

**StorageTek Enterprise Library Software**  
Command, Control Statement, and Utility Reference  
Release 7.3 for MVS  
**E60543-07**

October 2019

StorageTek Enterprise Library Software Command, Control Statement, and Utility Reference, Release 7.3 for MVS

E60543-07

Copyright © 2015, 2019, 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 documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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 that 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 its 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

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. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about 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 unless otherwise set forth in an applicable agreement between you and Oracle. 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, except as set forth in an applicable agreement between you and Oracle.

---

---

# Contents

<b>Preface</b> .....	xli
Audience .....	xli
Documentation Accessibility .....	xli
Related Documents .....	xlii
Conventions .....	xlii
<b>What's New?</b> .....	xlvii
<b>1 ELS Command Interfaces</b>	
<b>Unified User Interface (UI) Support</b> .....	1-1
<b>SMC Command Interfaces</b> .....	1-1
Issuing SMC Commands from the Console .....	1-2
Issuing Commands Using an SMC Command Prefix .....	1-2
Issuing SMC Commands Using the MVS Modify Command .....	1-2
Keyword Parameters .....	1-3
Specifying SMC Commands in the SMCCMDS or SMCPARMS Data Sets .....	1-3
SMC Commands that Specify JOBname, STEPname, and PROCstep .....	1-4
Issuing SMC Commands from a Utility .....	1-4
<b>HSC and VTCS Command Interfaces</b> .....	1-5
Issuing HSC and VTCS Commands from the Console .....	1-5
Issuing HSC and VTCS Commands Using a Command Prefix .....	1-5
Issuing HSC and VTCS Commands Using the MVS Modify Command .....	1-5
Keyword Parameters .....	1-6
Issuing HSC and VTCS Commands from a Utility .....	1-6
Issuing HSC and VTCS Commands from PARMLIB .....	1-7
HSC Service Levels .....	1-8
Base Service Level Functions .....	1-8
Mount Requests Intercepted During Base Service Level Operations .....	1-8
Full Service Level Functions .....	1-8
<b>2 SMC Commands, Utilities, and Control Statements</b>	
<b>ALLOCDf</b> .....	2-1
Description .....	2-1
Syntax .....	2-1
Parameters .....	2-2

Optional Parameters .....	2-3
<b>ALLOCJob</b> .....	2-7
Description .....	2-7
Syntax.....	2-7
Parameters.....	2-8
Optional Parameters .....	2-8
<b>CMDDef</b> .....	2-10
Description .....	2-10
Syntax.....	2-10
Parameters.....	2-10
<b>COLlector</b> .....	2-11
Description .....	2-11
Syntax.....	2-11
Parameters.....	2-12
Optional START Parameters .....	2-12
<b>COMMtest</b> .....	2-13
Description .....	2-13
Syntax.....	2-13
Parameters.....	2-14
<b>DISMount</b> .....	2-15
Description .....	2-15
Syntax.....	2-15
Parameters.....	2-16
<b>Display DRive</b> .....	2-16
Description .....	2-16
Syntax.....	2-16
Parameters.....	2-17
<b>Display POLicy</b> .....	2-18
Description .....	2-18
Syntax.....	2-18
Parameters.....	2-19
<b>Display RC</b> .....	2-19
Description .....	2-19
Syntax.....	2-19
Parameters.....	2-19
<b>Display SERVer</b> .....	2-20
Description .....	2-20
Syntax.....	2-20
Parameters.....	2-20
<b>Display SMC</b> .....	2-21
Description .....	2-21
Syntax.....	2-21
Parameters.....	2-21
<b>Display STORMNGR</b> .....	2-21
Description .....	2-21
Syntax.....	2-22
Parameters.....	2-22



<b>Display TAPEPlex</b> .....	2-22
Description .....	2-22
Syntax.....	2-22
Parameters.....	2-23
<b>Display Volume</b> .....	2-23
Description .....	2-23
Syntax.....	2-23
Parameters.....	2-24
<b>DRIVemap</b> .....	2-24
Description .....	2-24
Syntax.....	2-24
Parameters.....	2-25
Optional Parameters .....	2-25
<b>Help</b> .....	2-26
Description .....	2-26
Syntax.....	2-26
Parameters.....	2-26
<b>HTTP</b> .....	2-27
Description .....	2-27
Syntax.....	2-27
Parameters.....	2-28
Tuning Parameters.....	2-29
<b>IDAX</b> .....	2-29
Description .....	2-29
Syntax.....	2-30
Parameters.....	2-30
Optional Parameters .....	2-30
<b>LIMIT</b> .....	2-31
Description .....	2-31
Syntax.....	2-32
Parameters.....	2-33
<b>List</b> .....	2-35
Description .....	2-35
Syntax.....	2-35
Parameters.....	2-36
<b>LOG</b> .....	2-36
Description .....	2-37
Syntax.....	2-37
Parameters.....	2-37
Optional Parameters .....	2-38
<b>METAdata</b> .....	2-39
Description .....	2-39
Syntax.....	2-40
Parameters.....	2-40
<b>MONitor</b> .....	2-41
Description .....	2-41
Syntax.....	2-41

Parameters.....	2-42
Monitor Parameters.....	2-42
<b>MOUNT</b> .....	2-44
Description.....	2-44
Syntax.....	2-44
Parameters.....	2-44
<b>MOUNTDef</b> .....	2-45
Description.....	2-45
Syntax.....	2-45
Parameters.....	2-46
Optional Parameters.....	2-47
<b>MSGDef</b> .....	2-49
Description.....	2-50
Syntax.....	2-50
Parameters.....	2-50
Optional Parameters.....	2-50
<b>MSGJob</b> .....	2-52
Description.....	2-52
Syntax.....	2-52
Parameters.....	2-52
Optional Parameters.....	2-53
<b>POLicy</b> .....	2-53
Description.....	2-54
Syntax.....	2-55
Parameters.....	2-57
Named Policy Parameters.....	2-58
<b>READ</b> .....	2-62
Description.....	2-62
Syntax.....	2-62
Parameters.....	2-63
<b>RESYNChronize</b> .....	2-64
Description.....	2-64
Syntax.....	2-65
Parameters.....	2-65
<b>Route</b> .....	2-65
Description.....	2-65
Syntax.....	2-66
Parameters.....	2-66
<b>SERVer</b> .....	2-66
Description.....	2-67
Syntax.....	2-67
Parameters.....	2-68
Named SERVer Parameters.....	2-69
<b>SIMulate</b> .....	2-70
Description.....	2-71
Syntax.....	2-71
Parameters.....	2-71

Examples .....	2-73
<b>SMSDef</b> .....	2-74
Description .....	2-74
Syntax.....	2-74
Parameters.....	2-75
Optional Parameters .....	2-75
<b>STORMNGR</b> .....	2-77
Description .....	2-77
Syntax.....	2-77
Parameters.....	2-78
<b>TAPEPlex</b> .....	2-79
Description .....	2-79
Syntax.....	2-79
Parameters.....	2-80
Named TAPEPlex Parameters .....	2-81
<b>TCPip</b> .....	2-82
Description .....	2-82
Syntax.....	2-82
Parameters.....	2-82
TCPIP Parameters .....	2-83
<b>TRace</b> .....	2-84
Description .....	2-85
Syntax.....	2-85
Parameters.....	2-86
Jobname/Component Parameters.....	2-87
Optional Parameters .....	2-88
<b>TREQDef</b> .....	2-88
Description .....	2-89
Syntax.....	2-89
Parameters.....	2-89
TAPEREQ Control Statement.....	2-90
Syntax .....	2-91
Parameters.....	2-92
Disabling a TAPEREQ Definition.....	2-96
<b>UExit</b> .....	2-96
Description .....	2-96
Syntax.....	2-97
Parameters.....	2-97
<b>UNITAttr</b> .....	2-98
Description .....	2-98
Syntax.....	2-99
Parameters.....	2-100
Device Attributes .....	2-101
<b>USERMsg</b> .....	2-101
Description .....	2-102
Syntax.....	2-102
Parameters.....	2-102

<b>VMSG</b> .....	2-102
Description .....	2-102
Syntax.....	2-103
Parameters.....	2-103
<b>XCLIENT</b> .....	2-105
Description .....	2-105
Syntax.....	2-105
Parameters.....	2-105
<b>XUDB</b> .....	2-106
Description .....	2-107
Syntax.....	2-107
Parameters.....	2-107
XUDB Security Data set.....	2-108

### 3 HSC Commands, Utilities, and Control Statements

<b>ACTIVities</b> .....	3-1
Description .....	3-1
Syntax.....	3-2
Parameters.....	3-2
Additional JCL Requirements .....	3-3
Additional Sort Control Cards.....	3-4
<b>ACTMVCgn</b> .....	3-4
Description .....	3-4
Syntax.....	3-4
Parameters.....	3-5
Additional JCL Requirements .....	3-5
<b>ARCHive</b> .....	3-6
Description .....	3-6
Syntax.....	3-6
Parameters.....	3-6
Examples .....	3-7
ARCHive Report (MOVEVTV not specified) .....	3-7
Fields.....	3-8
Fields (MVC Report) .....	3-9
ARCHive MOVEVTV Report (MOVEVTV specified).....	3-9
<b>AUDit</b> .....	3-10
Description .....	3-10
Syntax.....	3-10
Parameters (Real Volume Audit).....	3-11
Optional Parameters (ALL or ACS) .....	3-12
Optional Parameters (ACS Only) .....	3-13
Parameters (Virtual Volume Audit) .....	3-14
Parameters (Virtual Volume Audit) .....	3-14
Audit Report .....	3-15
AUDit Report Messages.....	3-16
<b>BACKup</b> .....	3-19
Description .....	3-19

Syntax.....	3-19
Parameters.....	3-19
Additional JCL Requirements .....	3-19
<b>CANcel</b> .....	3-20
Description .....	3-20
Syntax.....	3-20
Parameters.....	3-20
<b>CAPPref</b> .....	3-21
Description .....	3-21
Syntax.....	3-21
Parameters.....	3-21
<b>CDs</b> .....	3-23
Description .....	3-23
Syntax.....	3-24
Parameters.....	3-24
<b>CDSCREat</b> .....	3-25
Description .....	3-25
Syntax.....	3-25
Parameters.....	3-26
Additional JCL Requirements .....	3-27
<b>CDSData</b> .....	3-28
Description .....	3-28
Syntax.....	3-28
Parameters.....	3-28
<b>CDSDEF</b> .....	3-29
Description .....	3-29
Syntax.....	3-29
Parameters.....	3-30
<b>Clean</b> .....	3-31
Description .....	3-31
Syntax.....	3-32
Parameters.....	3-32
<b>COMMPath</b> .....	3-32
Description .....	3-32
Syntax.....	3-33
Parameters.....	3-34
<b>CONFIg</b> .....	3-37
Description .....	3-37
Syntax.....	3-37
Parameters.....	3-37
CONFIg CLINK Statement.....	3-39
Syntax .....	3-40
Parameters.....	3-40
CONFIg CLUSTER Statement.....	3-41
Syntax .....	3-41
Parameters.....	3-41
CONFIg GLOBAL Statement .....	3-42

Syntax .....	3-42
Parameters.....	3-43
CONFIG HOST Statement.....	3-50
Syntax .....	3-50
Parameters.....	3-50
CONFIG MVCVOL Statement.....	3-51
CONFIG RECLAIM Statement .....	3-51
Syntax .....	3-51
Parameters.....	3-51
CONFIG RTDpath Statement.....	3-53
Syntax .....	3-53
Parameters.....	3-54
CONFIG STORMNGR Statement.....	3-55
Syntax .....	3-55
Parameters.....	3-56
CONFIG TAPEPLEX Statement.....	3-56
Syntax .....	3-56
Parameters.....	3-57
CONFIG VTD Statement .....	3-57
Syntax .....	3-58
Parameters.....	3-58
Specifying VTD Unit Addresses .....	3-58
CONFIG VTSS Statement .....	3-59
Syntax .....	3-59
Parameters.....	3-60
CONFIG VTVVOL Statement .....	3-62
<b>CONSolid</b> .....	3-62
Description .....	3-62
Syntax.....	3-63
Parameters.....	3-63
CONSolid Report Messages .....	3-63
<b>DBSERVer</b> .....	3-64
Description .....	3-65
Syntax.....	3-65
Parameters.....	3-65
<b>DEComp</b> .....	3-66
Description .....	3-66
Syntax.....	3-66
Parameters.....	3-67
Output.....	3-67
FLATDD Parameter Output.....	3-67
SLSPRINT Output.....	3-67
<b>DELETSCR</b> .....	3-68
Description .....	3-68
Syntax.....	3-68
Parameters.....	3-69
DELETSCR Report .....	3-70

<b>DIRBLD</b> .....	3-71
Description.....	3-71
Syntax.....	3-71
Parameters.....	3-71
<b>DISMount</b> .....	3-71
Description.....	3-72
Syntax.....	3-72
Parameters.....	3-72
<b>Display</b> .....	3-73
Display ACS.....	3-74
Description.....	3-75
Syntax.....	3-75
Parameters.....	3-75
Output.....	3-75
Display ACTive.....	3-76
Description.....	3-76
Syntax.....	3-76
Parameters.....	3-76
Output.....	3-76
Fields.....	3-77
Display ALL.....	3-83
Description.....	3-83
Syntax.....	3-83
Parameters.....	3-84
Display Cap.....	3-84
Description.....	3-84
Syntax.....	3-84
Parameters.....	3-84
Output.....	3-85
Display CDS.....	3-86
Description.....	3-86
Syntax.....	3-86
Parameters.....	3-86
Output.....	3-87
Display CLInk.....	3-87
Description.....	3-87
Syntax.....	3-88
Parameters.....	3-88
Output.....	3-88
Fields.....	3-88
Display CLUster.....	3-90
Description.....	3-90
Syntax.....	3-90
Parameters.....	3-90
Output.....	3-90
Fields.....	3-91
Display CMD.....	3-92

Description.....	3-93
Syntax .....	3-93
Parameters.....	3-93
Display COMMPath .....	3-93
Description.....	3-93
Syntax .....	3-94
Parameters.....	3-94
Display CONFIG .....	3-94
Description.....	3-94
Syntax .....	3-94
Parameters.....	3-95
Output.....	3-95
Fields.....	3-95
Display DRives .....	3-98
Description.....	3-98
Syntax .....	3-98
Parameters.....	3-99
Display DRIVE_INFO .....	3-100
Description.....	3-100
Syntax .....	3-100
Parameters.....	3-101
Display EXceptns .....	3-102
Description.....	3-102
Syntax .....	3-103
Parameters.....	3-103
Output.....	3-103
Display LINKSto .....	3-104
Description.....	3-104
Syntax .....	3-104
Parameters.....	3-105
Output.....	3-105
Fields.....	3-105
Display LMUPDEF .....	3-107
Description.....	3-107
Syntax .....	3-107
Parameters.....	3-107
Display LOCKs.....	3-107
Description.....	3-107
Syntax .....	3-107
Parameters.....	3-108
Output.....	3-108
Fields.....	3-108
Display Lsm .....	3-109
Description.....	3-109
Syntax .....	3-109
Parameters.....	3-109
Output.....	3-110



Display Message.....	3-110
Description.....	3-111
Syntax .....	3-111
Parameters.....	3-111
Display MGMTDEF .....	3-111
Description.....	3-111
Syntax .....	3-112
Parameters.....	3-112
Display MIGrate.....	3-112
Description.....	3-112
Syntax .....	3-112
Parameters.....	3-112
Output (Display Migrate) .....	3-113
Fields.....	3-113
Output (Display Migrate DETail).....	3-113
Fields.....	3-114
Output (Display MIGrate AUTO) .....	3-115
Output (Display MIGrate DELAY) .....	3-115
Display MNTD .....	3-115
Description.....	3-115
Syntax .....	3-115
Parameters.....	3-116
Display MONitor .....	3-116
Description.....	3-116
Syntax .....	3-116
Parameters.....	3-116
Display MVC .....	3-116
Description.....	3-117
Syntax .....	3-117
Parameters.....	3-117
Output.....	3-117
Fields.....	3-117
Display MVCPool .....	3-122
Description.....	3-122
Syntax .....	3-123
Parameters.....	3-123
Output.....	3-123
Fields.....	3-125
Display OPTion .....	3-127
Description.....	3-127
Syntax .....	3-127
Parameters.....	3-127
Display PATH.....	3-127
Description.....	3-127
Syntax .....	3-127
Parameters.....	3-128
Output.....	3-128

Fields.....	3-128
Display Queue .....	3-129
Description.....	3-130
Syntax .....	3-130
Parameters.....	3-130
Output.....	3-130
Display REPlicat.....	3-130
Description.....	3-130
Syntax .....	3-130
Parameters.....	3-131
Output.....	3-131
Fields.....	3-132
Display Requests.....	3-132
Description.....	3-132
Syntax .....	3-133
Parameters.....	3-133
Display RTD.....	3-133
Description.....	3-133
Syntax .....	3-133
Parameters.....	3-133
Output.....	3-133
Fields.....	3-134
Output.....	3-136
Fields.....	3-136
Display SCRAtch.....	3-137
Description.....	3-137
Syntax .....	3-137
Parameters.....	3-137
Output.....	3-139
Display SEN.....	3-140
Description.....	3-141
Syntax .....	3-141
Parameters.....	3-141
Supported HSC SEN Events.....	3-141
Supported VTCS SEN Events.....	3-141
Display SERVER.....	3-142
Description.....	3-142
Syntax .....	3-142
Parameters.....	3-142
Output.....	3-143
Display SRVlev .....	3-143
Description.....	3-143
Syntax .....	3-143
Parameters.....	3-143
Display Status .....	3-143
Description.....	3-144
Syntax .....	3-144

Parameters.....	3-144
Output.....	3-144
Display STORCLas.....	3-144
Description.....	3-145
Syntax .....	3-145
Parameters.....	3-145
Output.....	3-145
Display STORMNgr.....	3-146
Description.....	3-146
Syntax .....	3-146
Parameters.....	3-146
Output.....	3-146
Fields.....	3-147
Display TASKs.....	3-147
Description.....	3-148
Syntax .....	3-148
Parameters.....	3-148
Output.....	3-148
Fields.....	3-148
Display THReshld .....	3-149
Description.....	3-149
Syntax .....	3-149
Parameters.....	3-150
Output.....	3-151
Display Volser.....	3-152
Description.....	3-152
Syntax .....	3-152
Parameters.....	3-152
Output.....	3-152
Display VOLume_Info .....	3-154
Description.....	3-154
Syntax .....	3-154
Parameters.....	3-154
Display VSCRatch .....	3-155
Description.....	3-155
Syntax .....	3-155
Parameters.....	3-155
Output.....	3-155
Fields.....	3-156
Display VTD.....	3-156
Description.....	3-156
Syntax .....	3-156
Parameters.....	3-157
Output.....	3-157
Fields.....	3-157
Display VTSS .....	3-159
Description.....	3-159

Syntax .....	3-159
Parameters.....	3-159
Output.....	3-159
Fields.....	3-160
Display VTSS DETail Output.....	3-161
Fields.....	3-161
Display VTV.....	3-163
Description.....	3-163
Syntax .....	3-163
Parameters.....	3-164
Output.....	3-164
Fields.....	3-164
<b>DRAin</b> .....	3-167
Description .....	3-167
Syntax.....	3-167
Parameters.....	3-168
<b>DRCHKPT</b> .....	3-168
Description .....	3-168
Syntax.....	3-168
Parameters.....	3-169
<b>DRCLEAN</b> .....	3-169
Description .....	3-169
Syntax.....	3-169
Parameters.....	3-169
<b>DRMONitr</b> .....	3-170
Description .....	3-170
Syntax.....	3-170
Parameters.....	3-171
<b>DRTEST</b> .....	3-172
DRTEST CREATE.....	3-173
Description.....	3-173
Syntax .....	3-173
Parameters.....	3-174
Additional JCL Requirements .....	3-175
DRTEST PRIMEprd .....	3-176
Description.....	3-176
Syntax .....	3-177
Parameters.....	3-177
DRTEST RESET .....	3-178
Description.....	3-178
Syntax .....	3-179
Parameters.....	3-179
DRTEST START.....	3-179
Description.....	3-179
Syntax .....	3-179
Parameters.....	3-180
DRTEST STOP .....	3-180

