purpose register 1 set to zeros. Therefore, the listener routine must be sensitive to this condition and process it accordingly. The listener routine can simply return back to the caller on this condition or perform any type of initialization required. If LISTEN request processing detects an abend during validation, the request is rejected. A successful LISTEN request returns a token. Programs invoking SLSXSEN with DELETE, DISABLE, or ENABLE requests must provide this token to target a specific LISTEN request. The HSC SAMPLIB contains a sample program showing the use of SLSXSEN.

When events occur for which there is a listener routine, it is invoked and receives data that describes the event. The data passed is in XML format, which is provided on the StorageTek Customer Resource Center (CRC). For more information on the supported events, see "Supported HSC and VTCS SEN Events" on page 344.

When SEN request processing completes and control is returned to the requesting program, the parameter list and Register 15 contain the return code. Return codes are described in "Execute Form - Syntax and Parameters" on page 328.

If HSC detects an abend while the listener routine is in control, the associated request is disabled and must be programmatically reenabled or deleted. When a notification request is disabled, the listener routine is not invoked.

Note – The HSC 5.0 OPTION command and control statement lets you set the SEN facility ON or OFF.

OPTion SEN(ON) | OPTion SEN(OFF)

Note that even if SEN is disabled, SLSXSEN requests can still be processed.

SEN Macro Interface Authorization Requirements and Module Attributes

Modules invoking the SEN macro interface must:

- Have an authorization code of 1 and reside in an APF authorized library, **or**
- Be running in key 0 - 7 or supervisor state and be running in AMODE 31.

The listener routine must be running in TCB mode (not in SRB mode) and must be re-entrant.

The SLSXSEN Macro

The SLSXSEN macro has two forms:

- A **list form** that generates a parameter list. The parameter list is mapped by macro SLSXSENM (see "SLSXSENM Macro Format" on page 327) and must be included in routines invoking SLSXSEN. On completion of SEN request processing, the SEN parameter list contains the return code and token response areas.
- The **execute form** that populates the parameter list and calls the SEN request module SLSXSENR.

List Form - Syntax and Parameters

Syntax

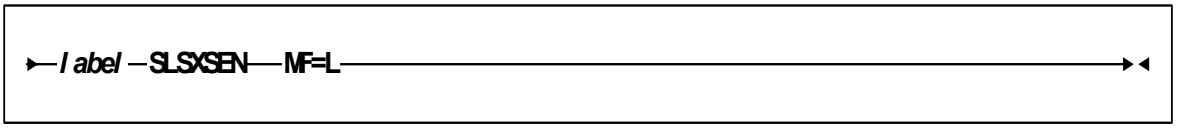


FIGURE C-1 SLSXSEN Macro Syntax - List Form

Parameters

MF=L

Specifies that a remote parameter list is generated. All other options are ignored.

SLSXSENM Macro Format

TABLE 5-2 shows the format of macro SLSXSENM, which generates a DSECT that maps the SEN parameter list generated by the list form of macro SLSXSEN.

TABLE 5-2 SLSXSENM Macro Format

Decimal Offset	Hexadecimal Offset	Type	Length	Label	Description
0	0			SLSXSENM	Start of DSECT
0	0	hexstring	2	SLSXSLEN	Length of parameter list
2	3	bitstring	1	SLSXSFC	SEN request code
		X’01’		SLSXLIS	LISTEN
		X’02’		SLSXSDEL	DELETE
		X’03’		SLSXSDIS	DISABLE
		X’04’		SLSXSENA	ENABLE
3	3	bitstring	1	SLSXSFLG	Processing flag
		X’80’		SLSXSEOT	EOT = YES
		X’40’		SLSXSEOM	EOM = YES
4	4	hexstring	8	SLSXSEM	Event mask
12	C	hexstring	4	SLSXSLA	Listener routine address
16	10	hexstring	4	SLSXSRT	Current TCB of requestor
20	14	hexstring	4	SLSXSRA	ASCB of requestor
22	16	hexstring	2	SLSXSTOK	Listen request token
30	1E	character	8	SLSXSREQ	Requestor name
38	26	character	8	SLSXSLNR	Listener routine name
40	28	hexstring	2	SLSXSQA	SEN queue entry address
42	2A	hexstring	2	SLSXSRC	SEN return code
			36	SLXRDI	Duplicate request encountered, request ignored
			32	SLXRNHSC	HSC not active or at proper level
			28	SLXRBADA	Authorization requirements not met
			24	SLXRBILR	Listener failed validation
			20	SLSXRSAB	Abend in HSC SEN processing
			16	SLSXRIR	Invalid request
			12	SLSXRNSQ	No LISTEN request found for DELETE, DISABLE, or ENABLE
			08	SLSXRDR	Duplicate listener encountered
			04	SLSXRDOK	SEN request successful, however the SEN facility is disabled
44	2C			SLSXSMLN	Length

Execute Form - Syntax and Parameters

Syntax

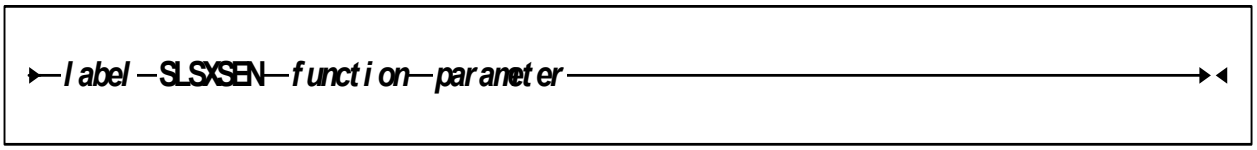


FIGURE C-2 SLSXSEN Macro Syntax - Execute Form

Parameters

function is one of the following:

- “LISTEN” on page 329
- “DELETE” on page 332.
- “DISABLE” on page 334
- “ENABLE” on page 336

LISTEN

LISTEN requests notification for the events described in"Supported HSC and VTCS SEN Events" on page 344. Also see "The SEN Listener Exit Routine" on page 338.

Note – A sample program of a LISTEN request (SENRQST) is provided in file SLSSAMP during SMP/E installation.

Syntax

label SLSXSEN LISTEN,

EVENT=(*event_1,event_2,..*),

RTOKEN=*token*,

LNRADR=*listener_address*,

SENRADR=*SLSXSENR_address*,

EOT=*YES/NO*,

EOM=*YES/NO*,

REQNAME=*requestor_name*,

LNRNAME=*listener_name*,

MF=(E,*parm_list*)

Parameters

EVENT=

Specifies the event(s) requiring notification. For more information, see "Supported HSC and VTCS SEN Events" on page 344. You can specify the event name or the numeric equate. Event names are case sensitive. You can specify a single event or EVENT=ALL. This parameter is required and there is no default.

Note – To specify multiple events on a single LISTEN request, StorageTek recommends that you use the numeric equates to ensure that the parameter value specified will not exceed IBM's 256 byte size limitation for macro variables.

RTOKEN=

Specifies the address of a fullword token associated with a successful LISTEN request. Valid values are any RX-type address or registers 2 through 12. This parameter is required and there is no default.

LNRADR=

Specifies the entry point address of the listener routine that is invoked when the events specified occur. This program must reside in persistent common storage (that is, subpool 241) if EOT = YES or EOM=YES. It is your responsibility to delete the listener routine and free the storage associated with it. Valid values are any RX-type address or registers 2 through 12. This parameter is required and there is no default.

SENRADR=

Specifies the entry point address of the SEN request module (SLSXSENR) that is provided with HSC. SLSXSENR must be loaded before invoking macro SLSXSEN. Valid values are any RX-type address or registers 2 through 12. This parameter is required and there is no default.

EOT=

Specifies whether the listener routine associated with this request is invoked if the task that issued the LISTEN request ends.

YES

Invoke the listener routine if the associated task ends.

NO

Do not invoke the listener routine if the task associated with this request ends.

This parameter is optional and NO is the default.

EOM=

Specifies whether the listener routine associated with this request is invoked if the address space that issued the LISTEN request ends.

YES

Invoke the listener routine if the associated address space ends.

NO

Do not invoke the listener routine if the address space associated with this request ends.

This parameter is optional and NO is the default.

Caution – EOT and EOM control execution of the listener routine, so make sure you set them to specify how you want the listener routine to operate.

REQNAME=

Specifies the name of the requestor and is used to uniquely identify a specify request for display purposes.The name must be an eight byte field containing a combination of uppercase alphanumeric values (A-Z,0-9), national characters ($,@,#) with no embedded blanks.Valid values are any RX-type address or registers 2 through 12. This parameter is optional if LNRNAME is not specified and has no default.

LNRNAME=

Specifies the name of the listener routine The name must be an eight byte field containing a combination of uppercase alphanumeric values (A-Z,0-9), national characters ($,@,#) with no embedded blanks. Valid values are any RX-type address or registers 2 through 12. If LNRNAME is specified, REQNAME must also be specified. Otherwise, this parameter is optional and has no default.