Description.....	3-180
Syntax.....	3-180
Parameters.....	3-180
<b>EEXPORT</b> .....	3-180
Description.....	3-181
Syntax.....	3-181
Parameters.....	3-181
<b>Eject</b> .....	3-183
Description.....	3-183
Syntax.....	3-183
Parameters.....	3-184
Option 1 Parameters.....	3-185
Option 2 Parameters.....	3-186
<b>ENter</b> .....	3-187
Description.....	3-187
Syntax.....	3-187
Parameters.....	3-188
<b>EXECParM</b> .....	3-188
Description.....	3-189
Syntax.....	3-189
Parameters.....	3-189
<b>EXPORT</b> .....	3-190
Description.....	3-190
Syntax.....	3-190
Parameters.....	3-191
Additional JCL Requirements.....	3-192
<b>FMTLOG</b> .....	3-192
Description.....	3-193
Syntax.....	3-193
Parameters.....	3-193
Additional JCL Requirements.....	3-193
<b>IMPORT</b> .....	3-193
Description.....	3-193
Syntax.....	3-194
Parameters.....	3-194
Additional JCL Requirements.....	3-196
<b>INITialize</b> .....	3-196
Description.....	3-196
Syntax.....	3-196
Parameters.....	3-197
Additional JCL Requirements.....	3-198
<b>INVENTORY</b> .....	3-198
Description.....	3-199
Syntax.....	3-199
Parameters.....	3-199
Return Codes.....	3-200
Inventory Report.....	3-201

<b>LIBGen</b> .....	3-201
Description .....	3-201
Syntax.....	3-201
Parameters.....	3-202
Additional JCL Requirements .....	3-202
Output.....	3-202
<b>LMUPDEF</b> .....	3-203
Description .....	3-203
Syntax.....	3-204
Parameters.....	3-204
LMUPATH Control Statement.....	3-205
Syntax .....	3-205
Parameters.....	3-205
<b>LOGUTIL</b> .....	3-206
Description .....	3-207
Syntax.....	3-207
Parameters.....	3-207
LOGUTIL FOR_LOSTMVC Statement .....	3-208
Syntax .....	3-208
Parameters.....	3-208
LOGUTIL GENAUDIT Statement.....	3-209
Syntax .....	3-209
Parameters.....	3-209
LOGUTIL LOCATE_VTV Statement .....	3-210
Syntax .....	3-210
Parameters.....	3-210
LOGUTIL UNDELETE Statement .....	3-211
Syntax .....	3-211
Parameters.....	3-211
<b>MEDVERfy</b> .....	3-212
Description .....	3-212
Syntax.....	3-212
Parameters.....	3-212
MEDVERfy Reports .....	3-214
<b>MERGEcds</b> .....	3-214
Description .....	3-214
Syntax.....	3-215
Parameters.....	3-215
Additional JCL Requirements .....	3-216
SLSMERGE Control Statement .....	3-216
Syntax .....	3-217
Parameters.....	3-217
Parameters (Real Volumes) .....	3-218
Parameters (CDS Containing Virtual Data) .....	3-218
Parameters (CDS Containing VAULT Data).....	3-218
<b>MERGMFST</b> .....	3-219
Description .....	3-219

Syntax.....	3-219
Parameters.....	3-219
Additional JCL Requirements .....	3-219
<b>METAdata</b> .....	3-220
Description .....	3-220
Syntax.....	3-220
Parameters.....	3-220
Tags Produced .....	3-220
<b>MGMTDEF</b> .....	3-221
Description .....	3-221
Syntax.....	3-222
Parameters.....	3-222
MGMTclas Control Statement.....	3-222
Syntax .....	3-223
Parameters.....	3-224
MIGRSEL Control Statement.....	3-233
Syntax .....	3-233
Parameters.....	3-233
MIGRVTV Control Statement .....	3-235
Syntax .....	3-235
Parameters.....	3-235
MVCATTR Control Statement .....	3-237
Syntax .....	3-237
Parameters.....	3-237
STORclas Control Statement .....	3-238
Syntax .....	3-238
Parameters.....	3-238
Dynamic Reclaim Usage .....	3-241
STORMNGR and ACS Parameter Interactions .....	3-242
FROMLST Parameter Usage .....	3-242
STORLST Control Statement .....	3-243
Syntax .....	3-243
Parameters.....	3-244
STORSEL Control Statement .....	3-244
Syntax .....	3-244
Parameters.....	3-245
VTSSLST Control Statement.....	3-246
Syntax .....	3-246
Parameters.....	3-246
VTSSSEL Control Statement.....	3-247
Syntax .....	3-247
Parameters.....	3-248
<b>MIGrate</b> .....	3-250
Description .....	3-250
Syntax (Format 1) .....	3-250
Parameters (Format 1) .....	3-250
Syntax (Format 2) .....	3-251

Parameters (Format 2) .....	3-252
<b>MNTD</b> .....	3-252
Description .....	3-252
Syntax.....	3-252
Parameters.....	3-253
<b>MODify</b> .....	3-257
Description .....	3-257
Syntax.....	3-258
Parameters.....	3-258
<b>Mount</b> .....	3-261
Description .....	3-261
Syntax.....	3-261
Parameters.....	3-262
To Mount a Specific Volume on a Transport .....	3-262
To Mount a Scratch Volume on a Transport.....	3-263
To Mount a VTV on a VTD and Optionally, Assign a Management Class .....	3-264
<b>MOVE</b> .....	3-264
Description .....	3-264
Syntax.....	3-264
Parameters.....	3-265
<b>MVCDRain</b> .....	3-266
Description .....	3-266
Syntax.....	3-266
Parameters.....	3-267
<b>MVCMaint</b> .....	3-268
Description .....	3-268
Syntax.....	3-269
Parameters.....	3-269
Additional JCL Requirements .....	3-271
Return Codes .....	3-272
MVCMaint Report .....	3-272
<b>MVCPLRPT</b> .....	3-273
Description .....	3-273
Syntax.....	3-273
Parameters.....	3-273
Named MVC Pool Report .....	3-274
Fields.....	3-275
Summary for Storage Class or Named MVC Pool.....	3-278
Summary of MVCs by usage.....	3-278
<b>MVCRPt</b> .....	3-278
Description .....	3-278
Syntax.....	3-278
Parameters.....	3-279
MVC Reports .....	3-280
MVC Summary Report.....	3-280
Fields.....	3-280
MVC Detailed Report.....	3-284



Fields.....	3-285
<b>OFFload</b> .....	3-286
Description .....	3-286
Syntax.....	3-286
Parameters.....	3-287
Additional JCL Requirements .....	3-287
<b>OPTION TITLE Control Statement</b> .....	3-287
Syntax.....	3-288
Parameters.....	3-288
<b>OPTion</b> .....	3-288
Description .....	3-288
Syntax.....	3-289
Parameters.....	3-289
<b>PITCOPY</b> .....	3-292
Description .....	3-292
Syntax.....	3-292
Parameters.....	3-292
Additional JCL Requirements .....	3-292
<b>RECall</b> .....	3-293
Description .....	3-293
Syntax.....	3-293
Parameters.....	3-294
<b>RECLaim</b> .....	3-295
Description .....	3-295
Syntax.....	3-295
Parameters.....	3-296
Output.....	3-298
Fields.....	3-298
<b>RECONcil</b> .....	3-299
Description .....	3-299
Syntax.....	3-300
Parameters.....	3-300
RECONcil Reports .....	3-301
RECONcil Report (MOVEVTV Not Specified).....	3-301
Fields.....	3-302
MVC Report Fields .....	3-303
RECONcil MOVEVTV Report .....	3-304
<b>RECOVer</b> .....	3-304
Description .....	3-305
Syntax.....	3-305
Parameters.....	3-305
<b>RELease</b> .....	3-305
Description .....	3-305
Syntax.....	3-306
Parameters.....	3-306
<b>REPLaceall</b> .....	3-306
Description .....	3-306

Syntax.....	3-306
Parameters.....	3-307
<b>REStore</b> .....	3-307
Description .....	3-307
Syntax.....	3-307
Parameters.....	3-308
Additional JCL Requirements .....	3-308
<b>SCRAtch</b> .....	3-308
Description .....	3-308
Syntax.....	3-308
Parameters.....	3-309
<b>SCREdist</b> .....	3-309
Description .....	3-309
Syntax.....	3-310
Parameters.....	3-310
<b>SCRPT</b> .....	3-312
Description .....	3-312
Syntax.....	3-312
Parameters.....	3-312
SCRPT Report .....	3-312
Fields.....	3-313
<b>SENter</b> .....	3-314
Description .....	3-314
Syntax.....	3-314
Parameters.....	3-314
<b>SET</b> .....	3-314
<b>SET CLNPRFX</b> .....	3-315
Description.....	3-316
Syntax .....	3-316
Parameters.....	3-316
SET CLNPRFX Procedure.....	3-316
<b>SET COMPREFX</b> .....	3-317
Description.....	3-317
Syntax .....	3-317
Parameters.....	3-317
<b>SET DRVHOST</b> .....	3-319
Description.....	3-319
Syntax .....	3-319
Parameters.....	3-319
<b>SET EJCTPAS</b> .....	3-319
Description.....	3-319
Syntax .....	3-320
Parameters.....	3-320
<b>SET EJCTSKP</b> .....	3-320
Description.....	3-320
Syntax .....	3-320
Parameters.....	3-321

SET FREEZE.....	3-321
Description.....	3-321
Syntax .....	3-321
Parameters.....	3-321
SET HOSTID .....	3-323
Description.....	3-323
Syntax .....	3-323
Parameters.....	3-323
SET HSCLEVEL.....	3-324
Description.....	3-324
Syntax .....	3-324
Parameters.....	3-324
SET LOGFILE.....	3-325
Description.....	3-325
Syntax .....	3-325
Parameters.....	3-325
SET MAJNAME.....	3-326
Description.....	3-326
Syntax .....	3-326
Parameters.....	3-326
SET MIGOPT.....	3-327
Description.....	3-327
Syntax .....	3-327
Parameters.....	3-327
SET NEWHOST .....	3-328
Description.....	3-328
Syntax .....	3-328
Parameters.....	3-328
SET RMM .....	3-329
Description.....	3-329
Syntax .....	3-329
Parameters.....	3-329
SET SCRLABL.....	3-330
Description.....	3-330
Syntax .....	3-330
Parameters.....	3-330
SET SLIDRIVS.....	3-331
Description.....	3-331
Syntax .....	3-331
Parameters.....	3-332
Running SET SLIDRIVS With the HSC Active.....	3-333
SET SLIDRIVS Procedure .....	3-334
For the 9310 and 9740:.....	3-334
For the SL3000 and SL8500:.....	3-334
SET SLISTATN .....	3-334
Description.....	3-335
Syntax .....	3-335

Parameters.....	3-335
SET SMF.....	3-335
Description.....	3-336
Syntax .....	3-336
Parameters.....	3-336
SET TAPEPlex.....	3-336
Description.....	3-336
Syntax .....	3-336
Parameters.....	3-336
SET TCHNIQE.....	3-337
Description.....	3-337
Syntax .....	3-337
Parameters.....	3-337
SET TCHNIQE Procedure .....	3-338
SET VAULT.....	3-338
Description.....	3-338
Syntax .....	3-338
Parameters.....	3-339
SET VAULTVOL.....	3-340
Description.....	3-340
Syntax .....	3-341
Parameters.....	3-341
SET VOLPARM.....	3-341
Description.....	3-341
Syntax .....	3-342
Parameters.....	3-342
Additional JCL Requirements.....	3-343
Using SET VOLPARM to Disable POOLPARM/VOLPARM.....	3-343
POOLPARM Control Statement .....	3-343
Syntax .....	3-344
Parameters .....	3-344
TYPE(SCRATCH) Subparameters.....	3-345
TYPE(CLEAN) Subparameters.....	3-345
TYPE(MVC) Subparameters .....	3-346
TYPE(EXTERNAL) Subparameters .....	3-348
VOLPARM Control Statement.....	3-348
Syntax .....	3-349
Parameters .....	3-349
SET VOLPARM UPDATE.....	3-350
Description.....	3-350
Syntax .....	3-350
Parameters.....	3-351
Additional JCL Requirements.....	3-351
POOLPARM Change Control Statement .....	3-351
Syntax .....	3-351
Parameters .....	3-352
VOLPARM Change Control Statement .....	3-352

Syntax .....	3-352
Parameters .....	3-353
SET VOLPARM JOIN .....	3-353
Description .....	3-353
Syntax .....	3-353
Parameters.....	3-354
Additional JCL Requirements .....	3-354
<b>SRVlev</b> .....	3-354
Description .....	3-354
Syntax.....	3-354
Parameters.....	3-354
<b>STOPMN</b> .....	3-355
Description .....	3-355
Syntax.....	3-355
Parameters.....	3-355
<b>SWitch</b> .....	3-356
Description .....	3-356
Syntax.....	3-357
Parameters.....	3-357
Examples .....	3-359
<b>TRace</b> .....	3-359
Description .....	3-360
Syntax.....	3-360
Parameters.....	3-360
<b>TRACELKP</b> .....	3-361
Description .....	3-361
Syntax.....	3-361
Parameters.....	3-362
<b>UEXIT</b> .....	3-362
Description .....	3-363
Syntax.....	3-363
Parameters.....	3-363
<b>UNSCratch</b> .....	3-364
Description .....	3-365
Syntax.....	3-365
Parameters.....	3-365
<b>UNSElect</b> .....	3-365
Description .....	3-365
Syntax.....	3-366
Parameters.....	3-366
<b>Vary</b> .....	3-366
Description .....	3-367
Syntax.....	3-367
Parameters.....	3-368
<b>Vlew</b> .....	3-370
Description .....	3-370
Syntax.....	3-370

Parameters.....	3-371
<b>VLEMAINT</b> .....	3-374
Description .....	3-374
Syntax.....	3-374
Parameters.....	3-375
Optional Parameters (VLEMAINT COPYMVC).....	3-376
<b>VOLPCONV</b> .....	3-376
Description .....	3-376
Syntax.....	3-376
Parameters.....	3-377
Additional JCL Requirements .....	3-377
<b>VOLRpt</b> .....	3-377
Description .....	3-378
Syntax.....	3-378
Parameters.....	3-379
Additional JCL Requirements .....	3-384
Volume Report.....	3-384
Fields.....	3-385
Total Reports.....	3-386
Volume Report Totals .....	3-386
Subpool Totals.....	3-386
<b>VTVMMAINT</b> .....	3-387
Description .....	3-387
Syntax.....	3-387
Parameters.....	3-387
Return Codes .....	3-389
VTVMMAINT Report .....	3-389
<b>VTVRPt</b> .....	3-390
VTVRPt BASIC .....	3-390
Description.....	3-391
Syntax .....	3-391
Parameters.....	3-391
VTV Report (Basic) .....	3-392
Fields.....	3-393
VTVRPt COPIES.....	3-394
Description.....	3-395
Syntax .....	3-395
Parameters.....	3-395
VTV Report (COPIES) .....	3-396
Fields.....	3-396
<b>VVAUDIT</b> .....	3-397
Description .....	3-397
Syntax.....	3-397
Parameters.....	3-398
<b>Warn</b> .....	3-398
Description .....	3-398
Syntax.....	3-398

## 4 SMC Utilities

<b>SMC Utility Return Codes and Messages</b> .....	4-1
<b>Unified User Interface Utility (SMCUUUI)</b> .....	4-2
Sample JCL .....	4-3
Parameters .....	4-3
Required Data Set Definition (DD) Statements .....	4-4
UUIIN Data Set.....	4-5
Syntax rules.....	4-5
Control Statements .....	4-5
STDOUT Data Set .....	4-7
Output Data Set Definition (DD) Statements .....	4-7
Remote UI Input and Output .....	4-8
Remote UI Input.....	4-8
Remote UI Output .....	4-8
SDD Command .....	4-8
Syntax .....	4-9
Parameters .....	4-9
CONFIG Command Example.....	4-10
IMPORT Command Example .....	4-11
EXPORT Command Example .....	4-12
MERGMST Command Example.....	4-12
<b>JES3 Configuration Report Utility (SMCUPJS)</b> .....	4-13
Sample JCL .....	4-14
Sample Reports.....	4-14
XTYPE Number to XTYPE Name (EXTON) .....	4-14
XTYPE Number to Esoteric (EXTOE).....	4-15
Esoteric to XTYPE (EETOX) .....	4-15
XTYPE to Device Number (EXTOD).....	4-16
Device Number to XTYPE (EDTOX).....	4-17
<b>POLicy and TAPEREQ Batch Test Utility (SMCUTRQ)</b> .....	4-17
Sample JCL .....	4-17
Control Cards.....	4-18
Sample Input.....	4-19
Example 1 .....	4-19
Example 2 .....	4-19
Example 3 .....	4-20
Sample Output.....	4-20
<b>Tape Management Extract Utility (SMCUDBX)</b> .....	4-21
Running the SMC Scratch Extract Utility .....	4-22
Using SMCUDBX with CA-1 Release 4.9 or Below.....	4-24
Syntax.....	4-24
Parameters .....	4-24
Date Related Parameters .....	4-25
JCL Requirements .....	4-26
Output Description .....	4-26

<b>SMC UII Simulation Utility (SMCUSIM)</b> .....	4-27
Phase 1 Processing .....	4-27
Phase 2 Processing .....	4-28
Sample JCL .....	4-28
Parameters .....	4-29
Required Data Set Definition (DD) Statements .....	4-31
UIIIN Data Set.....	4-31
Syntax Rules .....	4-31
Control Statements .....	4-31
STDOUT Data Set .....	4-32
SMCUSIM Utility Sample CSV .....	4-32
<b>Trace Format Utility (SMCUGTF)</b> .....	4-32
Sample JCL .....	4-32
<b>VSM console SMF Offload utility (SMCUSMF)</b> .....	4-33
Sample JCL .....	4-33
Parameters.....	4-33
Required Data Set Definition (DD) Statements .....	4-34
SMCUSMF Usage.....	4-34

## 5 HSC Stand-Alone Utilities

<b>Utility Administrator (SLUADMIN)</b> .....	5-1
Invoking SLUADMIN .....	5-1
SLUADMIN Execution Parameters.....	5-2
SLUADMIN Definition Statements.....	5-3
General Definition Statements .....	5-3
CDS Definition Statements .....	5-4
UII Output Definition Statements .....	5-4
Requesting SLUADMIN Output.....	5-5
Requesting XML Output.....	5-5
Requesting CSV Output.....	5-5
CSV Command.....	5-6
SLUADMIN Program Return Codes.....	5-8
Reports Created by Utility Commands.....	5-9
Report Headings .....	5-9
<b>SLICREAT</b> .....	5-9
Creating Control Data Set Initialization JCL.....	5-10
DD Statements .....	5-11
Executing the SLICREAT Program.....	5-12
SLICREAT EXEC Statement .....	5-12
Verifying Successful Completion of the SLICREAT Program.....	5-12
Calculating Cartridge Capacity - SL8500 and SL3000 .....	5-12
Using SLICREAT to Create Only the Standby CDS.....	5-13
SLICREAT EXEC Statement to Create Standby CDS .....	5-13
Using SLICREAT to Calculate DASD Space .....	5-13
<b>Scratch Conversion Utility (SLUCONDB)</b> .....	5-13
Input Formats .....	5-14
Keyword Format .....	5-14



Syntax .....	5-14
Parameters .....	5-15
Positional Format .....	5-17
Syntax .....	5-17
Parameters .....	5-17
Date Related Parameters.....	5-18
JCL Requirements .....	5-19
JCL Examples.....	5-20
Example 1 .....	5-20
Example 2 .....	5-20
Example 3 .....	5-21
Example 4 .....	5-21
Example 5 .....	5-21
Output Description .....	5-21

## 6 VTCS Stand-Alone Utilities

RTV Utility .....	6-1
Syntax.....	6-1
Parameters.....	6-2
JCL Requirements .....	6-4
RTV Utility Report Messages .....	6-4
RTV LISTONLY Listing .....	6-7
RTV Decompress Listing.....	6-8