Tip – Although not required, REQNAME and LNRNAME are very useful when displaying active SEN requests. For more information, see "The listener routine must be running in TCB mode (not in SRB mode) and must be re-entrant." on page 261.

MF=(E,*parm_list*)

Specifies the execute form of the macro using the specified parameter list generated by the list form. The parameter list, *parm_list*, can be specified as any RX-type address or registers 2 through 12.

Responses

The return code is set in register 15 and the SEN parameter list. If the return code is zero, the address specified by RTOKEN contains the token representing this LISTEN request. This token is used when deleting, disabling or enabling a request. TABLE C-1 describes the SLSXSEN LISTEN return codes.

TABLE C-1 SLSXSEN LISTEN Return Codes

Return Code	Meaning
00	SEN LISTEN request successful
04	SEN LISTEN request successful however the SEN facility is disabled.
08	Listen request with conflicting event lists. For more information, see "Detecting Duplicate Listen Requests" on page 277.
12	N/A
16	Invalid parameter list received.
20	Indicates that an abend occurred in HSC SEN processing, request rejected.
24	Indicates that listener routine validation failed, request rejected
28	The HSC determined that the SLSXSEN macro is being invoked by a program that does not meet the authorization requirements specified in "SEN Macro Interface Authorization Requirements and Module Attributes" on page 261.
32	The HSC is not active or at the proper initialization level for SEN services.
36	An identical LISTEN request already exists, request ignored. For more information, see "Detecting Duplicate Listen Requests" on page 277.
40	Indicates that the REQNAM or LNRNAME syntax is invalid. Valid syntax is uppercase A-Z, 0-9, valid national characters ($,#,@), with no embedded blanks.

DELETE

Syntax

label SLSXSEN DELETE,

RTOKEN=*token*,

MF=(E,*parm_list*)

Parameters

RTOKEN=

Specifies the address of a fullword token representing the request to be removed. This token was returned from a successful LISTEN request. This parameter is required and there is no default.

MF=(E,*parm_list*)

Specifies the execute form of the macro using the specified parameter list generated by the list form. The parameter list, *parm_list*, can be specified as any RX-type address or registers 2 through 12.

Note – All other parameters are ignored if entered on the DELETE request.

Responses

The return code is set in register 15 and the SEN parameter list. TABLE C-2 describes the SLSXSEN DELETE return codes.

TABLE C-2 SLSXSEN DELETE Return Codes

Return Code	Meaning
00	SEN DELETE request successful.
04	N/A
08	N/A
12	No existing LISTEN request was found using the TOKEN specified on the DELETE request.
16	Invalid request received (not LISTEN, DELETE, DISABLE, or ENABLE).
20	An abend occurred in HSC SEN processing, request rejected.
24	N/A
28	The HSC determined that the SLSXSEN macro is being invoked by a program that does not meet the authorization requirements specified in “SEN Macro Interface Authorization Requirements and Module Attributes” on page 261.
32	HSC is not active or at the proper initialization level for SEN services.
36	N/A

DISABLE

Disable notification for a specific LISTEN request. The targeted request remains but the associated listener routine is not invoked when the specified events occur.

Syntax

label SLSXSEN DISABLE,

RTOKEN=*token*,

MF=(E,parm_list)

Parameters

RTOKEN=

Specifies the address of a fullword token representing the request to be disabled. This token was returned from a successful LISTEN request. This parameter is required and there is no default.

MF=(E,*parm_list*)

Specifies the execute form of the macro using the specified parameter list generated by the list form. The parameter list, *parm_list*, can be specified as any RX-type address or registers 2 through 12.

Note – All other parameters are ignored if entered on the DISABLE request.

Responses

The return code is set in register 15 and the SEN parameter list. TABLE C-3 describes the SLSXSEN DISABLE return codes.

TABLE C-3 SLSXSEN DISABLE Return Codes

Return Code	Meaning
00	SEN DISABLE request successful.
04	SEN DISABLE request successful however the SEN facility is already disabled.
08	N/A
12	No existing LISTEN request was found using the TOKEN specified on the DISABLE request.
16	Invalid request received (not LISTEN, DELETE, DISABLE, or ENABLE).
20	An abend occurred in HSC SEN processing, request rejected.
24	N/A
28	The HSC determined that the SLSXSEN macro is being invoked by a program that does not meet the authorization requirements specified in “SEN Macro Interface Authorization Requirements and Module Attributes” on page 261.
32	HSC is not active or at the proper initialization level for SEN services.
36	N/A

ENABLE

Enable notification for a specific LISTEN request. The targeted request's listener routine is now invoked when the specified events occur.

Syntax

label SLSXSEN ENABLE,

RTOKEN=*token*,

MF=(E,*parm_list*)

Parameters

RTOKEN=

Specifies the address of a fullword token representing the request to be enabled. This token was returned from a successful LISTEN request. This parameter is required and there is no default.

MF=(E,*parm_list*)

Specifies the execute form of the macro using the specified parameter list generated by the list form. The parameter list, *parm_list*, can be specified as any RX-type address or registers 2 through 12.

All other parameters are ignored if entered on the ENABLE request.

Responses

The return code is set in register 15 and the SEN parameter list. TABLE C-4 describes the SLSXSEN ENABLE return codes.

TABLE C-4 SLSXSEN ENABLE Return Codes

Return Code	Meaning
00	SEN ENABLE request successful.
04	SEN ENABLE request successful however the SEN facility is disabled.
08	N/A
12	No existing LISTEN request was found using the TOKEN specified on the ENABLE request.
16	Invalid request received (not LISTEN, DELETE, DISABLE, or ENABLE).
20	An abend occurred in HSC SEN processing, request rejected.
24	N/A
28	The HSC determined that the SLSXSEN macro is being invoked by a program that does not meet the authorization requirements specified in “SEN Macro Interface Authorization Requirements and Module Attributes” on page 261.
32	HSC is not active or at the proper initialization level for SEN services.
36	N/A

The SEN Listener Exit Routine

To use the SEN, you must write a listener exit routine that is invoked when the requested event(s) occur. The following sections describe the requirements and implementation of this exit.

Input Registers

Registers on entry to the SEN listener routine are as follows:

- R1 points to the address of input parameter list (mapped by macro SWSPGMIA) or zero.
- R13 points to the address of register save area.
- R15 points to the listener routine entry point address.
- All other registers are undefined.

On entry, Register 1 always contains the address of the SWSPGMIA data area except during SEN event registration. When requesting an event, you must provide the address of a listener exit routine that is invoked whenever the specified event(s) occur (for more information, see "LNRADR=" on page 265). As part of this process, HSC invokes the listener exit routine to ensure the address provided is valid. When this validation call is made, Register 1 contains zeros, indicating validation is being performed. During validation, the listener routine can simply return control or perform any routine specific initialization. If the validation call is not successful, the SEN request fails.

Output Registers

Registers on exit from the SEN listener exit routine must be as follows:

- R13 points to the address of original register save area.
- All other registers are undefined (however R15 can be set to 4 upon return to instruct SEN to stop passing any further XML data. For more information, see "How to Tell SEN to Stop Calling Your Listener Routine During Event Processing" on page 276).

Entry Environment

On entry, the SEN listener exit receives control as follows:

- STATE - supervisor
- KEY - key 0
- AMODE: 24 or 31 bit
- LOCKS - none

Listener Exit Routine Programming Considerations

During the SEN registration process each successful request is queued in FIFO order. When a SEN defined event occurs all registered listener exits are invoked serially, in the order in which registration requests were received. A potential performance degradation can occur if a listener exit performs long running units of work. This is particularly critical because each SEN event causes listener exits to be invoked multiple times.

StorageTek **strongly recommend**s that you design "short running" listener exit routines. In addition, the use of facilities that can result in "wait conditions", such as WAIT/POST, STIMER/STIMERM and file I/O, should be used with caution or avoided all together.

The SWSPGMIA Data Area

On entry to the SEN listener exit during event processing, Register 1 contains the address of a data area mapped by macro SWSPGMIA. This storage area contains pointers to the XML defined elements describing each SEN defined event. For more information on the XML structures and tags, see "VTCS and HSC Events XML Tags" on page 348.

The listener exit XML interface basically consists of three types of tag definitions:

- Structure start tags
- Structure end tags
- Data tags and associated data

Structure start tags are passed to the listener exit, without any other data, and are used to indicate a set of logically grouped tags will follow.

Data tags (both start and end) along with the associated data are self defining XML elements.

Structure end tags are passed to the listener exit, without any other data, and are used to indicate the set of logically grouped tags has ended.

The following fields of SWSPGMIA support these XML tag structures:

PGMIRSVA
: Address of complete XML element being passed.

PGMIRSVL
: Length of the start tag plus, optionally, data length plus end tag length.

PGMIRSTA
: Address of XML start or end tag.

PGMIRSTL
: Length of the XML tag.

PGMIRSDA
: Address of associated data, or zero.

PGMIRSDL
: Length of associated data, or zero.