## A MEDIA, RECTech, and MODEL Values

Media Type (MEDia).....	A-1
Recording Technique (RECTech) .....	A-5
Model Type (MODEL).....	A-8

## B Library Identification

ACSid.....	B-1
LSMid .....	B-1
CAPid.....	B-1
CAP Numbers.....	B-1
Panels, Rows and Columns .....	B-2
Panels .....	B-2
Rows.....	B-3
Columns.....	B-4

## C Tape Management System Interface (SLUDR\*) Routines

SLUDRCA1.....	C-1
SLUDRRMM.....	C-1
SLUDRTL.....	C-2
SLUDRSMC .....	C-2
SLUDRZAR.....	C-2

**Glossary**

**Index**



## List of Examples

1-1	JCL Defining Sequential PARMLIB Data Sets .....	1-7
1-2	JCL Defining Partitioned PARMLIB Data Sets.....	1-8
2-1	SIMulate Command Output - No Tape Policies Applied.....	2-73
2-2	SIMulate output - Tape Policies Applied at IDAX.....	2-73
3-1	ARCHive report (MOVEVTV not specified) .....	3-7
3-2	ARCHive MOVEVTV report.....	3-9
3-3	Audit report .....	3-15
3-4	DEComp FLATDD output.....	3-67
3-5	DEComp SLSPRINT output .....	3-67
3-6	DELETSCR report .....	3-70
3-7	Display ACS output.....	3-75
3-8	Display ACTive output (no detail).....	3-76
3-9	Display Active DETail output.....	3-76
3-10	Display Cap output .....	3-85
3-11	Display CDS output.....	3-87
3-12	Display CLink output (no parameters).....	3-88
3-13	Display CLink output (FROMvtss specified) .....	3-88
3-14	Display CLUster output .....	3-91
3-15	Display CONFIG output.....	3-95
3-16	Display LINKSto output .....	3-105
3-17	Display LOCKs output.....	3-108
3-18	Display MIGrate output.....	3-113
3-19	Display MIGrate DETail output .....	3-114
3-20	Display MIGrate AUTO output .....	3-115
3-21	Display MIGrate DELAY output.....	3-115
3-22	Display MVC output .....	3-117
3-23	Console Display MVCPool output (NAME POOL1) .....	3-124
3-24	Console Display MVCPool output (no NAME specified) .....	3-124
3-25	Console Display MVCPool SPACE output (no NAME specified) .....	3-124
3-26	Console Display MVCPool COUNTS output (no NAME specified) .....	3-125
3-27	Utility Display MVCPool output (no NAME specified) .....	3-125
3-28	Display PATH output .....	3-128
3-29	Display Active output (no detail).....	3-130
3-30	Display REPlicat output.....	3-131
3-31	Display REPlicat LISTVTVS DELAY output .....	3-132
3-32	Display RTD output.....	3-134
3-33	Display RTD output (Display RTD Queued).....	3-136
3-34	Display SCRatch output (all).....	3-139
3-35	Display SCRatch output (subpool).....	3-139
3-36	Display SCRatch output (ACS).....	3-140
3-37	Display SCRatch output (LSM).....	3-140
3-38	Display SERVER output.....	3-143
3-39	Display STORCLas output .....	3-145
3-40	Display STORMNgr output .....	3-146
3-41	Display TASKs output .....	3-148
3-42	Display THReshld command output.....	3-151
3-43	Display VSCRatch output.....	3-155
3-44	Display VTD output .....	3-157
3-45	Display VTSS output .....	3-159
3-46	Display VTSS Detail output.....	3-161
3-47	Display VTV output.....	3-164
3-48	SLSSTATN File.....	3-175
3-49	SLSVTSS File.....	3-176
3-50	INVENTORY report .....	3-201

3-51	MEDVERfy report (1 of 2) .....	3-214
3-52	MEDVERfy report (2 of 2) .....	3-214
3-53	MVCMaint Report .....	3-272
3-54	Named MVC Pool report .....	3-274
3-55	MVCRPt MVC summary report .....	3-280
3-56	MVCRPt MVC detailed report (additional fields) .....	3-284
3-57	RECLaim output .....	3-298
3-58	RECONcil report (MOVEVTV not specified) .....	3-301
3-59	RECONcil MOVEVTV report.....	3-304
3-60	SCRPT report .....	3-313
3-61	VOLRpt report.....	3-384
3-62	VTVMAINT report .....	3-389
3-63	VTVRPt BASIC report .....	3-392
3-64	VTVRPt COPIES report.....	3-396
4-1	JCL to Execute the SMCUUI Utility .....	4-3
4-2	JCL to Produce the JES3 Configuration Report .....	4-14
4-3	EXTOD report fragment.....	4-16
4-4	EDTOX report fragment.....	4-17
4-5	JCL to Execute the SMCUTRQ Utility .....	4-17
4-6	SMCUTRQ Utility Output .....	4-20
4-7	Sample JCL for Scratch Extraction Utility .....	4-26
4-8	Scratch Extraction Utility Sample Output Messages .....	4-26
4-9	JCL to Execute the SMCUSIM Utility.....	4-28
4-10	JCL to Execute the SMCUGTF Utility .....	4-32
4-11	JCL to Execute the SMCUSMF Utility.....	4-33
5-1	JCL to Invoke SLUADMIN.....	5-2
5-2	CSV Output for the MVCRPT command .....	5-8
5-3	SLICREAT JCL .....	5-10
5-4	Scratch Conversion Utility output (without LIBONLY or TAPEPLEX) .....	5-22
5-5	Scratch Conversion Utility output (with LIBONLY or TAPEPLEX).....	5-22
6-1	RTV LISTONLY Listing .....	6-7
6-2	RTV Decompress Listing .....	6-8



## List of Figures

2-1	ALLOCDef command syntax.....	2-2
2-2	ALLOCJob command syntax .....	2-8
2-3	CMDDef command syntax .....	2-10
2-4	COLlector command syntax.....	2-12
2-5	COMMtest command syntax .....	2-14
2-6	DISMOunt command syntax.....	2-16
2-7	Display DRive command syntax .....	2-17
2-8	Display POLicy command syntax .....	2-19
2-9	Display RC command syntax.....	2-19
2-10	Display SERVer command syntax.....	2-20
2-11	Display SMC command syntax.....	2-21
2-12	Display STORMNGR command syntax .....	2-22
2-13	Display TAPEPlex command syntax .....	2-23
2-14	Display Volume command syntax .....	2-24
2-15	DRIVemap command syntax .....	2-25
2-16	Help command syntax .....	2-26
2-17	HTTP command syntax .....	2-28
2-18	IDAX command syntax.....	2-30
2-19	LIMIT command syntax.....	2-33
2-20	List command syntax .....	2-36
2-21	LOG command syntax .....	2-37
2-22	METAdata command syntax.....	2-40
2-23	MONitor command syntax.....	2-42
2-24	MOunt command syntax .....	2-44
2-25	MOUNTDef command syntax.....	2-46
2-26	MSGDef command syntax.....	2-50
2-27	MSGJob command syntax .....	2-52
2-28	POLicy command syntax.....	2-56
2-29	READ command syntax.....	2-63
2-30	RESYNChronize command syntax.....	2-65
2-31	Route command syntax.....	2-66
2-32	SERVer command syntax .....	2-68
2-33	SIMulate command syntax .....	2-71
2-34	SMSDef command syntax.....	2-75
2-35	STORMNGR command syntax.....	2-78
2-36	TAPEPlex command syntax .....	2-80
2-37	TCPip command syntax.....	2-82
2-38	TRace command syntax .....	2-86
2-39	TREQDef command syntax .....	2-89
2-40	TAPEREQ control statement syntax .....	2-92
2-41	UEXit command syntax .....	2-97
2-42	UNITAttr command syntax.....	2-100
2-43	USERMsg command syntax .....	2-102
2-44	VMSG command syntax .....	2-103
2-45	XCLIENT command syntax.....	2-105
2-46	XUDB command syntax.....	2-107
3-1	ACTivities command syntax.....	3-2
3-2	ACTMVCgn command syntax.....	3-5
3-3	ARCHive command syntax.....	3-6
3-4	AUDit command syntax .....	3-11
3-5	BACKup command syntax.....	3-19
3-6	CANcel command syntax.....	3-20
3-7	CAPPref command syntax .....	3-21
3-8	CDs command syntax .....	3-24

3-9	CDSCREat command syntax.....	3-26
3-10	CSDaTa command syntax.....	3-28
3-11	CDSDEF command syntax .....	3-30
3-12	CLeaN command syntax.....	3-32
3-13	COMMPaTh command syntax .....	3-34
3-14	CONFIg command syntax .....	3-37
3-15	CONFIg CLINK statement syntax.....	3-40
3-16	CONFIg CLUSTER statement syntax .....	3-41
3-17	CONFIg GLOBAL statement syntax.....	3-43
3-18	CONFIg HOST statement syntax .....	3-50
3-19	CONFIg RECLAIM statement syntax.....	3-51
3-20	CONFIg RTDpaTh statement syntax .....	3-54
3-21	CONFIg STORMNGR statement syntax .....	3-56
3-22	CONFIg TAPEPLEX statement syntax .....	3-56
3-23	CONFIg VTD statement syntax .....	3-58
3-24	CONFIg VTSS statement syntax.....	3-59
3-25	CONSolid command syntax.....	3-63
3-26	DBSERVer command syntax .....	3-65
3-27	DEComp command syntax.....	3-66
3-28	DELETSCR command syntax.....	3-69
3-29	DIRBLD command syntax .....	3-71
3-30	DISMounT command syntax.....	3-72
3-31	Display Acs command syntax.....	3-75
3-32	Display ACTive command syntax.....	3-76
3-33	Display ALl command syntax.....	3-84
3-34	Display Cap command syntax .....	3-84
3-35	Display CDS command syntax .....	3-86
3-36	Display CLink command syntax .....	3-88
3-37	Display CLUster command syntax .....	3-90
3-38	Display CMD command syntax.....	3-93
3-39	Display COMMPaTh command syntax.....	3-94
3-40	Display CONFIg command syntax .....	3-95
3-41	Display DRives command syntax.....	3-98
3-42	Display DRIVE_INFO command syntax.....	3-101
3-43	Display EXceptns command syntax.....	3-103
3-44	Display LINKSto command syntax.....	3-105
3-45	Display LMUPDEF command syntax.....	3-107
3-46	Display LOCKS command syntax .....	3-108
3-47	Display Lsm command syntax.....	3-109
3-48	Display Message command syntax .....	3-111
3-49	Display MGMTDEF command syntax.....	3-112
3-50	Display MIGrate command syntax .....	3-112
3-51	Display MNTD command syntax.....	3-115
3-52	Display MONitor command syntax.....	3-116
3-53	Display MVC command syntax.....	3-117
3-54	Display MVCPool command syntax.....	3-123
3-55	Display OPTion command syntax.....	3-127
3-56	Display PATH command syntax .....	3-128
3-57	Display Queue command syntax.....	3-130
3-58	Display REPLICat command syntax .....	3-131
3-59	Display Requests command syntax .....	3-133
3-60	Display RTD command syntax .....	3-133
3-61	Display SCRatch command syntax .....	3-137
3-62	Display SEN command syntax .....	3-141
3-63	Display SERVER command syntax .....	3-142



3-64	Display SRVlev command syntax .....	3-143
3-65	Display Status command syntax.....	3-144
3-66	Display STORCLas command syntax .....	3-145
3-67	Display STORMNgr command syntax .....	3-146
3-68	Display TASKs command syntax .....	3-148
3-69	Display THReshld command syntax .....	3-150
3-70	Display Volser output example .....	3-152
3-71	Display VOLume_Info command syntax.....	3-154
3-72	Display VSCRatch command syntax .....	3-155
3-73	Display VTD command syntax .....	3-156
3-74	Display VTSS command syntax.....	3-159
3-75	Display VTV command syntax .....	3-164
3-76	DRAin command syntax.....	3-168
3-77	DRCHKPT command syntax .....	3-169
3-78	DRCLEAN command syntax .....	3-169
3-79	DRMONitr command syntax .....	3-171
3-80	DRTEST CREATE command syntax.....	3-174
3-81	DRTEST PRIMEprd syntax.....	3-177
3-82	DRTEST RESET command syntax.....	3-179
3-83	DRTEST START command syntax .....	3-179
3-84	DRTEST STOP command syntax.....	3-180
3-85	EEXPORT command syntax.....	3-181
3-86	Eject command syntax .....	3-184
3-87	ENter command syntax.....	3-188
3-88	EXECParm command syntax .....	3-189
3-89	EXPORT command syntax .....	3-191
3-90	FMTLOG command syntax .....	3-193
3-91	IMPORT command syntax .....	3-194
3-92	INITialize command syntax .....	3-197
3-93	INVENTORY command syntax .....	3-199
3-94	LIBGen command syntax.....	3-202
3-95	LMUPDEF command syntax.....	3-204
3-96	LMUPATH control statement syntax .....	3-205
3-97	LOGUTIL command syntax .....	3-207
3-98	LOGUTIL FOR_LOSTMVC statement syntax .....	3-208
3-99	LOGUTIL GENAUDIT statement syntax.....	3-209
3-100	LOGUTIL LOCATE_VTV statement syntax .....	3-210
3-101	LOGUTIL UNDELETE statement syntax.....	3-211
3-102	MEDVERfy command syntax .....	3-212
3-103	MERGEcds command syntax.....	3-215
3-104	SLSMERGE control statement syntax .....	3-217
3-105	MERGMFST command syntax.....	3-219
3-106	METAdata command syntax.....	3-220
3-107	MGMTDEF command syntax .....	3-222
3-108	MGMTclas control statement syntax .....	3-224
3-109	MIGRSEL control statement syntax .....	3-233
3-110	MIGRVTV control statement syntax .....	3-235
3-111	MVCATTR control statement syntax.....	3-237
3-112	STORclas control statement syntax .....	3-238
3-113	VLE to VLE Replication (Versus VTSS to VLE).....	3-243
3-114	STORLST control statement syntax.....	3-244
3-115	STORSEL control statement syntax.....	3-245
3-116	VTSSLST control statement syntax.....	3-246
3-117	VTSSSEL control statement syntax.....	3-248
3-118	MIGrate command syntax (Format 1).....	3-250

3-119	MIGrate command syntax (Format 2).....	3-251
3-120	MNTD command syntax .....	3-253
3-121	MODify command syntax .....	3-258
3-122	MouNt command syntax .....	3-262
3-123	MOVe command syntax .....	3-265
3-124	MVCDRain command syntax .....	3-267
3-125	MVCMaINT command syntax.....	3-269
3-126	MVCPLRPT command syntax .....	3-273
3-127	MVCRPt command syntax .....	3-279
3-128	OFFload command syntax.....	3-287
3-129	OPTION TITLE control statement syntax .....	3-288
3-130	OPTion command syntax .....	3-289
3-131	PITCOPY command syntax.....	3-292
3-132	RECall command syntax.....	3-294
3-133	RECLaim command syntax .....	3-296
3-134	RECONcil command syntax.....	3-300
3-135	RECOVer command syntax.....	3-305
3-136	RELease command syntax .....	3-306
3-137	REPLaceall command syntax .....	3-307
3-138	REStore command syntax.....	3-307
3-139	SCRAtch command syntax .....	3-308
3-140	SCREdist command syntax .....	3-310
3-141	SCRPT command syntax.....	3-312
3-142	SEnTer command syntax .....	3-314
3-143	SEt CLNPRFX command syntax.....	3-316
3-144	SEt COMPRFX command syntax .....	3-317
3-145	SEt DRVHOST command syntax .....	3-319
3-146	SEt EJCTPAS command syntax .....	3-320
3-147	SEt EJCTSKP command syntax.....	3-321
3-148	SEt FREEZE command syntax .....	3-321
3-149	SEt HOSTID command syntax.....	3-323
3-150	SEt HSCLEVeL command syntax .....	3-324
3-151	SEt LOGFILE command syntax .....	3-325
3-152	SEt MAJNAME command syntax .....	3-326
3-153	SEt MIGOPT command syntax.....	3-327
3-154	SEt NEWHOST command syntax .....	3-328
3-155	SEt RMM command syntax.....	3-329
3-156	SEt SCRLABL command syntax .....	3-330
3-157	SEt SLIDRIVS command syntax .....	3-331
3-158	SEt SLISTATN command syntax.....	3-335
3-159	SEt SMF command syntax .....	3-336
3-160	SEt TAPEPlex command syntax .....	3-336
3-161	SEt TCHNIQE command syntax .....	3-337
3-162	SEt VAULT command syntax .....	3-339
3-163	SEt VAULTVOL command syntax.....	3-341
3-164	SEt VOLPARM command syntax.....	3-342
3-165	POOLPARM control Statement syntax.....	3-344
3-166	VOLPARM Control Statement syntax .....	3-349
3-167	SEt VOLPARM UPDATE command syntax .....	3-350
3-168	POOLPARM Change control statement syntax .....	3-352
3-169	VOLPARM Change control statement syntax .....	3-353
3-170	SEt VOLPARM JOIN command syntax .....	3-354
3-171	SRVlev command syntax .....	3-354
3-172	STOPMN command syntax.....	3-355
3-173	SWitch command syntax .....	3-357

3-174	SL8500 Cluster Library IDs.....	3-358
3-175	TRace command syntax .....	3-360
3-176	TRACELKP command syntax.....	3-362
3-177	UEXIT command syntax .....	3-363
3-178	UNSCratch command syntax.....	3-365
3-179	UNSElect command syntax.....	3-366
3-180	Vary command syntax .....	3-367
3-181	Vliew command syntax.....	3-371
3-182	VLEMAINT command syntax .....	3-375
3-183	VOLPCONV command syntax .....	3-377
3-184	VOLRpt command syntax .....	3-378
3-185	VTVMAINT command syntax.....	3-387
3-186	VTVRpt BASIC syntax .....	3-391
3-187	VTVRpt COPIES syntax .....	3-395
3-188	VVAUDIT command syntax .....	3-397
3-189	Warn command syntax .....	3-399
4-1	SDD command syntax.....	4-9
4-2	Scratch Synchronization Between SMC Client Host and HSC CDS .....	4-22
4-3	SMCUDBX PARM parameter syntax.....	4-24
5-1	CSV command syntax .....	5-6
5-2	SLUCONDB PARM parameter syntax (keyword format).....	5-15
5-3	SLUCONDB PARM parameter syntax (positional format).....	5-17
6-1	RTV utility syntax .....	6-1

## List of Tables

3-1	Display Options.....	3-73
3-2	Display EXceptns Error Messages .....	3-103
3-3	HSC SEN Events .....	3-141
3-4	VTCS SEN Events .....	3-142
3-5	DRTEST Options .....	3-172
3-6	LIBGen Output Labels.....	3-203
3-7	MGMTclas ACSlist/DUPlex Scenarios .....	3-227
3-8	IMMDELAY and IMMEdmig Equivalent Values .....	3-229
3-9	Valid MVC Media Names and Compatible SWAPTO RTD Device Types.....	3-237
3-10	STORSEL Functions.....	3-245
3-11	VTSSSEL Functions.....	3-248
3-12	Valid Wildcards.....	3-250
3-13	SET Options .....	3-315
3-14	Mapping of Command Prefix Codes to Characters.....	3-318
3-15	HSC State/SET SLIDRIVS Operation .....	3-334
3-16	LSM ID and Library ID .....	3-358
3-17	TRace Components .....	3-360
3-18	TRACELKP Lookup Names .....	3-362
3-19	VTVRPt Options.....	3-390
4-1	SLUADMIN Return Codes .....	4-1
4-2	Control Card 1 Information.....	4-18
4-3	Control Card 2 Information.....	4-18
4-4	Control Card 3 Information.....	4-18
4-5	Tape Management Subsystem Extract File Format.....	4-23
6-1	RTV Label Types .....	6-3
A-1	Media Types.....	A-2
A-2	Media Type Defaults .....	A-4
A-3	Recording Techniques .....	A-5
A-4	Recording Technique Defaults.....	A-7
A-5	Model Types .....	A-8

---

---

# Preface

This publication describes commands, control statements, and utilities provided by ELS base software products, SMC, HSC, VTCS and CDRT.