Fields PGMIRSVA, PGMIRSVL, PGMIRSTA, and PGMIRSTL always contain values. PGMIRSVA contains the address of either a structure start tag, a structure end tag or a data start tag. If PGMIRSVA points to a structure start or end tag, PGMIRSVL contains the length of the

tag (including both the leading and trailing '<','>'). If PGMIRSVA points to a data tag, PGMIRSVL contains the length of the tag (including both the leading and trailing '<','>'), plus the length of the data, plus the length of the data end tag.

Fields PGMIRSTA and PGMIRSTL always contain values. PGMIRSTA contains the address of either a structure start tag or a structure end tag. PGMIRSTL contains the length of the tag (including both the leading and trailing '<','>'),

Fields PGMIRSDA and PGMIRSDL optionally contain values. If PGMIRSVA points to either a structure start or structure end tag, PGMIRSDA and PGMIRSDL will contain zero since there is no data associated with these tags. But if PGMIRSVA points to a data start tag, then PGMIRSDA will contain the address of the associated data, and PGMIRSDL will contain the data's length. The following sections show examples of how the listener exit routine processes XML structures and tags.

Example 1

XML tag <libvol_insert_event> is passed to the listener exit, which produces the following:

- PGMIRSVA points to: <libvol_insert_event>.
- PGMIRSVL contains: 00000015.
- PGMIRSTA points to: <libvol_insert_event>.
- PGMIRSTL contains: 00000015.

Because <libvol_insert_event> is a structure tag and has no related data:

- PGMIRSDA contains: 0.
- PGMIRSDL contains: 0.

Example 2

XML tag <hsc_version>5.0.0</hsc_version> is passed to the listener exit, which produces the following:

- PGMIRSVA points to: <hsc_version>.
- PGMIRSVL contains: 00000020.
- PGMIRSTA points to: <hsc_version>.
- PGMIRSTL contains: 0000000D.

Since <hsc_version> is a data start tag, it does have related data. As a result:

- PGMIRSDA points to: 5.0.0.
- PGMIRSDL contains: 00000005.

As described in "Listener Exit Routine Programming Considerations" on page 274, each SEN defined event causes multiple invocations of registered listener exit routines. For example, when an HSC insert volume event occurs the following XML structure is built. The listener routine will be invoked for each of the tag sets:

<libvol_insert_event> (passed on the 1st call to listener exit)

<header> (passed on 2nd call to the listener exit)

<hsc_version>5.0.0</hsc_version> (passed on 3rd call to the listener exit)

<date>2006Mar19</date> (passed on 4th call to the listener exit)

<time>17:53:17</time> (passed on 5th call to the listener exit)

<host_name>HOSTA</host_name> (passed on 6th call to the listener exit)

</header> (passed on 7th call to listener exit)

<libvol_data> (passed on 8th call to listener exit)

<vol_status>*xx*</*vol_status*> (passed on 9th call to listener exit, *xx* is a 1 byte flag field documented in member SLSUX06P of the SLSMAC installation file, field UX06FLGS)

<volser>VOL001</volser> (passed on 10th call to listener exit)

<volume_location> (passed on 11th call to listener exit)

<acs>00</acs> (passed on 12th call to listener exit)

<lsm>000</lsm> (passed on 13th call to listener exit)

<panel>07</panel> (passed on 14th call to listener exit)

<row>02</row> (passed on 15th call to listener exit)

<column>10</column> (passed on 16th call to listener exit)

</volume_location> (passed on 17th call to listener exit)

<select_count>0</select_count> (passed on 18th call to listener exit)

</libvol_data> (passed on 19th call to listener exit)

</libvol_insert_event> (passed on 20th and final call to listener exit)

How to Tell SEN to Stop Calling Your Listener Routine During Event Processing

Because listener routines are invoked to process each XML tag for the event, the data required by the listener may be satisfied before the last tag for the event is passed. For example, your listener routine only requires ACS and LSM number when a cartridge is entered into an LSM. Using "Example 2" on page 340, the listener routines requirements are satisfied on the 13th call. In this situation, a listener routine can set general purpose register 15 to 4 prior to returning control. This instructs SEN not to call it any more for this event. The remaining XML tags are bypassed.