This software solution consists of the following software:

Base software:

- Oracle's StorageTek Storage Management Component (SMC)  
(includes the product formerly known as StorageTek HTTP Server)
- Oracle's StorageTek Host Software Component (HSC)
- Oracle's StorageTek Virtual Tape Control Software (VTCS)
- Oracle's StorageTek Concurrent Disaster Recovery Test (CDRT)

Additional supportive software:

- Oracle's StorageTek Library Content Manager (LCM). LCM includes an enhanced version of the product formerly known as Offsite Vault Feature.
- Oracle's StorageTek Client System Component for MVS Environments (MVS/CSC)
- Oracle's StorageTek LibraryStation

## Audience

This document is intended for storage administrators, system programmers and operators responsible for configuring and maintaining ELS.

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

- z/OS operating system
- JES2 or JES3
- Enterprise Library Software (ELS)

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

## Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

## Related Documents

Visit the Oracle Technical Network (OTN) at the following URL to access related documentation for StorageTek libraries, tape drives, and associated software and hardware:

<http://docs.oracle.com>

## Conventions

The following text conventions are used in this document:

### Typographic Conventions

Typographic conventions include the following:

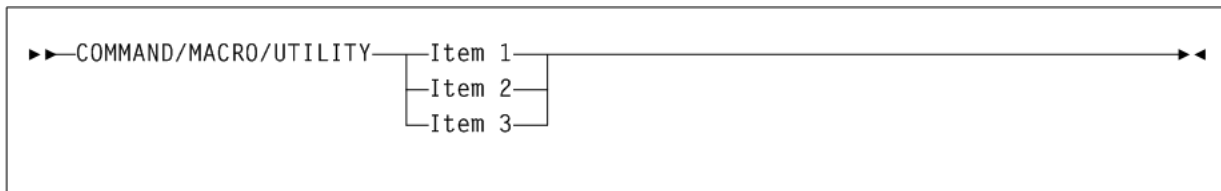
Convention	Meaning
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

### Syntax Conventions

Syntax flow diagramming conventions include the following:

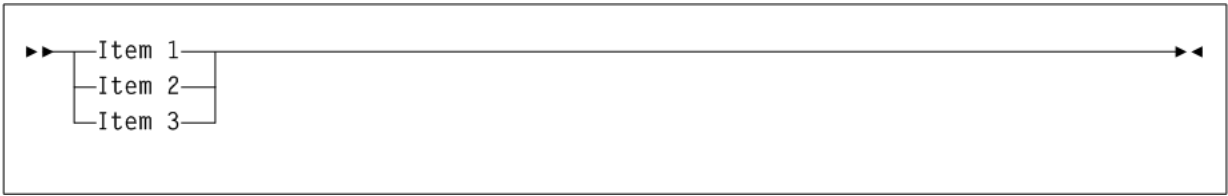
#### Flow Lines

Syntax diagrams consist of a horizontal base line, horizontal and vertical branch lines, and the text for a command, control statement, macro, or utility. Diagrams are read left to right, and top to bottom. Arrows indicate flow and direction. For example:



#### Single Required Choice

Branch lines (without repeat arrows) indicate that a single choice must be made. If one of the items to choose from is positioned on the baseline of the diagram, one item must be selected. For example:



### Single Optional Choice

If the first item is positioned on the line below the baseline, one item may be optionally selected. For example:



### Defaults

Default values and parameters appear above the baseline. For example:



Some keyword parameters provide a choice of values in a stack. When the stack contains a default value, the keyword and the value choices are placed below the baseline to indicate that they are optional, and the default value appears above the keyword line. For example:



### Repeat

A repeat symbol indicates that more than one choice can be made or that a single choice can be made more than once. The following example indicates that a comma is required as the repeat delimiter. For example:



## Keywords

All command keywords are shown in all upper case or in mixed case. When commands are not case sensitive, mixed case implies that the lowercase letters may be omitted to form an abbreviation.

## Variables

Italic type indicates a variable.

## Alternatives

A bar ( | ) separates alternative parameter values.

## Optional

Brackets [ ] indicate that a command parameter is optional.

## Delimiters

If a comma (,), a semicolon (;), or other delimiter is shown with an element of the syntax diagram, it must be entered as part of the statement.

## Ranges

An inclusive range is indicated by a pair of elements of the same length and data type, joined by a dash. The first element must be strictly less than the second element.

A hexadecimal range consists of a pair of hexadecimal numbers (for example, 0A2-0AD, or 000-0FC).

A decimal range consists of a pair of decimal numbers (that is, 1-9, or 010-094). Leading zeros are not required. The decimal portion is an incremental range. The character positions of the incremental portion of both range elements must match, and the non incremental characters of the first element must be identical to those of the second element.

A numeric VOLSER range (vol-range) consists of a pair of VOLSER elements containing a decimal numeric portion of 1 to 6 digits (for example, ABC012-ABC025, or X123CB-X277CB). The decimal portion is an incremental range. The following additional restrictions apply:

- The character positions of the incremental portion of both range elements must match.
- The non incremental characters of the first element must be identical to those of the second element.
- You cannot increment two portions of a range element. If 111AAA is the first element, you cannot specify 112AAB for the second element.
- If a VOLSER range contains more than one decimal portion, any portion is valid as the incremental range. For example:
  - A00B00 - The largest range that can be specified is A00B00 through A99B99.
  - A0B0CC - The largest range that can be specified is A0B0CC through A9B9CC.
  - 000XXX - The largest range that can be specified is 000XXX through 999XXX.

An alphabetic VOLSER range (vol-range) consists of a pair of VOLSER elements containing an incremental portion of 1 to 6 characters (for example, 000AAA-000ZZZ, or 9AAA55-9ZZZ55). This portion is an incremental range. The following additional restrictions apply:



- The character positions of the incremental portion of both range elements must match.
- The non incremental characters of the first element must be identical to those of the second element.
- You cannot increment two portions of a range element. If 111AAA is the first element, you cannot specify 112AAB for the second element.
- The alphabetic portion of the VOLSER range is defined as being from character A to Z. To increment multi-character sequences, each character increments to Z. For instance, ACZ is part of the AAA-AMM range. Examples are:
  - A00A0-A99A0  
increments VOLSERS A00A0 through A09A0, then A10A0 through A99A0.
  - 9AA9A-9ZZ9A  
increments VOLSERS 9AA9A through 9AZ9A, then 9BA9A through 9ZZ9A.
  - 111AAA-111ZZZ  
increments VOLSERS 111AAA through 111AAZ, then 111ABA through 111ZZZ
  - 999AM8-999CM8  
increments VOLSERS 999AM8 through 999AZ8, then 999BA8 through 999CM8
  - A3BZZ9-A3CDE9  
increments VOLSERS A3BZZ9 through A3CAA9, then A3CAB9 through A3CDE9
  - AAAAAA-AAACCC  
increments VOLSERS AAAAAA through AAAAAZ, then AAAABA through AAACCC
  - CCNNN-DDDNNN  
increments VOLSERS CCNNN through CCNNZ, then CCCNOA through DDDNNN. This is a very large range.

The number of volumes in an alphabetic VOLSER range depends on the number of elements in the incrementing portion of the VOLSER range. 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.

- A-Z is equivalent to  $26^1$  or 26 volumes.
- AA-ZZ is equivalent to  $26^2$  or 676 volumes.
- AAA-ZZZ is equivalent to  $26^3$  or 17,576 volumes.
- AAAA-ZZZZ is equivalent to  $26^4$  or 456,976 volumes.
- AAAAA-ZZZZZ is equivalent to  $26^5$  or 11,881,376 volumes.
- AAAAAA-ZZZZZZ is equivalent to  $26^6$  or 308,915,776 volumes.

### Lists

A list consists of one or more elements. If more than one element is specified, the elements must be separated by a comma or a blank space, and the entire list must be enclosed in parentheses.

## **Blanks**

Keyword parameters and values may be separated by any number of blanks.

## **Control Statement Conventions**

The standard syntax conventions for control statements are as follows:

- The only valid control statement information area is from column 1 to column 72. Columns 73-80 are ignored.
- Parameters may be separated by one or more blanks or a comma.
- A value is associated with a parameter by an equal (=) sign or by enclosing the value in parentheses, and concatenating it immediately after the parameter.
- Case (upper or lower) is ignored in actual control statements.
- Continuations are supported by including a plus (+) sign at the end of the line to be continued. A control statement is terminated if the statement is not continued.
- Use /\* and \*/ to enclose comments in the job stream. HSC PARMLIB members and definition data sets must specify comments in this format.
  - A comment is not required as the first control statement of any PARMLIB member.
  - Comments can be continued over multiple lines, but cannot be nested.
- The maximum length for any control statement is 1024 characters.

---

---

## What's New?

This revision includes the following updates:

- The SMC `ALLOCDef` command includes the `VTVINValidmsg` parameter, used to specify whether SMC automatically fails a job step during allocation under certain conditions:
  - Specific volume request.
  - Volume is unknown to HSC or VTCS.
  - Volser does not match any `VOLATTR` or `VOLPARM` statements.
  - The incoming eligible device list contains only virtual devices.
- Added the VTV pinning feature, used to guarantee creation of a resident VTV copy in the VTSS buffer. This feature addresses the issue of VTV copies being deleted by VTCS from the VTSS buffers once they are fully migrated.
  - The `MGMTclas` control statement includes the `PINPOL` parameter, used to specify one or two pinning locations. A location can be either a VTSS name or VTSSLST name. See [MGMTclas Control Statement](#).
  - `Display VTV` output is updated to indicate when a VTV has been pinned. See [Display VTV](#).
  - Auto migration or Migrate to threshold behavior is updated to react to pinned VTVs the same as VTVs with a `NOMIGRAT` policy on their management class. If a request is made to migrate to a particular threshold, and all remaining VTVs in the buffer are pinned, then the migration fails and error messages SLS6702E and SLS6667I are issued.
- Added Oracle recommendation to avoid draining an MVC on a `VOLSAFE (WORM)` tape, as once it is drained, it cannot be used for migration and must be replaced.  
See:
  - [Display MVC](#)
  - [MVCDRain](#)
  - [MVCRPt](#)
- The `MGMTclas` control statement includes the `REPDELAY` parameter, used to specify the asynchronous replication and/or asynchronous electronic export delay time, the minimum amount of time after a VTV is dismounted that the asynchronous replication and/or asynchronous electronic export will be delayed.  
See [MGMTclas Control Statement](#).

- The `Display REPLICat` command, `Listvtvs` parameter includes the `DELAY` option, used to list VTVs for the specified VTSS that are being delayed for both asynchronous replication and asynchronous electronic export. Output also indicates the minimum and maximum age of the delayed VTVs.

See [Display REPLICat](#).

- Removed `SSYS` and `TAPEplex` parameters from the `SMCUSIM` utility, as they are invalid.

See "[SMC UUI Simulation Utility \(SMCUSIM\)](#)".

- General maintenance updates to address code fixes.

---

# ELS Command Interfaces

This chapter describes the various interfaces you can use to issue ELS commands. It is organized in the following sections:

- Unified User Interface (UII) Support
- SMC Command Interfaces
- HSC and VTCS Command Interfaces

## Unified User Interface (UII) Support

Certain SMC, HSC, and VTCS commands described in this publication include support for the ELS UII (Unified User Interface).

The UII interface is designed to standardize both internal and external interfaces into SMC, HSC and VTCS functions. The UII enables you to do the following:

- Execute commands via a utility.
- Invoke commands from a programmatic interface.
- Request output in text, XML or Comma Separated Values (CSV) format.
- Invoke commands to an HSC server from a remote client.

ELS expands the commands available through the UII interface to include all functions previously supported by the HSC `PGMI` and batch API interfaces. Although the existing interfaces will continue to be supported for an undetermined number of future releases, these interfaces will not be enhanced to add new data items.

---

---

**Note:**

- Refer to the *ELS Programming Reference* for detailed information about the UII.
  - See "[Unified User Interface Utility \(SMCUUI\)](#)" on page 4-2 for information about the `SMCUUI` utility, used to issue UII supported SMC, HSC, and VTCS commands from a batch job.
- 
- 

## SMC Command Interfaces

This section includes the following topics:

- Issuing SMC Commands from the Console
- Specifying SMC Commands in the `SMCCMDS` or `SMCPARMS` data sets

- SMC Commands that Specify JOBname, STEPname, and PROCstep
- Issuing SMC Commands from a Utility

## Issuing SMC Commands from the Console

This section describes how to issue SMC commands from the console.

### Issuing Commands Using an SMC Command Prefix

Use an SMC command prefix to issue a command from the SMC console.

To define your prefix, specify the SMC `CMDDef` command with the `PREFIX` parameter in the `SMCPARMS` data set.

In the following example, the command prefix is defined as `B@F$`:

```
CMDDEF PREFIX(B@F$)
```

---

---

**Note:** The `PREFIX` parameter can only be specified in the `SMCPARMS` data set. See "COLlector" on page 2-11 for more information about the `CMDDef` command.

---

---

Use the following format to issue an SMC command using a command prefix:

```
PREFIXcommand-name [parameter]
```

- `PREFIX` is the command prefix.
- *command-name* indicates a SMC operator command.
- *parameter* indicates an optional or required command parameter.

The command must appear immediately following (concatenated to) the prefix character.

A null character can be specified as the command prefix character. In this case, you must use the MVS `Modify` command to issue commands. See "Issuing SMC Commands Using the MVS Modify Command" on page 1-2.

### Issuing SMC Commands Using the MVS Modify Command

Use the following command format to issue a SMC command from the MVS console using the MVS `Modify (F)` command:

```
F started-task-name,command-name [parameter]
```

- `F` indicates the MVS Modify command
- *started-task-name* indicates the SMC started task name.
- *command-name* indicates a SMC operator command.
- *parameter* indicates an optional or required command parameter.

---

---

**Note:**

- The started task name and command name must be separated with a comma. Spaces are not allowed between the subsystem name and command name.
- Parameters and values may be separated with any number of spaces, and may optionally include an equal (=) sign. Values may also be included in parentheses.

The following are equivalent commands:

```
F SMC1MVS,MSGDEF LVL 4
F SMC1MVS,MSGDEF LVL=4
F SMC1MVS,MSGDEF LVL(4)
```

---

---

**Keyword Parameters**

Keyword parameters that require or allow a user-supplied value can be specified in either of following formats:

- Enclosing the user-supplied value(s) in parentheses and concatenating it to the keyword. For example:

```
MINLVL(minlevel)
```

- Concatenating the user-supplied value(s) to the keyword with an equal sign. For example:

```
MINLVL=minlevel
```

---

---

**Note:**

- Unless otherwise specified, a list of user-supplied values must be enclosed in parentheses. See ["Lists"](#) for more information about specifying lists.
  - Except as noted, parameters can be separated by a comma or a blank. Consecutive blanks following either of these delimiters are ignored.
  - Commands and parameters can be entered in any combination of uppercase and lowercase letters.
  - Command replies are always sent to the issuing console and, in some cases such as mounts and dismounts, are routed to other consoles such as the tape library console and tape pool console.
- 
- 

**Specifying SMC Commands in the SMCCMDS or SMCPARMS Data Sets**

SMC operator commands specified in the *SMCPARMS* or *SMCCMDS* data set are automatically processed at startup.

The *SMCCMDS* data set specifies user-configured settings that can be changed while the SMC is active. Issue the *READ* command from the console to reprocess the *SMCCMDS* data set at any time.

The following is a sample *SMCCMDS* member entry:

```
MSGDEF CASE(MIXED) LVL(4)
TAPEPLEX NAME(HSCPLEX) LOCSUB(HSC0)
READ DSN('MY.PARMLIB(POL)')
TREQDEF DSN('MY.PARMLIB(TREQ)')
RESYNCHRONIZE
```

The SMCPARMS data set specifies user-configured items that cannot be changed while the SMC is active. SMCPARMS cannot be reprocessed using the READ command.

The following is a sample SMCPARMS member entry:

```
CMDDEF PREFIX(B@F$)
```

---

---

**Note:**

- It is recommended that you include only the CMDDEF PREFIX and USERMSG ID parameter settings in the SMCPARMS data set. Specify all other commands in the SMCCMDS data set.
  - See "[Control Statement Conventions](#)" for syntax conventions used when specifying commands in the SMCPARMS or SMCCMDS data set.
- 
- 

## SMC Commands that Specify JOBname, STEPname, and PROCstep

SMC ALLOCJOB, MSGJOB, and TRACE commands allow specification of JOBname, STEPname, or PROCstep. These commands are evaluated by the SMC in order of most specific to least specific job name specification. Therefore, commands can be entered in any order. Consider the following example:

Two ALLOCJOB commands are entered:

```
ALLOCJOB JOBNAME=NOALLOC* MINLVL=4
ALLOCJOB JOBNAME=NOALLOC1 MINLVL=3
```

Regardless of the order in which these commands are entered, job name NOALLOC1 is processed with MINLVL 3 because this command's job name is more specific than job name NOALLOC\*.

---

---

**Note:**

- If an SMC ALLOCJOB or MSGJOB command is entered specifying only JOBname, STEPname, or PROCstep, the command is interpreted as if LIST had also been specified.
  - The above behavior does not apply to TAPEREQ statements, which are always evaluated in the order in which they appear in the TREQDEF DD.
- 
- 

## Issuing SMC Commands from a Utility

Use the SMCUUUI utility to issue UI-supported SMC, HSC, and VTCS commands in a batch job. These commands can be routed to a local or remote HSC TapePlex.

The SMCUUUI utility can generate several types of output, including plain text, structured XML, and Comma Separated Values (CSV).

See "[Unified User Interface Utility \(SMCUUUI\)](#)" on page 4-2 for more information about the SMCUUUI utility.



---



---

**Note:** SMC commands cannot be issued from the utility administrator (SLUADMIN).

---



---

## HSC and VTCS Command Interfaces

This section includes the following topics:

- Issuing HSC and VTCS Commands from the Console
- Issuing HSC and VTCS Commands from a Utility
- Issuing HSC and VTCS Commands from PARMLIB
- HSC Service Levels

### Issuing HSC and VTCS Commands from the Console

Use any of the following methods to issue HSC and VTCS operator commands:

- Issue commands using the MVS `Modify` command
- Issue commands using a command prefix

#### Issuing HSC and VTCS Commands Using a Command Prefix

Use a command prefix to issue a HSC or VTCS command from the console. This prefix is assigned during the `LIBGEN` process. Examples include `.,` and `#`.

Use the `SET COMPRFX` command to define a new prefix. This command specifies the 2-character hexadecimal code of the command prefix.

- The command prefix is not valid until the HSC or VTCS system is recycled.
- See [Table 3-14](#) under the `SET COMPRFX` command for a list of characters associated with each code.
- See "[SET COMPRFX](#)" on page 3-317 for more information about the `SET COMPRFX` command and prefix restrictions.

Use the following format to issue a command using a command prefix:

```
PREFIXcommand-name [parameter]
```

- `PREFIX` is the command prefix.
- `command-name` indicates a HSC or VTCS operator command.
- `parameter` indicates an optional or required command parameter.

The command must appear immediately following (concatenated to) the prefix character.

A null character can be specified as the command prefix character. In this case, you must use the MVS `Modify` command to issue commands.

---



---

**Note:** The `VT` command prefix is no longer required for VTCS commands. If entered, it is ignored.

---



---

#### Issuing HSC and VTCS Commands Using the MVS Modify Command

Use the following format to issue a command from the console using the MVS `Modify` (`F`) command:

F *started-task-name,command-name* [parameter]

- F indicates the MVS `Modify` command
- *started-task-name* indicates the entry in the subsystem name table for the HSC.

A system programmer specifies this one- to four-character HSC subsystem name by adding an element to the `IEFSSNxx` entry in the `SYS1.PARMLIB`. For example, `SLS0`).

- *command-name* indicates a HSC or VTCS operator command.
- *parameter* indicates an optional or required command parameter.