Detecting Duplicate Listen Requests

When an application attempts to register a Listen request with the SEN facility, HSC first validates the request. As part of this validation, HSC insures duplicate registration requests are not accepted to prevent the same user exit routine from being called multiple times for a single event. The following list differentiates between a unique and a duplicate registration request:

- If a new Listen request specifies an exit routine entry point address that is different than all other previously registered entry point addresses, the Listen request is accepted.
- If a new Listen request specifies an exit routine entry point address that matches a previously registered entry point address but their respective event lists contain no common events (that is, each Listen request specified a unique event list), the Listen request is accepted.
- If a new Listen request specifies an exit routine entry point address that matches a previously registered entry point address and their respective event lists contain all common events (that is, each Listen request specified identical event lists), the Listen request is rejected as an identical duplicate and terminates with a return code of 36.
- If a new Listen request specifies an exit routine entry point address that matches a previously registered entry point address and their respective event lists contain some common events but not all (that is, each Listen request specified partially matching event lists), the Listen request is rejected as a duplicate with conflicting event lists and terminates with a return code of 8.

Display SEN Command

You can use the Display SEN command to display the SEN LISTEN requests and their statuses. The display can either show a specific request, using the requestor and listener names, a list of related requests using only the requestor name, or all SEN requests. The information displayed includes:

- Requestor name and listener name if any.
- EOT/EOM settings.
- Disable/enable status.
- The token associated with the request.
- The event list of the request.

Syntax

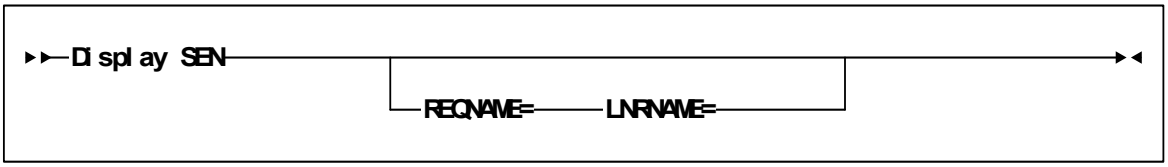


FIGURE C-3 Display SEN Syntax

Parameters

REQNAME=

Specifies the name of the requestor, which is a unique name eight byte alphanumeric field, and is used to uniquely identify a specify request for display purposes.Valid values are any RX-type address or registers 2 through 12. This parameter is optional if LNRNAME is not specified and has no default.

LNRNAME=

Specifies the name of the listener routine The name must be an eight byte alphanumeric field. Valid values are any RX-type address or registers 2 through 12. If LNRNAME is specified, REQNAME must also be specified. Otherwise, this parameter is optional and has no default.

Note – REQNAME= and LNRNAME= are optional. If you specify one, however, you must specify the other.

Supported HSC and VTCS SEN Events

HSC Events

TABLE C-5 describes the supported HSC SEN events.

TABLE C-5 HSC SEN Events

Event Name	Numeric Equate
libvol_insert_event	18
libvol_delete_event	19
hsc_termination_event	20
ACS_added_event	23
ACS_removed_event	24
lsmrail_added_event	25
lsmrail_removed_event	26
libdrive_added_event	27
libdrive_removed_event	28

VTCS SEN Events

TABLE C-6 describes the supported VTCS SEN events.

TABLE C-6 VTCS SEN Events

Event Name	Numeric Equate
vtss_performance_event	01
vtss_chanif_performace_event	02
vtv_mount_event	03
vtv_dismount_event	04
vtv_delete_event	05
mvc_mount_event	06
mvc_dismount_event	07
vtv_migrate_event	08
vtv_recall_event	09
rtd_performance_event	10
rtd_vary_event	11
mvc_usage_event	12

TABLE C-6 VTCS SEN Events

Event Name	Numeric Equate
vtss_performance_event	01
vtv_movement_event	13
vtv_scratch_event	14
vtv_replicate_event	15
vtv_unlink_from_mvc_event	16
clink_vary_event	17

SEN Messages

SLS4970I

SEN QUEUE DISPLAY STATUS CURRENT REQUESTORS *99999999* REQNAME *RRRRRRRR* LNRNAME *LLLLLLLL* TOKEN *TTTTTTTT* FLAGS *FFFFFFFF* EVENT *EEEEEEEE*

Explanation. This is the output from the Display SEN Queue command. The output produced varies depending on the parameters specified. If no parameters are specified all entries in the SEN QUEUE are displayed. If REQname is specified then only Requestors matching REQname will be displayed. If LNRname is specified REQname must be specified. If LNRname is specified then only Requestors and Listeners matching REQname and LNRname will be displayed.

99999999 Number of Requestors on the SEN QUEUE *RRRRRRRR* Requestor name

LLLLLLLL Listener name

TTTTTTTT Hex display of the Requestor Token

FFFFFFFF Description of the processing flags for the SEN QUEUE ENTRY

EEEEEEEE Description of the Event to be listened for.

System Action. None.

User Response. None.

SLS4971I

Significant Event Notification facility not active

Explanation. An attempt was made to display SEN Queue entries. However, Significant Event Notification facilitate is not active.

System Action. The DISPLAY SEN command is rejected.

User Response. Make certain you are running the correct version of HSC. If you are certain you are running the correct version contact StorageTek HSC support.

SLS4972I

SEN QUEUE IS EMPTY

Explanation. An attempt was made to display SEN Queue entries. However, the SEN QUEUE contained no entries.

System Action. Processing continues.

User Response. Make certain you have run the task to add entries to the SEN QUEUE. If the task has run corselette contact StorageTek HSC support.

SLS4973I

REQNAME *nnnnnnnn* NOT FOUND

Explanation. An attempt was made to display a specific Requestor on the SEN QUEUE. However, the specified Requestor was not found.

System Action. Processing continues.

User Response. Make certain you have spelled the Requestor name correctly.

SLS4974I

LNRNAME *nnnnnnnn* NOT FOUND

Explanation: An attempt was made to display a specific Listener on the SEN QUEUE. However, the specified Listenor was not found.

System Action: Processing continues.

User Response: Make certain you have spelled the Listener name correctly.

VTCS and HSC Events XML Tags

This section describes the XML output of the VTCS and HSC events processed by the HSC Significant Event Notification (SEN) facility as follows:

- Listing of the data tags.
- Tag definitions.
- Cross reference of data tags to structure or event tags.

For more information, see the following sections:

- "VTCS Events XML Tags" on page 349
- "HSC Events XML Tags" on page 353

VTCS Events XML Tags

TABLE C-7 VTCS Events XML Tags

XML Tag	Definition	Where Used
<addresses_trapped>	From SMF11NAT in turn from NOADRTRP parm returned from Channel Interface Performance ECAM-T request. Contains the number of unit addresses trapped by a channel interface.	<chanif_data>
<async_end_tod>	Structure that contains date and time that the asynchronous event completed. Date is supplied as: <date>yyyymmmdd</date> Time is supplied as: <time>hh:mm:ss.thm</time>	<vtv_mount_event>
		<vtv_dismount_event>
		<mvc_mount_event>
		<vtv_migrate_event>
		<vtv_recall_event>
		<vtv_replicate_event>
<async_start_tod>	Structure that contains date and time that the asynchronous event started. Contains <date> and <time> as in <async_end_tod>.	<vtv_mount_event>
		<vtv_dismount_event>
		<mvc_mount_event>
		<vtv_migrate_event>
		<vtv_recall_event>
		<vtv_replicate_event>
<base_cache_size>	Cache size in MB of VTSS	<vtss_performance_event>
<bytes_read_freespace_collecti on>	Bytes read for free space collection within a partition.	<part_data>
<bytes_xfered_read>	Bytes transferred for read	<rtd_preformance_event>
<bytes_xfered_write>	Bytes transferred for write	<rtd_preformance_event>
<chanif_data>	Structure that contains all the information for a single channel IF. Contains:	<vtss_chanif_performance_event>
	<name>	
	<installed>	
	<enabled>	

TABLE C-7 VTCS Events XML Tags

XML Tag	Definition	Where Used
	<link_type>	
	<addresses_trapped>	
	<speed>	
	<io_count>	
	<cu_busy_count>	
<collected_free_backend_capacity>	Bytes read for free space collection with the VTSS partition.	<part_data>
<cu_busy_count>	Count of control unit busy events on a chan IF	<chanif_data>
<customer_cache_size>	Customer cache size in MB	<vtss_performance_event>
<dev_activity>	RTD activity (initial selects)	<rtd_preformance_event>
<dev_available_time>	RTD mounted time	<rtd_preformance_event>
<dev_connect_time>	RTD connected time	<rtd_preformance_event>
<dev_util_time>	RTD utilized time	<rtd_preformance_event>
<dsname>	Data set name of the VTV being mounted/dismounted	<vtv_mount_event>
		<vtv_dismount_event>
<ecam_bypassed_bufferspace_count>	Count of ECAM-T requests bypassed – no buffer space	<vtss_performance_event>
<ecam_bypassed_configbusy_count>	Count of ECAM-T requests bypassed – configuration busy	<vtss_performance_event>
<ecam_processed_count>	Count of ECAM-T requests processed	<vtss_performance_event>
<enabled>	Indicates if a channel if is enabled - contains YES/NO	<chanif_data>
<installed>	Indicates if a channel if installed - contains YES/NO	<chanif_data>
<io_count>	Number of I/Os on this channel interface since last report.	<chanif_data>
<jobname>	MVS jobname requesting the mount/dismount	<vtv_mount_event>
		<vtv_dismount_event>
<link_type>	Identifies the link type contains HOST/RTD	<chanif_data>
<mount_type>	Indicates the type of mount performed. Contains:	<vtv_mount_event>
	EXISTING-SPECIFIC	
	CREATE-SL	