### Keyword Parameters

Keyword parameters that require or allow a user-supplied value can be specified in either of following formats:

- Enclosing the user-supplied value(s) in parentheses and concatenating it to the keyword. For example:

```
HOSTID(host-id)
```

- Concatenating the user-supplied value(s) to the keyword with an equal sign. For example:

```
HOSTID=host-id
```

---

---

#### Note:

- Unless otherwise specified, a list of user-supplied values must be enclosed in parentheses. See "[Lists](#)" for more information about specifying lists.
  - Except as noted, parameters can be separated by a comma or a blank. Consecutive blanks following either of these delimiters are ignored.
  - Commands and parameters can be entered in any combination of uppercase and lowercase letters.
  - Command replies are always sent to the issuing console and, in some cases such as mounts and dismounts, are routed to other consoles such as the tape library console and tape pool console.
- 
- 

## Issuing HSC and VTCS Commands from a Utility

HSC and VTCS contain utility functions designed to help you manage library resources. Certain HSC and VTCS commands are used to initiate these utilities. These commands are specified using the utility administrator (`SLUADMIN`).

Additionally, you can use the `SMCUUUI` utility to issue UI-supported SMC, HSC, and VTCS commands in a batch job. These commands can be routed to a local or remote HSC TapePlex.

Both `SLUADMIN` and `SMCUUUI` can generate several types of output, including plain text, structured XML, and Comma Separated Values (CSV).

**Note:**

- SWSADMIN and SWUADMIN are alias names for SLUADMIN, and are still honored.
- See "Utility Administrator (SLUADMIN)" on page 5-1 for more information about the SLUADMIN utility.
- See "Unified User Interface Utility (SMCUUUI)" on page 4-2 for more information about the SMCUUUI utility.

## Issuing HSC and VTCS Commands from PARMLIB

Certain HSC/VTCS commands can be defined in a user-defined sequential data set or single partitioned data set (PDS) member that is processed when HSC is initialized. These commands enable you to tailor HSC startup options to meet your requirements.

Unless otherwise specified, options defined in this user-defined PARMLIB data set can be dynamically changed during normal operations by issuing the corresponding operator command from the console.

PARMLIB commands are invoked by specifying MEMBER (xx) or M (xx) on the input parameter in the startup procedure. The suffix "xx" is concatenated with SLSSYS to form a name. The name is used as the ddname in the startup procedure.

- If the data set named in the SLSSYSxx DD is a partitioned data set (PDS), you MUST specify the member name containing the commands.
- If the SLSSYSxx DD is a sequential data set, only the data set name (DSN) needs to be specified.
- If the ddname is not specified in the startup procedure, then the SYS1.PARMLIB data set is dynamically allocated and a search for the member name is made in that data set.

When HSC is initialized, it reads the PARMLIB data set to obtain its operating parameters.

---

**Note:** Although you can define PARMLIB commands in SYS1.PARMLIB, it is recommended that a user-defined data set, other than SYS1.PARMLIB, be used.

---

The following JCL examples for HSC execution contain DD statements defining data sets and members that contain PARMLIB command definitions:

### Example 1-1 JCL Defining Sequential PARMLIB Data Sets

```
//IEFPROC EXEC PGM=SLSBINIT,
// TIME=1440,
// REGION=2000K,
// DPRTY=(7,5),
// PARM='E(E086) F(23) M(00)'
//*
//STEPLIB DD DISP=SHR,DSN=your.sea.SEALINK
//*
//SLSSYS00 DD DISP=SHR,DSN=parmlib0data set
//SLSSYS01 DD DISP=SHR,DSN=parmlib1data set
//SLSSYS02 DD DISP=SHR,DSN=parmlib2data set
```

**Example 1–2 JCL Defining Partitioned PARMLIB Data Sets**

```
//IEFPROC EXEC PGM=SLSBINIT,  
// TIME=1440,  
// REGION=2000K,  
// DPRTY=(7,5),  
// PARM='E(E086) F(23) M(00) '  
//*  
//STEPLIB DD DISP=SHR,DSN=your.sea.SEALINK  
//*  
//SLSSYS00 DD DISP=SHR,DSN=parmlib0data set(member)  
//SLSSYS01 DD DISP=SHR,DSN=parmlib1data set(member)  
//SLSSYS02 DD DISP=SHR,DSN=parmlib2data set(member)
```

---

---

**Note:**

- At HSC startup, a specific PARMLIB member can be specified at your discretion by specifying the 'M(nn)' startup parameter.
  - Sample SLSSYSxx command streams and PARMLIB commands are contained in member SLSSYS00 of the ELS SAMPLIB.
- 
- 

## HSC Service Levels

The HSC subsystem can operate at either **BASE** or **FULL** service level.

### Base Service Level Functions

The **BASE** service level is the nucleus of the HSC subsystem. It provides the functions necessary to execute as an extension of the operating system, and satisfies the requirements defined by the operating environment in place at the time of execution.

All HSC commands can be issued with the HSC executing at the **BASE** service level. However, commands that involve library hardware cannot perform their function completely.

VTCS commands cannot be issued with the HSC executing at the **BASE** service level.

### Mount Requests Intercepted During Base Service Level Operations

Mount messages intercepted by the SMC while the HSC is operating at the **BASE** service level are not sent to the HSC, but are left pending until the HSC reaches the **FULL** service level.

When the SMC recognizes that the HSC has reached the **FULL** service level, the mounts are redriven. These mount messages that occur with SMC intervention, and that are subsequently redriven when the HSC reaches the **FULL** service level, have their subpool specification honored.

### Full Service Level Functions

The **FULL** service level of operation for the HSC provides all of the functions available and necessary to invoke and sustain complete library operations. These functions include:

- mount/dismount processing
- CAP processing

- cartridge and cell inventory management
- LMU access
- library resource recovery
- support for all library utilities
- support for all HSC and VTCS commands

---

---

**Note:** All VTCS commands require a HSC executing at FULL service level.

---

---



---

## SMC Commands, Utilities, and Control Statements

This chapter contains reference information about SMC commands and control statements.

See [Chapter 4, "SMC Utilities"](#) for information about SMC utilities.

### ALLOCDef

#### Interfaces:

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (No XML/CSV output)

#### Subsystem Requirements:

Active SMC required, or may be input to the SMCUSIM utility

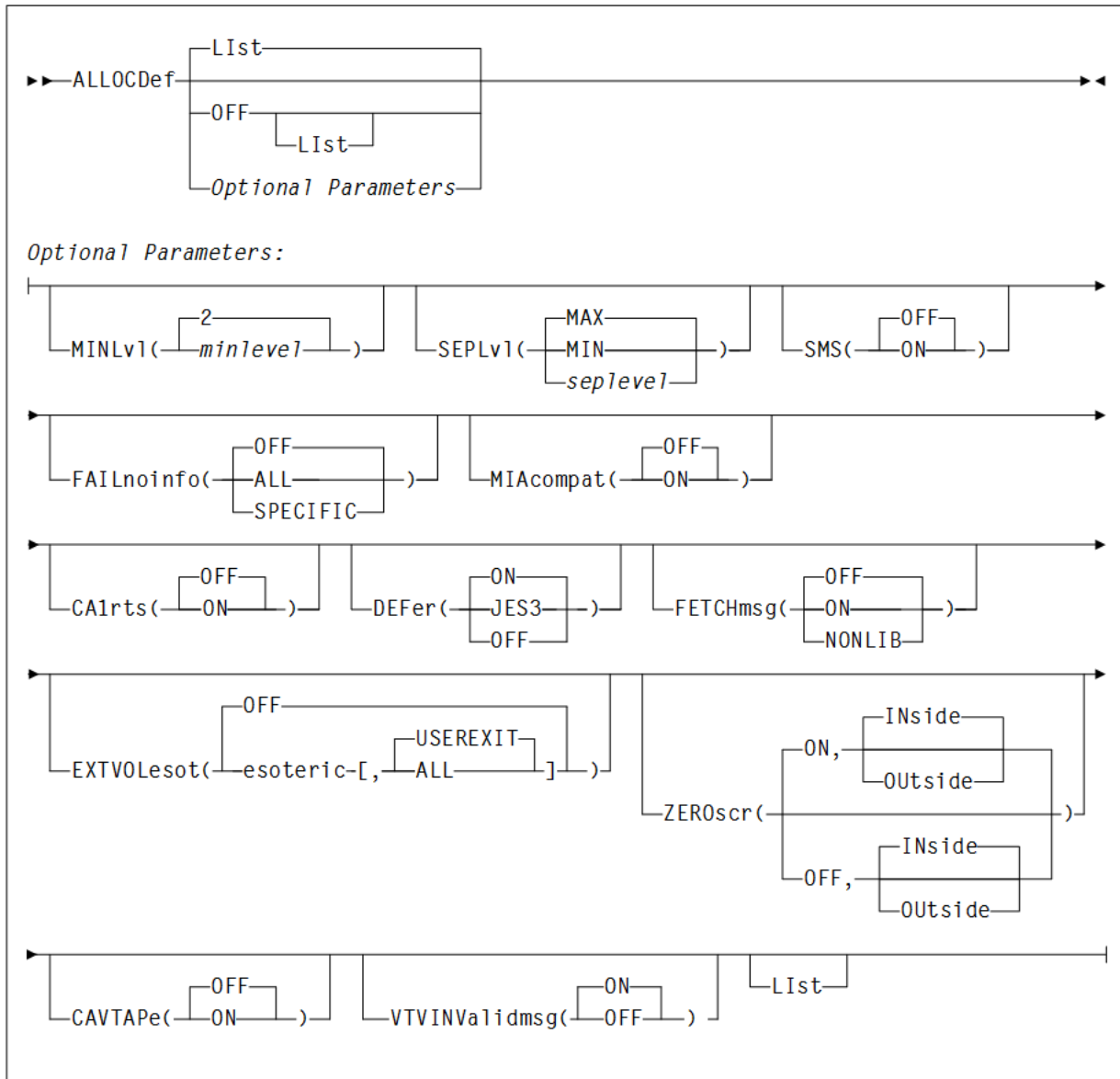
### Description

The SMC ALLOCDef command specifies default allocation settings for the SMC subsystem.

### Syntax

[Figure 2–1](#) shows ALLOCDef command syntax.

Figure 2–1 ALLOCDDef command syntax



## Parameters

As shown in Figure 2–1, the ALLOCDDef command includes the following parameters:

### LIst

optionally, lists current default allocation settings.

- LIst is the default when no parameters are specified on the ALLOCDDef command.
- LIst may be specified with other parameters. In this case, the LIst is generated after the other parameters are processed.

### OFF

optionally, resets all default allocation values to original SMC default settings. Specify LIst with this parameter to list these settings.



## Optional Parameters

### MINLvl(minlevel)

optionally, specifies the minimum level of drive exclusion. If a job is not allocatable at the minimum exclusion level, the SMC still excludes drives to the minimum level and allows the job to fail.

*minlevel* indicates the minimum allocation exclusion level. Valid values are 0-8. 2 is the default.

Specifying `MINLVL=0` indicates that the job will not be failed by the SMC even if the device and specific volume are incompatible. You can use `MINLVL=0` to force SMC to allow allocation to succeed when a “foreign” tape cartridge has the same volume serial number as an existing library or virtual volume.

Increasing *minlevel* has no effect unless message SMC0045 or SMC0091 has been issued for a specific job step.

Refer to Oracle’s ELS publication *Configuring and Managing SMC* for more information about SMC exclusion levels.

### SEPLvl

optionally, specifies the exclusion level at which affinity and GDG chains are separated.

Specify one of the following values:

#### **seplevel**

an exclusion level between the minimum and maximum levels. When sufficient drives exist, SMC attempts to separate chains at this level. This value cannot be less than the *minlevel* value.

#### **MAX**

Separate affinity and GDG chains for conflicting exclusion criteria whenever sufficient drives are available. This is the default.

#### **MIN**

Do not to separate affinity and GDG chains beyond the minimum level.

If HSC or MVS/CSC allocation previously specified `UNITAFF(NOSEP)` and `GDGALL(NOSEP)`, the default SMC exclusion tables can specify `SEPLvl=3` in order to preserve existing behavior for scratch affinity chains with different media types.

Refer to Oracle’s ELS publication *Configuring and Managing SMC* for more information about SMC exclusion levels.

### SMS

optionally, enables or disables the DFSMS interface. When the DFSMS interface is enabled, the SMC invokes the DFSMS ACS routines. The returned constructs are used to influence device allocation of data sets. See “[SMSDef](#)” on page 2-74 for information about using specific SMS features.

Specify one of the following values:

#### **OFF**

Disable the DFSMS interface. This is the default.

#### **ON**

Enable the DFSMS interface.

**FAILnoinfo**

optionally, specifies whether the SMC fails a job step during allocation when a communication failure prevents the retrieval of volume information from a TapePlex, or when no TapePlexes are available.

Specify one of the following values:

**OFF**

SMC does not fail the job step. This is the default.

**ALL**

SMC marks all devices ineligible and fails the job step.

**SPECIFIC**

If a communication failure occurs during the volume lookup process for a specific volume, or if no TapePlexes are available and the step contains allocations for specific volumes, the SMC marks all devices ineligible and fails the job step.

If the job step contains only scratch allocations, then the job step is allowed to proceed and allocation is based solely on SMC tape policy specifications.

**MICompat**

optionally, specifies whether the EDL is updated at SSI24 time for compatibility with Computer Associates Unicenter CA-MIA Tape Sharing for z/OS and OS/390 product. This parameter is not valid in JES3 environments with TAPE SETUP processing.

Specify one of the following values:

**OFF**

Do not update the EDL. This is the default.

**ON**

Update the EDL. Specify this value if you use Unicenter CA-MIA.

**CA1rts**

optionally, specifies whether the DEFER processing is performed at SSI24 time for compatibility with Computer Associates Real Time Stacking feature of its CA-1 tape management system. This parameter is not valid in JES3 environments with TAPE SETUP processing.

Specify one of the following values:

**OFF**

Do not update DEFER status. This is the default.

**ON**

Update DEFER status. Specify this value if you use the Real Time Stacking feature of CA-1.

**DEFer**

optionally, enables or disables deferred mount processing for library mounts. With deferred mounting enabled, a library resident volume is mounted when the data set is opened. If the data set is not opened, the cartridge is not mounted, freeing the robot to perform other work. If the data set is opened, however, the job waits until the cartridge is mounted.

Specify one of the following values:

**ON**

Enable deferred mount processing. This parameter overrides the user's JCL and defers all ACS mounts until the data set is opened. This is the default.

**JES3**

In a JES3 (with SETUP) environment, all mounts are JES3 deferred until a step begins execution.

**OFF**

Disable deferred mount processing and honor user JCL specifications.

---

---

**Note:** On the JES3 SETPARAM initialization statement, set the DEFERCT parameter to YES to ensure that jobs requiring deferred mounts are included in SDEPTH job counts. Refer to IBM JES3 publications for detailed information about SETPARAM.

---

---

**FETCHmsg**

optionally, for JES3 with SETUP environments that use volume fetch, specifies whether fetch messages are issued for tape volumes that are allocated to a library drive.

Specify one of the following values:

**OFF**

Fetch messages are not issued for volumes allocated to library drives. Fetch messages are still issued for volumes allocated to nonlibrary drives. This is the default.

**ON**

Fetch messages are issued for all volumes, regardless of whether they are allocated to library or nonlibrary drives.

**NONLIB**

Fetch messages are issued for nonlibrary volumes allocated to a library transport.

If the drive and volume both reside inside the library (for example, the volume can be automatically mounted) then the fetch message is suppressed. Queries to the library subsystem(s) are required to determine volume location, and may have a negative impact on performance.

**EXTVOLesot**

optionally, directs the SMC to use a specified esoteric to allocate a specific external volume (that is, a volume that is not in a TapePlex). When this esoteric is used depends upon the setting of the modifier value specified after the esoteric, as well as whether the specified esoteric is valid (intersects with the original esoteric specification).

Specify one of the following values:

**OFF**

Disable EXTVOLesot processing. If possible, select Nonlibrary drives when a specific external volume is allocated. This is the default.

**esoteric,USEREXIT**

Specifies an esoteric to be used to allocate a specific external volume when the "use specvol" (UX08) or "use library drives" (UX13) return code is specified. If the specific volume user exit returns the "use specvol" (UX08) or "use library drives" (UX13) return code for this DD, then any drives in the specified esoteric that

intersect with the original esoteric will be selected for allocation. This is the default if esoteric is specified.

**esoteric,ALL**

Whenever an external volume is allocated, select drives in the specified esoteric.

**ZEROscr**

optionally, specifies the exclusion action when there are no scratch volumes in any TapePlex (ON or OFF), or in one or more ACSs within a TapePlex in a multiple ACS environment (INside or OUTside).

Specify one of the following values:

**ON**

This is the default. ON indicates one of the following:

- If scratch subpools are not being used, and one or more ACSs contain zero scratch volumes, then drives in those ACSs are excluded from the list of eligible devices.
- If a scratch subpool is being requested (either through TAPEREQ or User Exit 02/04), and one or more ACSs contain zero scratch volumes in the requested subpool, then drives in those ACSs are excluded from the list of eligible devices.
- In JES3 with SETUP environments only, drives in the ACS with the largest number of available scratch volumes remain eligible. All other drives are excluded from selection.

INside specifies that all nonlibrary drives are excluded when there are no scratch volumes in any ACS, and there are library drives eligible for allocation. This is the default.

OUTside specifies that all library drives are excluded when there are no scratch volumes in any ACS, and there are nonlibrary drives eligible for allocation.

**OFF**

All drives in all ACSs are to remain eligible for selection.

INside specifies that all nonlibrary drives are excluded when there are no scratch volumes in any ACS, and there are library drives eligible for allocation. This is the default.

OUTside specifies that all library drives are excluded when there are no scratch volumes in any ACS, and there are nonlibrary drives eligible for allocation.

**CAVTAPe**

optionally, specifies whether scratch user exit 02 or specific user exit 08 are called when an SMC POLICY object is found that applies to the allocation event.

Specify one of the following values:

**OFF**

If an SMC POLICY object applies to an allocation event, then user exit 02 and user exit 08 are not called, even if active. The SMC POLICY object supplies all of the SMC tape policy in such instances. This is the default.

**ON**

If an SMC POLICY object applies to an allocation event, then the scratch user exit 02 or specific user exit 08 are called and any non-conflicting user exit policy is applied to the same allocation event.

**Note:**

This setting is recommended only for those customers with CA-Vtape installed who are using a default SMC POLICY object but where CA-Vtape supplied user exits are required.

Only non-conflicting policies from the user exit are applied. Thus, if the SMC POLICY specifies an ESOTERIC, or a TAPEPLEX name, and user exit 02 or user exit 08 also specifies an ESOTERIC or TAPEPLEX name, the SMC POLICY specifications will apply. Refer to Oracle's ELS publication *Configuring and Managing SMC* for more information about SMC interaction with CA-Vtape.

**VTVINValidmsg**

optionally, specifies whether the SMC will fail a job step during allocation under the following conditions:

- Specific volume request.
- Volume is unknown to HSC or VTCS.
- Volser does not match any VOLATTR or VOLPARM statements.
- The incoming eligible device list contains only virtual devices.

Specify one of the following values:

**ON**

SMC does not fail the job step. This is the default.

**OFF**

SMC marks all devices ineligible and fails the job step.

## ALLOcJob

**Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required, or may be input to the SMCUSIM utility

## Description

The SMC ALLOcJob command enables you to override SMC default allocation parameters by job name, step name, and PROC step.