TABLE C-7 VTCS Events XML Tags

XML Tag	Definition	Where Used
	CREATE-AL	
	EXISTING-SCRATCH	
<name>	The configured name of the channel IF.	<chanif_data>
<new_mvc>	Structure that describes the target MVC on a VTV move. Contains <mvc_data>	<vtv_movement_event>
<nvs_size>	NVS Size (MB)	<vtss_performance_event>
<offline_cache_size>	Offline cache size	<vtss_performance_event>
<old_mvc>	Structure that describes the source MVC on a VTV move. Contains <mvc_data>	<vtv_movement_event>
<part_data>	Structure containing all the information from a VTSS partition. Contains:	<vtss_performance_event>
	<total_backend_capacity>	
	<total_free_backend_capacity>	
	<collected_free_backend_capacity	
	<bytes_read_freespace_collection>	
	<standard_capacity_defined>	
< name>	Name of a VTSS partition	<part_data>
<pinned_cache_size>	Pinned Cache Size	<vtss_performance_event>
<read_buffered_log>	64-byte rbl data represented as 128 characters	<mvc_dismount_event>
<read_only>	Indicates whether or not the VTV was mounted READONLY – contains YES/NO	<vtv_mount_event>
<reason>	The reason for the VTV delete, MVC mount, VTV migrate, VTV recall or the VTV/MVC unlink.	<vtv_delete_event>
		<mvc_mount_event>
		<vtv_migrate_event>
		<vtv_recall_event>
		<vtv_unlink_from_mvc_event>
<recall_required>	Indicates whether or not a recall was required in order to mount the VTV – contains YES/NO	<vtv_mount_event>
<recall_with_error>	Indicates if the recall was performed with error – contains YES/NO	<vtv_recall_event>
<sense_data>	32-byte sense data represented as 64 characters	<mvc_mount_event>
		<mvc_dismount_event>

TABLE C-7 VTCS Events XML Tags

XML Tag	Definition	Where Used
		<vtv_recall_event>
		<vtv_replicate_event>
<speed>	The speed of a channel IF.	<chanif_data>
<standard_capacity_ defined>	Standard capacity defined for the partition	<part_data>
<stepname>	MVS stepname of the job requesting the mount/dismount of the VTV	<vtv_mount_event>
		<vtv_dismount_event>
<total_backend_capacity>	Total backend capacity for a partition	<part_data>
<total_free_backend_ capacity>	Total free backend capacity for a partition.	<part_data>

HSC Events XML Tags

TABLE C-8 HSC Events XML Tags

Head Tag	Structure/Data Tags			Definition
<header>				header structure, occurs in: <libvol_ insert_ event> <libvol_ delete_ event> <hsc_ termination_ event> **Note:** Any x22 abends resulting from the cancellation of HSC will not generate the HSC termination event. <libdrive_ added_event> <libdrive_ removed_ event> <lsmrail_ added_event> <lsmrail_ removed_ event>
	<hsc_version>			HSC Version
	<date>			date
	<time>			time
	<host_name>			host name
<libdrive_added_event>				dynamically added drive event
	<libdrive_data>			drive data
		<location_data>		drive location
			<acs>	ACS
			<lsm>	LSM
			<panel>	panel
			<libdrive_number>	drive number
		<libdrive_model>		drive model (for example, 9840C)

TABLE C-8 HSC Events XML Tags

Head Tag	Structure/Data Tags			Definition
		<libdrive_unit_address>		MVS unit address
<libdrive_removed_event>				dynamically removed drive event
	<libdrive_data>			drive data
		<location_data>		drive location
			<acs>	ACS
			<lsm>	LSM
			<panel>	panel
			<libdrive_number>	drive number
		<libdrive_model>		drive model (for example, 9840C)
		<libdrive_unit_address>		MVS unit address
<lsmrail_added_event>				dynamically added LSM rail event (SL8500 only)
	<lsmrail_data>			LSM rail data
		<location_data>		rail location
			<acs>	ACS
			<lsm>	LSM
<lsmrail_removed_event>				dynamically removed LSM rail event (SL8500 only)
	<lsmrail_data>			LSM rail data
		<location_data>		rail location
			<acs>	ACS
			<lsm>	LSM

Index

O

P

R

S

T

V

W