---

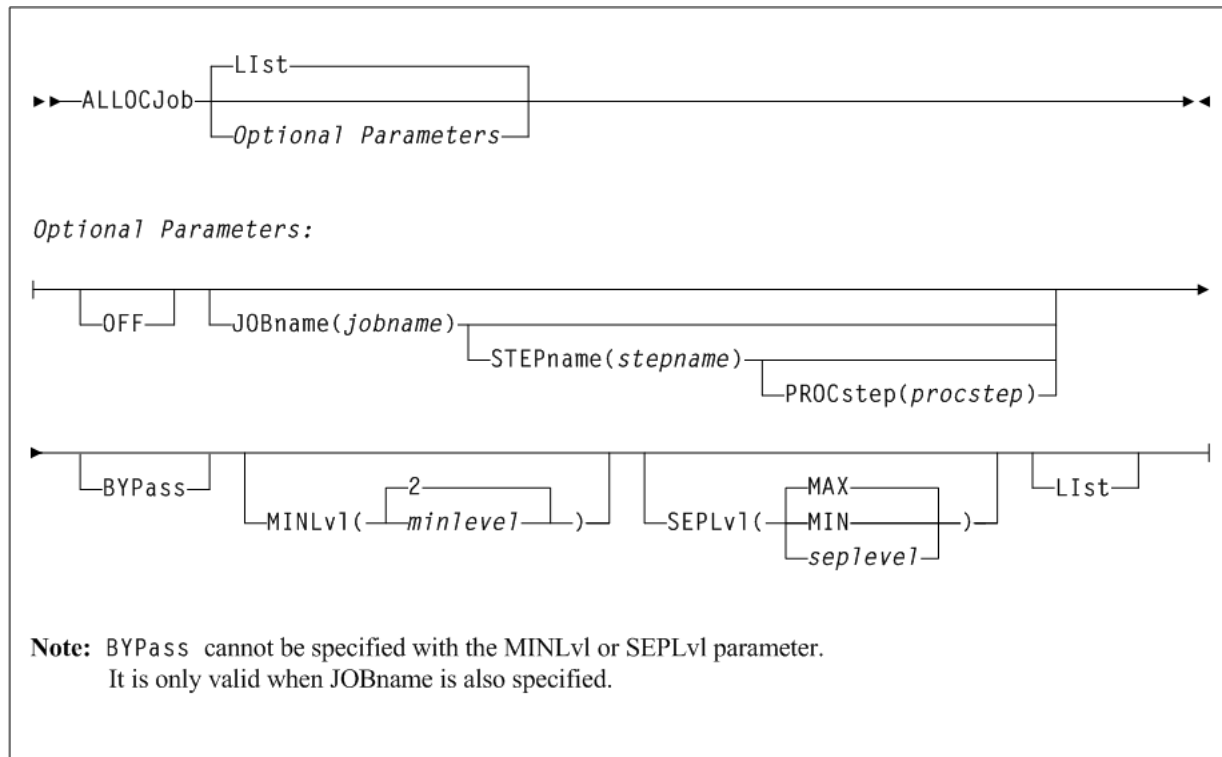
**Note:** The SMS parameter is no longer supported for this command. If necessary, code your SMS ACS routines to select jobnames.

---

## Syntax

Figure 2–2 shows ALLOcJob command syntax.

Figure 2-2 ALLOcJob command syntax



## Parameters

As shown in [Figure 2-2](#), the ALLOcJob command includes the following parameters:

### List

optionally, lists current default allocation settings and override settings (in the order they are processed) by job name, step name, and PROC step.

- List is the default when no parameters are specified on the ALLOcJob command.
- List may be specified with other parameters. In this case, the List is generated after the other parameters are processed.

### Optional Parameters

#### OFF

optionally, removes all job name, step name, and PROC step overrides for allocation.

- If no other parameters are specified, allocation overrides are removed for all jobs.
- If only JOBname is specified, allocation overrides are removed for ALL ALLOcJob entries for that job name.
- If JOBname, STEPname and PROCstep are specified, allocation overrides are removed only for the specified entry.

Global allocation settings are not affected by this parameter.

#### JOBname(*jobname*)

optionally, specifies a job name. This parameter is required unless List or OFF is specified.

*jobname* indicates the job name. The value entered must be one to eight characters in length. An asterisk (\*) can be used as a wildcard when included as the last character. ALLOcJob is set for all jobs whose job names match the characters preceding the asterisk.

#### **STEPname(stepname)**

optionally, specifies a step name. This parameter is only valid when JOBname is also specified. It is not valid in JES3 environments with TAPE SETUP processing.

*stepname* indicates the step name. The value entered must be one to eight characters in length. An asterisk (\*) can be used as a wildcard when included as the last character.

#### **PROCstep(procstep)**

optionally, specifies a PROC step. This parameter is only valid when JOBname and STEPname are also specified. It is not valid in JES3 environments with TAPE SETUP processing.

*procstep* indicates the PROC step. The value entered must be one to eight characters in length. An asterisk (\*) can be used as a wildcard when included as the last character.

#### **BYPass**

optionally, specifies that SMC allocation influencing is not performed for the indicated job (*jobname*, *stepname*, or *procstep*).

This parameter cannot be specified with the MINLVL, SEPLVL, or SMS parameter. It is only valid when JOBname is also specified.

---

**Note:** It is recommended that you use MINLV=0 instead of BYPASS, except when no mounts will be performed (that is, IEFBR14 jobs).

---

#### **MINLvl(minlevel)**

optionally, specifies the desired minimum level of drive exclusion. If a job is not allocatable at the minimum exclusion level, the SMC still excludes drives to the minimum level and allows the job to fail.

This parameter cannot be specified with the BYPass parameter. It is only valid when JOBname is also specified.

*minlevel* indicates the desired minimum allocation exclusion level. Valid values are 0-8. 0 indicates that the SMC will not exclude any drives, even if all drives would be excluded during level 1 exclusion processing. The default is the current value of the ALLOCDEF MINLVL setting. Refer to Oracle's ELS publication *Configuring and Managing SMC* for more information about SMC exclusion levels.

MINLVL=0 indicates that the job will not be failed by the SMC even if the device and specific volume are incompatible. MINLVL=0 may be used to force SMC to allow allocation to succeed when a "foreign" tape cartridge has the same volume serial number as an existing library or virtual volume.

Increasing *minlevel* has no effect unless message SMC0045 or SMC0091 has been issued for a specific job step.

#### **SEPLvl**

optionally, specifies the exclusion level at which affinity and GDG chains are separated. This parameter cannot be specified with the BYPass parameter. It is only valid when JOBname is also specified.

Specify one of the following values:

**seplevel**

an exclusion level between the minimum and maximum levels. When sufficient drives exist, the SMC attempts to separate chains at this level. Refer to Oracle’s ELS publication *Configuring and Managing SMC* for more information about SMC exclusion levels. The *seplevel* value cannot be less than the *minlevel* value.

**MIN**

Do not separate Affinity and GDG chains beyond the minimum level.

**MAX**

Separate Affinity and GDG chains for conflicting exclusion criteria whenever sufficient drives are available. This is the default.

## CMDDef

**Interfaces:**

- Console, utility, or SMCPARMS data set
- UII Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required

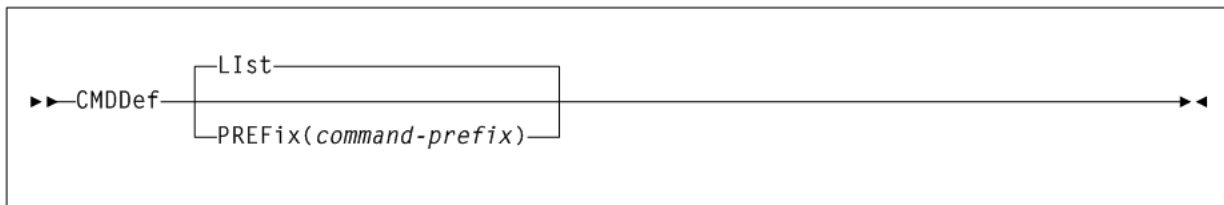
## Description

The SMC `CMDDef` command specifies an SMC command prefix.

## Syntax

Figure 2–3 shows `CMDDef` command syntax.

**Figure 2–3** *CMDDef* command syntax



## Parameters

As shown in Figure 2–3, the `CMDDef` command includes the following parameters:

**List**

optionally, lists the current command prefix.

This is the default if no parameters are specified. It cannot be specified with any other parameter.

**PREFIX(command-prefix)**

optionally, specifies a command prefix for the SMC subsystem. This parameter may only be specified in the `SMCPARMS` data set.

*command-prefix* indicates the command prefix. The value entered must be one to eight characters in length and meet the following requirements:



- Valid characters include A-Z 0-9 @ ¢ \$ # , . / ' ( ) < > \* & + - = | ! ; : " % \_ ?
- The command prefix cannot include a command string, a command abbreviation, or any string that invokes a command.
- The command prefix cannot include a string that is a subset or superset of an existing prefix beginning with the same character.

---

---

**Note:** The MVS command, `DISPLAY OPDATA`, displays all active command prefixes and their corresponding subsystem name.

---

---

## COLlector

### Interfaces:

- Console, SMCCMDS data set, or SMCPARMS data set
- UI Support: No

### Subsystem Requirements:

Active SMC required

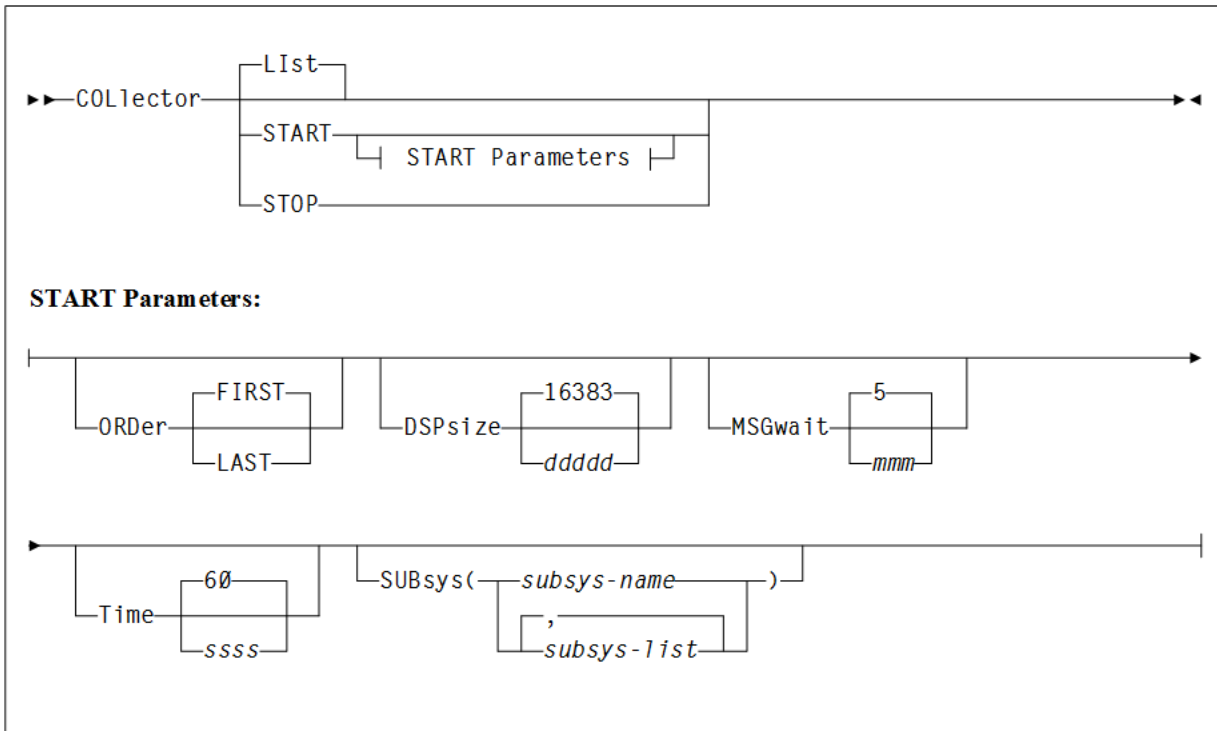
## Description

The SMC COLlector command is used to start or stop the statistical gathering function. This function allows the Virtual Storage Manger GUI to retrieve z/OS SMF records in real time.

## Syntax

Figure 2–4 shows COLlector command syntax.

Figure 2-4 COLlector command syntax



## Parameters

As shown in Figure 2-4, the COLlector command includes the following parameters:

### LIST

optionally, displays status information for the SMC collector.

This is the default if no parameters are specified. It cannot be specified with any other parameter.

### START

optionally, starts the SMC collector.

### STOP

optionally, stops the SMC collector.

### Optional START Parameters

#### ORDER

optionally, specifies the order of the SMCXU83 IEFU83/IEFU84 exit. The parameter is used to order the SMCXU83 exit when other exits are present. The default is FIRST.

#### DSPsize

optionally, specifies the size (number of 4K blocks) of the collector data space used to accumulate SMF records. The data space is created on the first COLlector START. DSPsize can not be changed after the first start. The default is a data space of 64M.

dddd indicates the number of 4K blocks.

**DISPMsg**

optionally, specifies whether SMC0274 and SMC0275 messages are displayed. Message SMC0274 messages indicates that the SMC collector data space is full.

Specify YES or NO. NO is the default.

**MSGwait**

optionally, specifies the time between SMC0274 messages. This message indicates that the SMC collector data space is full.

*mmm* indicates the number of minutes to wait.

**Time**

optionally, specifies the time to pause the data transmission to the requestor due to no data to send.

*ssss* indicates the number of seconds.

**SUBsys**

optionally, specifies one or more subsystem names if they are defined in the SMFPRMxx member. Up to 10 subsystem names can be entered.

*subsys-list* is a list of subsystem names.

## COMMtest

**Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required, or may be input to the SMCUSIM utility

## Description

The SMC `COMMtest` command performs 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, 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.

---

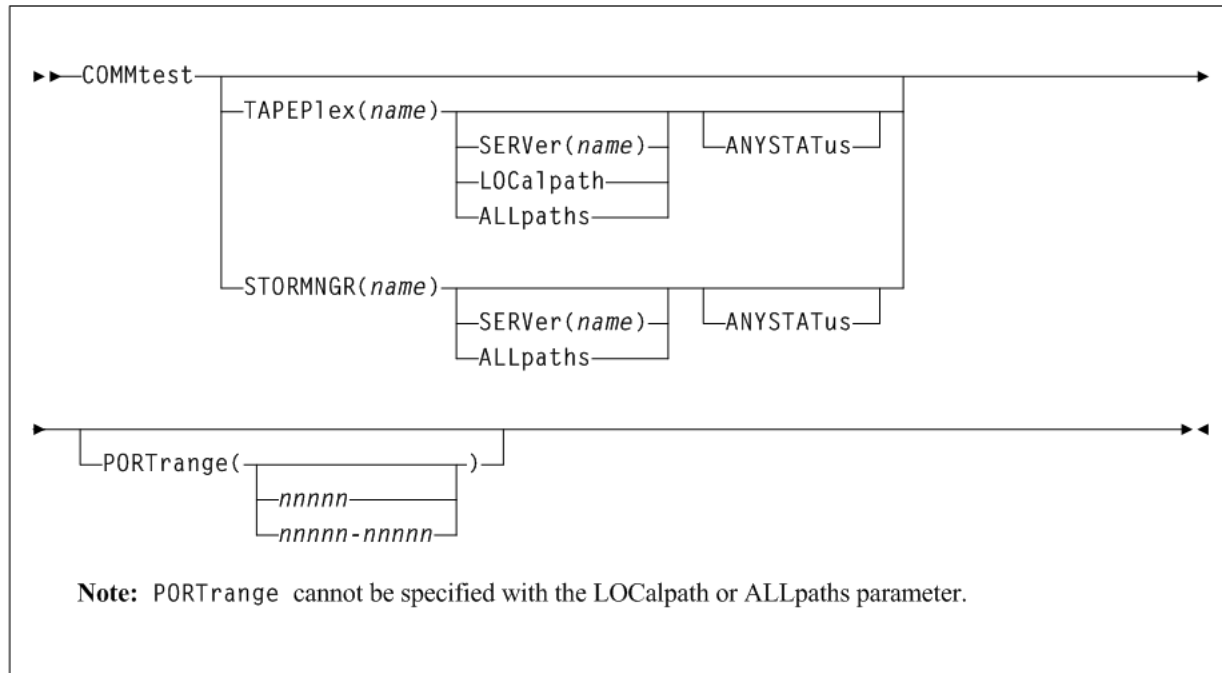


---

## Syntax

Figure 2–5 shows `COMMtest` command syntax.

Figure 2-5 COMMtest command syntax



## Parameters

As shown in [Figure 2-5](#), the COMMtest command includes the following parameters:

### TAPEPlex(tapeplex-name)

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

*tapeplex-name* indicates the TapePlex name as defined by the SMC TAPEPlex command. 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.

TAPEPlex and STORMNGR are mutually exclusive.

### STORMNGR(stormngr-name)

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

*stormngr-name* indicates a VLE name as defined by the SMC STORMNGR command.

TAPEPlex and STORMNGR are mutually exclusive.

### PORTrange(nnnnn or nnnnn-nnnnn)

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.

*nnnnn* or *nnnnn-nnnnn* indicates 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.

`PORTrange` is mutually exclusive with `LOCALpath` and `ALLpaths`.

#### **SERVER(server-name)**

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

*server-name* indicates the server path name as defined by the `SMC SERVER` command. 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.

#### **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.

## DISMount

#### **Interfaces:**

- Console or utility
- UUI Support: Yes (No XML/CSV output)

#### **Subsystem Requirements:**

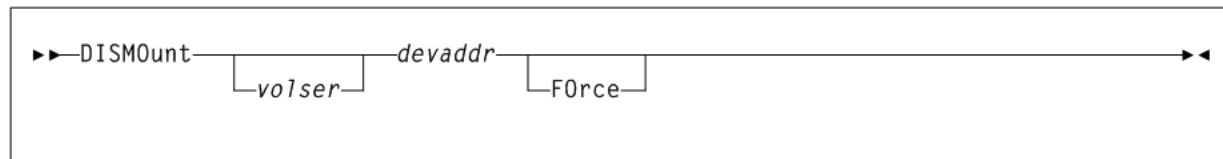
Active `SMC` required, or may be input to the `SMCUSIM` utility

## Description

The `SMC DISMount` command requests the dismount of a volume from a drive.

## Syntax

Figure 2–6 shows `DISMount` command syntax.

**Figure 2–6** DISMOunt command syntax

## Parameters

As shown in [Figure 2–6](#), the DISMOunt command includes the following parameters:

### **volser**

optionally, the volume to be dismounted. The volume is dismounted only if the volume on the drive matches the specified *volser*. If *volser* is omitted, any volume that is on the drive is dismounted.

### **devaddr**

The MVS device address for which the volume is to be dismounted. The device must be defined by a TapePlex known to SMC.

This is a required value.

---



---

**Note:** If *volser* is omitted, a comma must precede the device address.

---



---

### **FOrce**

optionally, indicates that if a loaded volume is found on the drive, it should be rewound and unloaded prior to attempting the requested dismount. This parameter is not supported for virtual drives.

## Display DDrive

### **Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (No XML/CSV output)

### **Subsystem Requirements:**

Active SMC required, or may be input to the SMCUSIM utility

## Description

The SMC Display DDrive command displays SMC drive attribute and TapePlex ownership information.

---



---

**Note:** TapePlex drive information can also be obtained by issuing the HSC Display DDrives command via the SMC Route command.

---

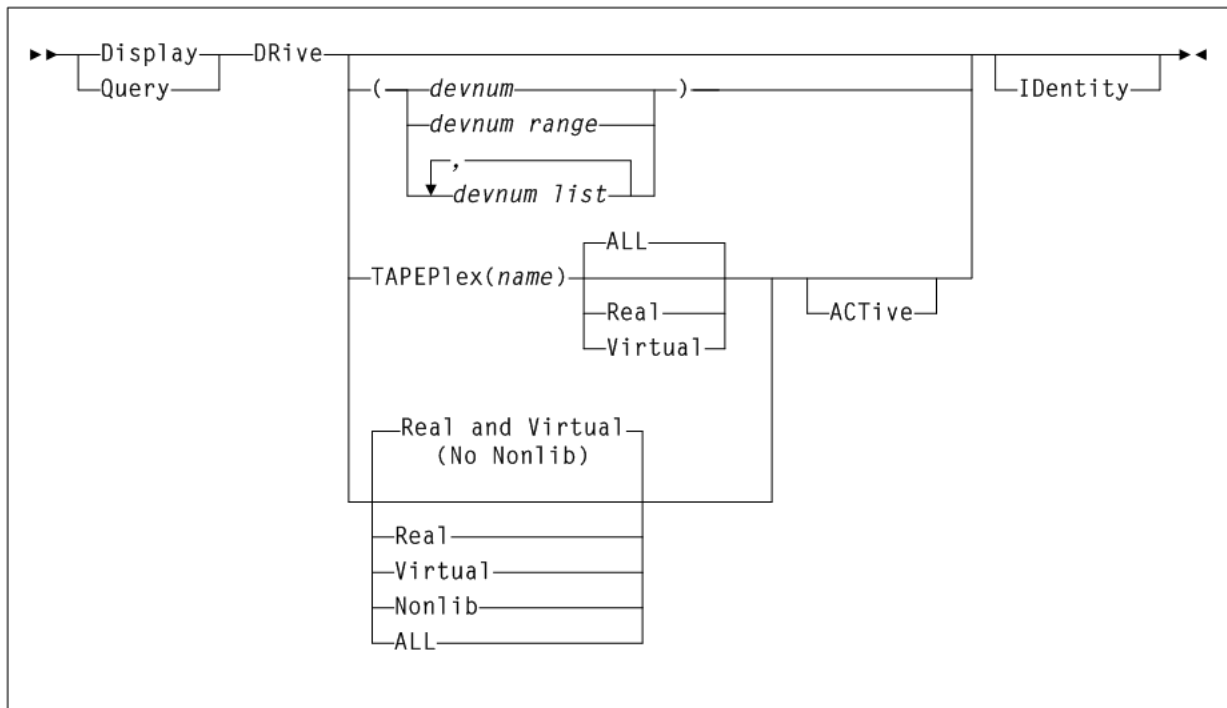


---

## Syntax

[Figure 2–7](#) shows Display DDrive command syntax.

Figure 2-7 Display DRive command syntax



## Parameters

As shown in [Figure 2-7](#), the `Display DRive` command includes the following parameters:

### **devnum, devnum-range, or devnum-list**

a device number, range of device numbers, or list of device numbers to be processed. If specified, all device addresses specified are displayed. The *devnum* form of the `Display DRive` command is the only way that defined devices that are not TapePlex owned, or not defined by an `SMC UNITAttr` command, can be displayed.

*devnum*, *devnum-range*, and *devnum-list* are mutually exclusive with all other `Display DRive` parameters.

### **TAPEPlex(name)**

optionally, lists only devices owned by the specified TapePlex.

*name* indicates the TapePlex name. 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.

### **Real**

optionally, lists only “real” (that is, nonvirtual) devices owned by the specified TapePlex.

### **Virtual**

optionally, lists only virtual devices owned by the specified TapePlex.

**ALL**

optionally, lists all devices owned by the specified TapePlex. This is the default if no parameters are specified.

**Real**

optionally, lists only “real” (that is, nonvirtual) devices owned by SMC-defined TapePlexes.

**Virtual**

optionally, lists only virtual devices owned by SMC-defined TapePlexes.

**Nonlib**

optionally, lists only “real” devices that are not owned by any SMC-defined TapePlex, but have UNITATTR statements defined.

**ALL**

optionally, lists all devices owned by SMC-defined TapePlexes.

**ACTive**

optionally, lists only drives currently mounted or awaiting mounts according to SMC internal data.

---

---

**Note:** If a `Display DRive` command is issued without any parameters, by default, the SMC lists all real and virtual devices owned by SMC-defined TapePlexes.

---

---

**IDentity**

optionally, displays informational message SMC0178, identifying the drive serial number.

## Display POLicy

**Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (supports XML and CSV)

**Subsystem Requirements:**

Active SMC required, or may be input to the SMCUSIM utility

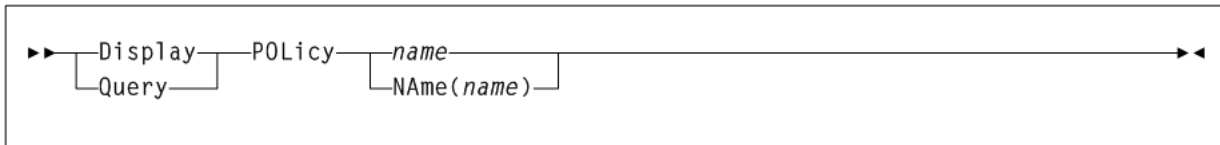
## Description

The SMC `Display POLicy` command displays information for a named policy. This command provides the same output as the SMC `POLicy` command with the `LIST` keyword.

## Syntax

Figure 2–8 shows `Display POLicy` command syntax.



**Figure 2–8 Display POLIcy command syntax**

## Parameters

As shown in [Figure 2–8](#), the `Display POLIcy` command includes the following parameters:

### **name or NAme(name)**

optionally, specifies the name of a policy that matches the name specified in a `TAPEREQ` or `DFSMS ACS` routine management class.

*name* may be specified with or without the keyword `NAme`. If the keyword `NAme` is not specified, the *name*, if specified, must be the first parameter.

This is a required value.

## Display RC

### Interfaces:

- Console, utility, `SMCCMDS` data set, or `SMCPARMS` data set
- UII Support: Yes (supports XML and CSV)

### Subsystem Requirements:

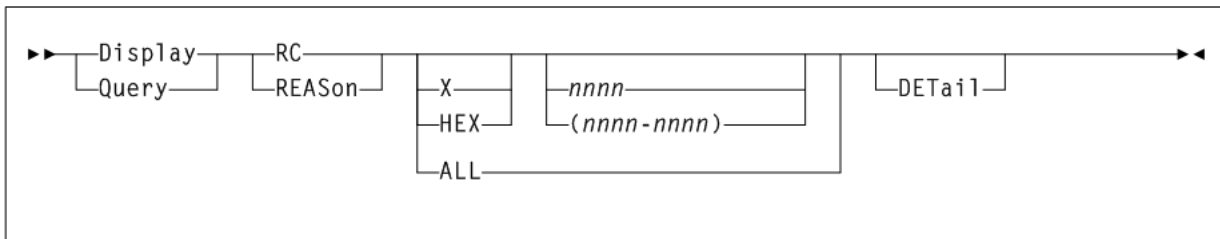
Active SMC required, or may be input to the `SMCUSIM` utility

## Description

The `SMC Display RC` command displays information about the meaning of an SMC return or reason code, or an HSC/VTCS UII reason code.

## Syntax

[Figure 2–9](#) shows `Display RC` command syntax.

**Figure 2–9 Display RC command syntax**

## Parameters

As shown in [Figure 2–9](#), the `Display RC` command includes the following parameters:

**X or HEX**

optionally, specifies that the reason or return code value or range is specified as a hexadecimal number.

**nnnn or nnnn-nnnn**

optionally, lists the return codes for which the explanation is to be displayed.

- If X or HEX is specified, the value may contain hexadecimal characters 0-9 and A-F.
- If X or HEX is not specified, the value may contain only numeric characters.

A range specified from the console may contain a maximum of 16 values.

**ALL**

optionally, indicates that all defined return or reason codes are to be listed. ALL is permitted only from a utility. ALL and H or HEX are mutually exclusive.

**DETail**

optionally, indicates that detailed information about the requested codes is to be listed.

## Display SERVER

**Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (supports XML and CSV)

**Subsystem Requirements:**

Active SMC required, or may be input to the SMCUSIM utility

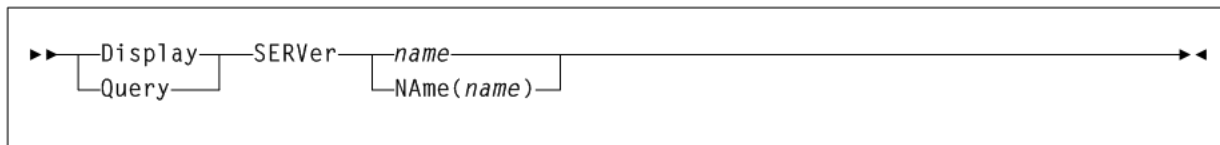
## Description

The SMC Display SERVER command displays information for a named server. This command provides the same output as the SMC SERVER command with the LIST keyword.

## Syntax

Figure 2–10 shows Display SERVER command syntax.

**Figure 2–10 Display SERVER command syntax**



## Parameters

As shown in Figure 2–10, the Display SERVER command includes the following parameters:

**name or NAME(name)**

indicates the server name to be defined, modified, or listed. This is a required value.

*name* may be specified with or without the keyword NAME. If the keyword NAME is not specified, the *name*, if specified, must be the first parameter.

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.

## Display SMC

### Interfaces:

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (supports XML and CSV)

### Subsystem Requirements:

Active SMC required, or may be input to the SMCUSIM utility

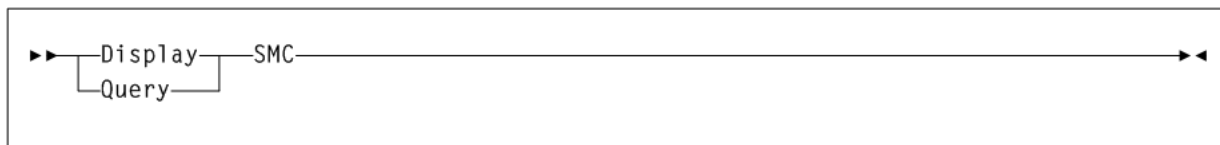
## Description

The SMC `Display SMC` command displays SMC status information, including start time, release, and host information.

## Syntax

Figure 2–11 shows `Display SMC` command syntax.

**Figure 2–11** *Display SMC command syntax*



## Parameters

None.

## Display STORMNGR

### Interfaces:

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (supports XML and CSV)

### Subsystem Requirements:

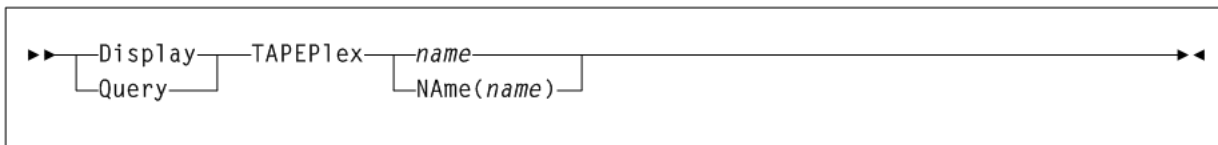
Active SMC required, or may be input to the SMCUSIM utility

## Description

The SMC `Display STORMNGR` command displays information for a named VLE. This command provides the same output as the SMC `STORMNGR` command with the `List` keyword.



Figure 2–13 Display TAPEPlex command syntax



## Parameters

As shown in Figure 2–13, the `Display TAPEPlex` command includes the following parameters:

### **name or NAme(name)**

indicates the TapePlex name to be added, modified, or listed. This is a required value.

*name* may be specified with or without the keyword `NAme`. If the keyword `NAme` is not specified, the name, if specified, must be the first parameter.

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.

## Display Volume

### **Interfaces:**

- Console, utility, `SMCCMDS` data set, or `SMCPARMS` data set
- UUI Support: Yes (No XML/CSV output)

### **Subsystem Requirements:**

Active SMC required, or may be input to the `SMCUSIM` utility

## Description

The `SMC Display Volume` command displays volume attribute and TapePlex ownership information.

---

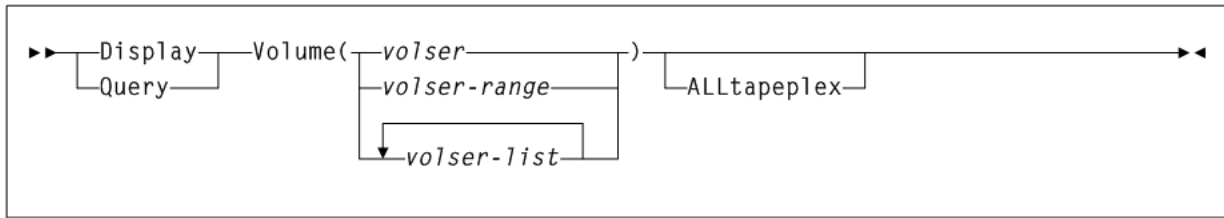
---

### **Note:**

- TapePlex volume information is also available by issuing the `HSC Display Volume` command via the `SMC Route` command.
  - When the server is `ACSLs`, this command displays `NONE` for `MEDIA` and `RECTECH`.
- 
- 

## Syntax

Figure 2–14 shows `Display Volume` command syntax.

**Figure 2–14 Display Volume command syntax**

## Parameters

As shown in [Figure 2–14](#), the `Display Volume` command includes the following parameters:

### **volser, volser-range, or volser-list**

indicates the *volser*, *volser range*, or *volser list* to be processed. If multiple volumes are specified, only the first 100 are queried. This is a required value.

### **ALLtapeplex**

optionally, specifies that all active TapePlexes are queried for the specified *volser*, *volser-range*, or *volser-list*. If specified, multiple display lines may be listed for the same *volser* if it is defined in multiple TapePlexes.

If this parameter is not specified, the `Display Volume` command queries TapePlexes in the order they are defined and lists only the first occurrence of the volume.

## DRIVemap

### **Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UI Support: Yes (No XML/CSV output)

### **Subsystem Requirements:**

Active SMC required, or may be input to the `SMCUSIM` utility

## Description

The SMC `DRIVemap` command maps client drive addresses to server drive addresses. This command enables users to specify different MVS device addresses on different hosts for the same TapePlex or virtual drives in client-server mode.

---



---

**Note:** Refer to Oracle's ELS publication *Configuring and Managing SMC* for more information about drive mapping and for information about using `DRIVemap` commands and the HSC `DRVHOST` parameter when client and server addresses differ.

---

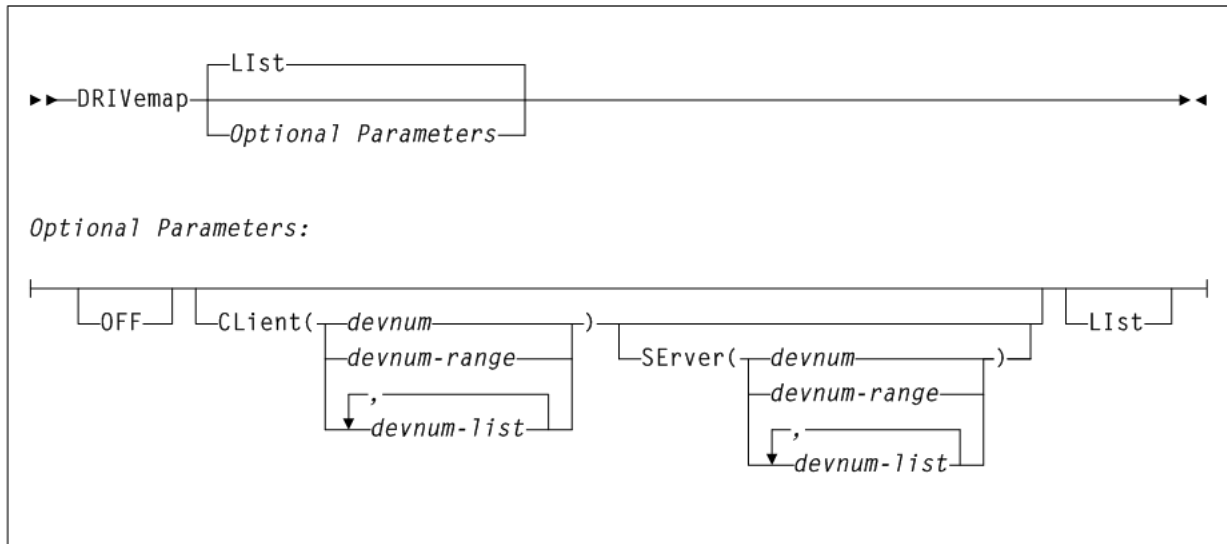


---

## Syntax

[Figure 2–15](#) shows `DRIVemap` command syntax.

Figure 2–15 DRIVemap command syntax



## Parameters

As shown in [Figure 2–15](#), the `MNTD` command includes the following parameters:

### List

optionally, lists all current DRIVemap mappings.

- `List` is the default when no parameters are specified on the DRIVemap command. In this case, all DRIVemap entries are listed.
- `List` may be specified with other parameters. In this case, the `List` is applied after the other parameters are processed.

### Optional Parameters

#### OFF

optionally, removes all current DRIVemap mappings. When specified with `CLient`, this parameter removes DRIVemap ranges specified in the `CLient` parameter.

#### CLient(devnum, devnum-range, or devnum-list)

optionally, specifies the device numbers mapped by the DRIVemap command.

*devnum*, *devnum-range*, or *devnum-list* indicates the device number, device number range, or device number list.

#### SErver(devnum, devnum-range, or devnum-list)

optionally, specifies the device numbers that are defined on the HSC server.

*devnum*, *devnum-range*, or *devnum-list* indicates the device number, device number range, or device number list.

The ranges and lists must match those in the `CLient` parameter. For example:

- `CL(AA0-AA2) SE(BA0-BA2)` is valid.
- `CL(AA0-AA2) SE(BA0, BA1, BA2)` is invalid.

## Help

### Interfaces:

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (No XML/CSV output)

### Subsystem Requirements:

Active SMC required, or may be input to the SMCUSIM utility

## Description

The SMC `Help` command displays SMC command and message information.

## Syntax

Figure 2–16 shows `Help` command syntax.

**Figure 2–16 Help command syntax**



## Parameters

As shown in Figure 2–16, the `Help` command includes the following parameters:

---



---

**Note:** If you enter the `Help` command without any parameters, information is displayed for all available SMC commands.

---



---

### **command-name**

optionally, indicates an SMC command name.

### **nnnn**

optionally, indicates the four-digit numerical portion of an SMC message identifier. Leading zeros are not required.

### **nnnn-nnnn**

optionally, indicates a range of SMC messages specified using the four-digit numerical portion of the message identifier.

### **SMCnnnn**

optionally, indicates a full SMC message identifier.

### **SMCnnnn-SMCnnnn**

optionally, indicates a range of SMC messages specified using full message identifiers.



## HTTP

**Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: All (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required

### Description

The SMC HTTP command starts an SMC socket listener component that supports a limited subset of HTTP capabilities, and enables the local SMC subsystem to act as an HTTP server for SMC client subsystems on other hosts. Requests that are sent from other SMC client subsystems to this SMC with an active HTTP Server component are processed by the local HSC.

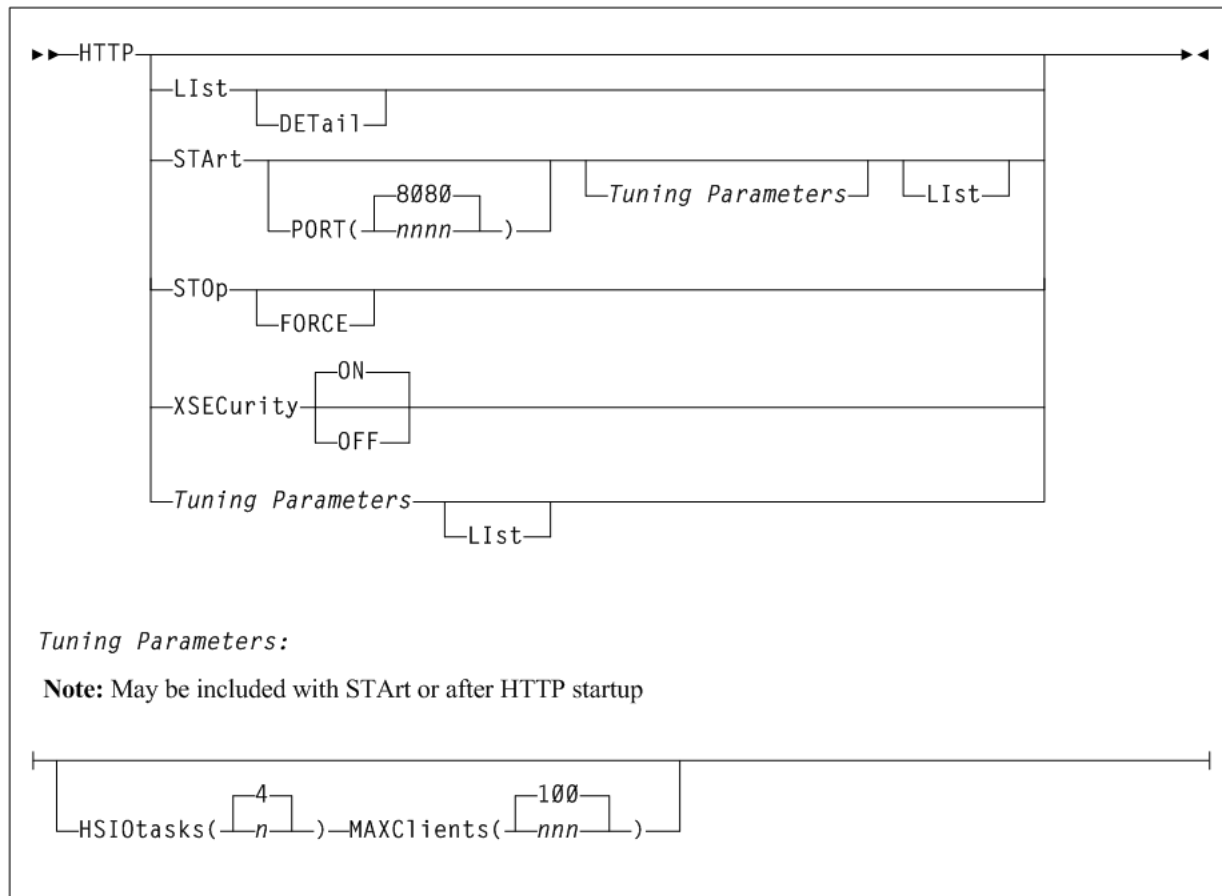
The SMC HTTP command should only be executed on hosts where an HSC subsystem is executing; otherwise the HTTP command provides no operational benefit.

SMC subsystems on other hosts that will be clients to this local SMC with active HTTP Server component must use the SMC SERVER command to define the communications path to this host.

### Syntax

Figure 2–17 shows HTTP command syntax.

Figure 2-17 HTTP command syntax



## Parameters

As shown in Figure 2-17, the HTTP command includes the following parameters:

### LIst

optionally, displays SMC HTTP server status information and interval statistics.

### DETAil

optionally, lists I/O, error, accept and reject counts, and CGI module use count information in addition to information displayed in the LIst command.

### STArt

optionally, starts the SMC HTTP server.

### PORT(nnnn)

optionally, specifies the port number to be listened to for incoming requests.

*nnnn* indicates the TCP/IP port number. It is recommended that you do not specify well known TCP/IP port numbers. If not specified, The default is 8080.

### STOp

optionally, stops the SMC HTTP server.

### FORCE

optionally, directs SMC to process the HTTP STOP command even if the status of the SMC HTTP server is not active. Use this parameter only if there has been an

unrecoverable HTTP service task termination and the normal SMC HTTP STOP command has been rejected.

### **XSECurity**

optionally, specifies whether the XAPI user password security protocol is globally enforced for requests processed by this HTTP server.

#### **ON**

specifies that the XAPI security protocol is enforced for this server. This is the default setting.

#### **OFF**

specifies that the XAPI security protocol is not enforced for this server. If specified then the server will accept all requests and operate like a pre-ELS 7.3 server.

### **Tuning Parameters**

The following parameters can be entered with START or after HTTP startup. It is strongly recommended that you do not change the default values for these parameters unless directed by Oracle StorageTek Software Support.

#### **HSIOtasks(n)**

optionally, specifies the maximum number of SMC HTTP socket I/O tasks to start.

*n* indicates the number of I/O tasks.

---



---

**Note:** The HSIOtasks parameter cannot be specified after HTTP server startup, it must be specified as part of the HTTP START command. However the remaining tuning parameters, MAXclients, SQUELEN, and WQUELEN may be changed after SMC HTTP server startup has completed.

---



---

#### **MAXclients(nnn)**

optionally, specifies the maximum number of connected requests that will be allowed at one time.

*nnn* indicates the number of requests. This value must be between 1 and 300. The default value is 100.

## **IDAX**

### **Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (No XML/CSV output)

### **Subsystem Requirements:**

Active SMC required

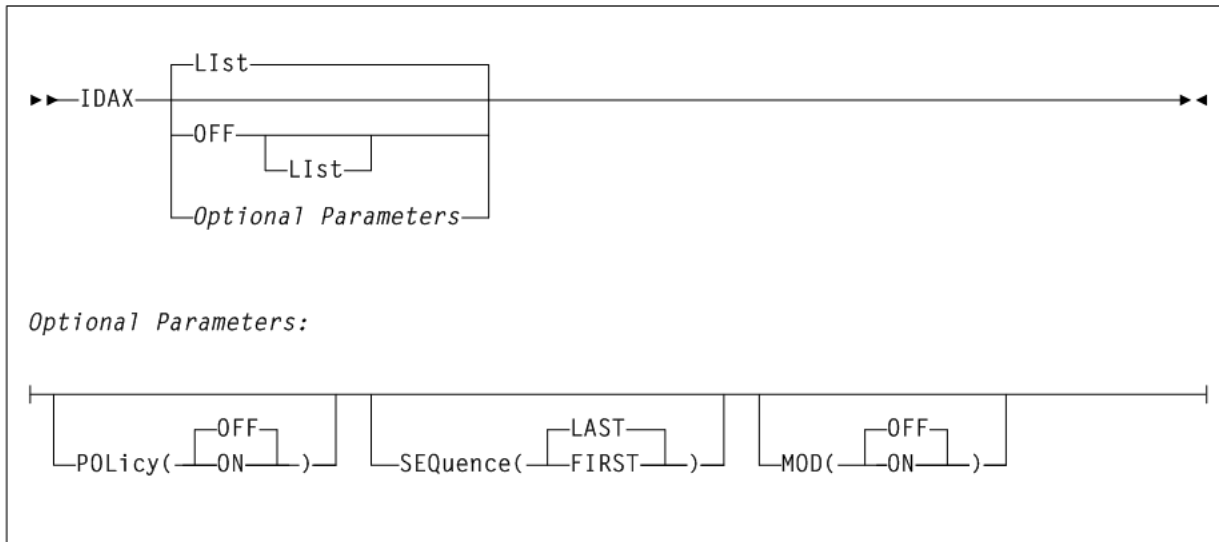
## **Description**

The SMC IDAX command controls SMC behavior during IDAX (MVS Interpreter/Dynamic Allocation Exit) processing.

## Syntax

Figure 2–16 shows IDAX command syntax.

**Figure 2–18 IDAX command syntax**



## Parameters

As shown in Figure 2–16, the IDAX command includes the following parameters:

---

**Note:** The `VTVMAXvol` parameter is no longer supported. Instead, use the `POLICY` parameter `IDAXVOLCNT`. Refer to *Configuring and Managing SMC* for more information about selecting IDAX policies.

---

### List

optionally, lists current SMC IDAX settings.

- `List` is the default when no parameters are specified on the IDAX command.
- `List` may be specified with other parameters. In this case, the `List` is applied after the other parameters are processed.

### OFF

optionally, resets all IDAX settings to original SMC default settings. Specify `List` with this parameter to list these settings. `OFF` cannot be specified with any other parameter.

### Optional Parameters

#### POLICY

optionally, specifies whether SMC `TAPEREQ` or `POLICY` command processing is performed during IDAX.

#### OFF

`TAPEREQ` or `POLICY` command processing is not performed during IDAX. This is the default.

#### ON

`TAPEREQ` or `POLICY` command processing is performed during IDAX.

---

---

**Note:**

- `IDAX POLICY(ON)` only applies to policy parameters that begin with the letters `IDAX`. If none of the policies contain `IDAX` policy parameters then setting `IDAX POLICY(ON)` adds unnecessary overhead.
  - SMC policy processing at `IDAX` using a DFSMS ACS routine `MGMTCLAS` as a policy name is controlled by the `MGMTPol` parameter of the `SMSDef` command.
- 
- 

**SEquence**

optionally, specifies the relative sequence of SMC and IBM DFSMS ACS routine invocation during `IDAX` (Interpreter/Dynamic Allocation Exit) processing.

**LAST**

SMC invokes the DFSMS ACS routines after IBM. This is the default.

**FIRST**

SMC invokes the DFSMS routines before IBM. This parameter allows your SMC DFSMS routines to assign tape esoterics to data sets with no specified esoteric, or to data sets that would otherwise be SMS-managed if IBM invoked the DFSMS routines before the SMC.

**MOD**

optionally, specifies whether the `POLICY IDAXESOT` parameters or SMC DFSMS esoteric substitution applies to `DISP=MOD` data sets as well as `DISP=NEW` data sets.

**OFF**

SMC `POLICY IDAXESOT` parameters or SMC DFSMS esoteric substitution do not apply to `DISP=MOD` data sets. This is the default.

**ON**

SMC `IDAX` parameters and SMC DFSMS esoteric substitution (if specified) apply to all `DISP=MOD` data sets when the first reference to the data set in the job's `JCL` specifies either `DISP=MOD` or `DISP=NEW`.

---

---

**Note:** During SMS `IDAX` substitution, the SMC cannot determine whether a `MOD` data set exists.

---

---

## LIMIT

**Interfaces:**

- Console, utility, `SMCCMDS` data set, or `SMCPARMS` data set
- UUI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required. Not allowed as input to the `SMCUSIM` utility.

## Description

The SMC `LIMIT` command enables you to limit resource usage in the SMC subsystem. This command can limit exposure to Sx78 type abends.

By default, the SMC subsystem starts a new SMC work task whenever a new request is received but no free SMC work task is available. This default behavior manifests using the default `LIMIT` settings of `LIMIT AUTO OFF` and `LIMIT MAXTASKS OFF`.

However, depending upon the SMC subsystem work load, the SMC subsystem may attempt to start work tasks beyond the limits of SMC subsystem resource availability. This may cause Sx78 type abends due to virtual memory constraints. Such abends may lead to misdirected allocation, lost mount conditions, or loss of the SMC subsystem altogether.

If you specify `LIMIT AUTO ON` or `LIMIT MAXTASKS nnn` to override the default SMC subsystem behavior, then SMC queues requests for new work when the calculated or specified SMC work task limit is reached.

- `LIMIT AUTO ON` causes SMC to use a calculated work task limit based upon SMC subsystem availability.
- `LIMIT MAXTASKS nnn` causes SMC to use the specified work task limit.

The calculated or specified SMC work task limit acts to slow the processing for new requests. For example, if the calculated or specified SMC work task limit is 200, and a new request is received when 200 SMC work tasks are already active, then the new request is queued and waits until either an existing SMC work task completes its processing, or the `LIMIT MAXIDLE nnn` time elapses.

If the `LIMIT MAXIDLE nnn` time elapses and no existing SMC work task completes its processing, then the SMC starts a new work task to process the new request even though the number of work tasks now exceeds the calculated or specified work task limit. Therefore, the calculated or specified work task limit is a "soft" limit.

The default `LIMIT MAXIDLE` time is ten seconds. Therefore, the number of additional SMC work tasks above the calculated or specified work task limit is limited to one new work task every ten seconds until existing work tasks complete their processing.

---

---

**Note:** Oracle recommends that you only use the `LIMIT` command to change the default `LIMIT` settings if your SMC subsystem experiences Sx78 type abends.

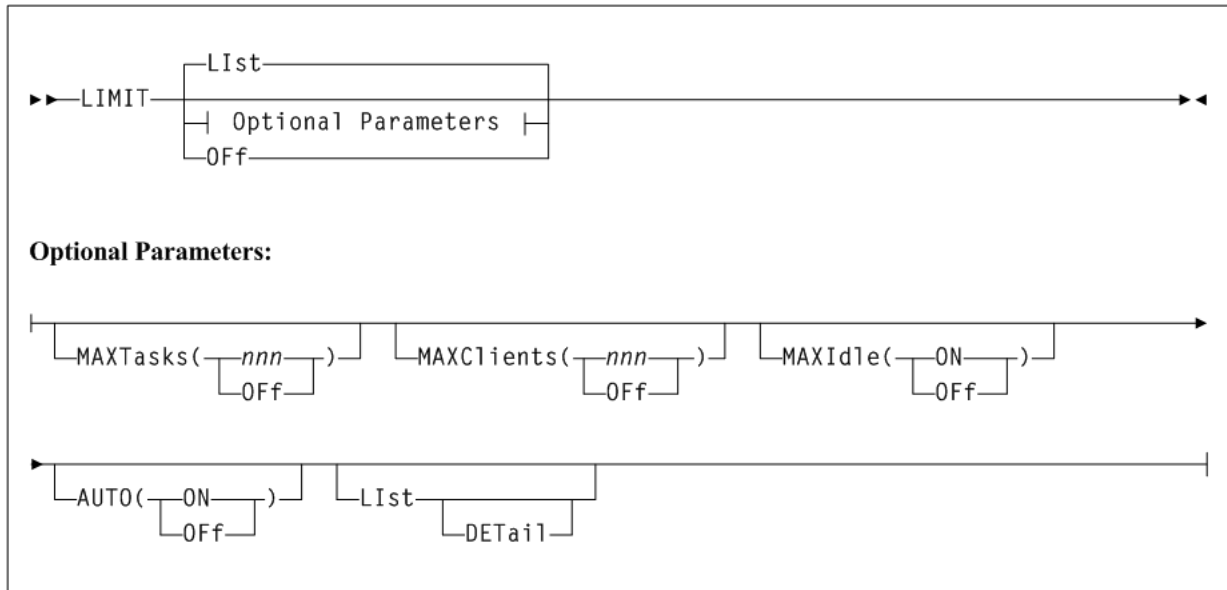
---

---

## Syntax

Figure 2–19 shows `LIMIT` command syntax.

Figure 2–19 LIMIT command syntax



## Parameters

As shown in [Figure 2–19](#), the `LIMIT` command includes the following parameters:

### MAXTasks

limits the number of work tasks within the SMC subsystem. Work tasks process operator commands, allocation queries, mount and dismount requests, and TCP/IP connections to remote hosts, and other system processes. Typically these requests are short lived and their resources are reusable.

#### nnn

the maximum number SMC subsystem work tasks, between 16 and 500.

#### OFF

specifies no `MAXTasks` limit. The default value is `OFF` which implies the maximum `MAXTasks` limit of 500 work tasks.

### MAXClients

optionally, limits the number of work tasks within the SMC subsystem that can be connected to a remote server at once. These work tasks are a subset of `MAXTASKS`.

The `LIMIT MAXClients` command is different from the `HTTP MAXCLIENTS` command:

- The `LIMIT MAXClients` limits the number of client connections on the client side of client/server.
- The `HTTP MAXClients` limits the number of client connections on the server side of client/server.

#### nnn

The number of SMC subsystem socket I/O work tasks between 1 and 500.

#### OFF

specifies no `MAXClients` limit. The default value is `OFF` which implies the maximum `MAXCLIENTS` limit of 500 SMC work tasks.

**MAXIdle**

optionally, limits the amount of time that the SMC subsystem waits before starting a new work task in excess of the calculated or specified work task limit when new requests are received.

**nnn**

The number of seconds between 0 and 2400. The default is 10 seconds.

**OFF**

Restores the default value of 10 seconds.

**AUTO**

optionally, specifies whether the SMC subsystem automatically sets a `MAXTasks` limit based on the amount of resources available.

**ON**

specifies that the SMC subsystem will automatically set a `MAXTasks` limit if not overridden by the `MAXTasks` parameter.

**OFF**

Specifies that the SMC subsystem will not automatically set a `MAXTasks` limit.

**OFF**

optionally, resets all `LIMIT` settings to their default values.

**List**

optionally, lists the current `LIMIT` parameter settings. If `LIMITS` is entered without any command parameters, a `List` is performed.

**DETail**

optionally, lists SMC resource usage and calculated maximum number of work subtasks supported.



**Note:**

- When `LIMIT AUTO ON` is specified, the SMC calculates the maximum number of SMC work tasks that it can support. This calculation is performed whenever the SMC starts a new work task that results in a new hi-water SMC work task count.
- The calculation for maximum number of SMC work tasks is based on storage usage in various subpools. However, the actual maximum number of SMC work tasks than can be supported in practice is based on the types and timing of requests, and general system workload. The calculation only results in a rough estimate which can change as new subtasks are started.
- `LIMIT MAXTASKS nnn` overrides the `LIMIT AUTO ON` or `LIMIT AUTO OFF` setting. Therefore, if both `LIMIT MAXTASKS 190` and `LIMIT AUTO ON` are specified, SMC limit processing is triggered when the number of SMC work tasks reaches 190, regardless of the SMC subsystem calculation of the maximum number of work tasks.
- `LIMIT MAXCLIENTS` is not the same as `HTTP MAXCLIENTS`. `LIMIT MAXCLIENTS` imposes restrictions on TCP/IP socket connections on the client side of client-server while `HTTP MAXCLIENTS` imposes restrictions on TCP/IP socket connections on the server side of client-server.

## Llst

**Interfaces:**

- Console, utility, `SMCCMDS` data set, or `SMCPARMS` data set
- UUI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required, or may be input to the `SMCUSIM` utility

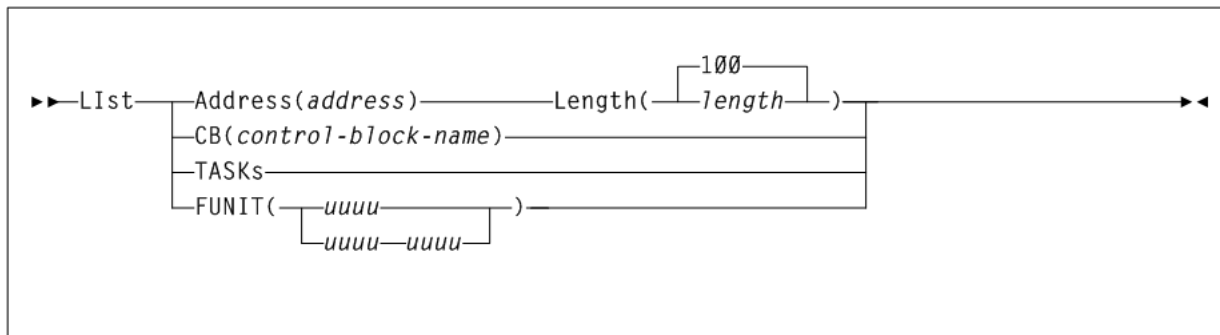
## Description

The SMC `Llst` command displays storage accessible from the SMC started task address space.

This command is intended to be used primarily as directed by Oracle StorageTek Software Support. Output is intended for debugging and support purposes.

## Syntax

Figure 2–20 shows `Llst` command syntax.

**Figure 2–20** *List command syntax*

## Parameters

As shown in [Figure 2–20](#), the `LIST` command includes the following parameters:

### **Address(address)**

optionally, specifies the address at which to begin listing SMC memory contents.

*address* indicates the address. The value entered must be a valid hexadecimal address.

### **Length(length)**

optionally, specifies the length of SMC memory (bytes) to be listed. `Length` is required if `Address` is specified.

*length* indicates the value entered must be a valid hexadecimal value between 1 and FFFF. Memory is listed beginning at the location specified in the `Address` parameter. The default value is 100 (decimal 256).

### **CB(control-block-name)**

optionally, specifies the internal SMC control block to be listed.

*control-block-name* indicates the control block name.

SMC control blocks are listed for diagnostic purposes. Specify *control-block-name* only as directed by Oracle StorageTek Software Support.

### **TASKs**

optionally, lists the tasks in the SMC subsystem address space.

### **FUNIT(uuuu or uuuu-uuuu)**

optionally, lists SMC control blocks associated with the specified unit address(es).

*uuuu* or *uuuu-uuuu* indicates a unit address or range of unit addresses.

## LOG

### **Interfaces:**

- Console, utility, `SMCCMDS` data set, or `SMCPARMS` data set
- UI Support: Yes (No XML/CSV output)

### **Subsystem Requirements:**

Active SMC required

## Description

The SMC LOG command logs data communications from either or all of the following:

- SMC Library Interface Component
- SMC Client Configuration Component
- SMC HTTP Server Component
- SMC Operator Command Component

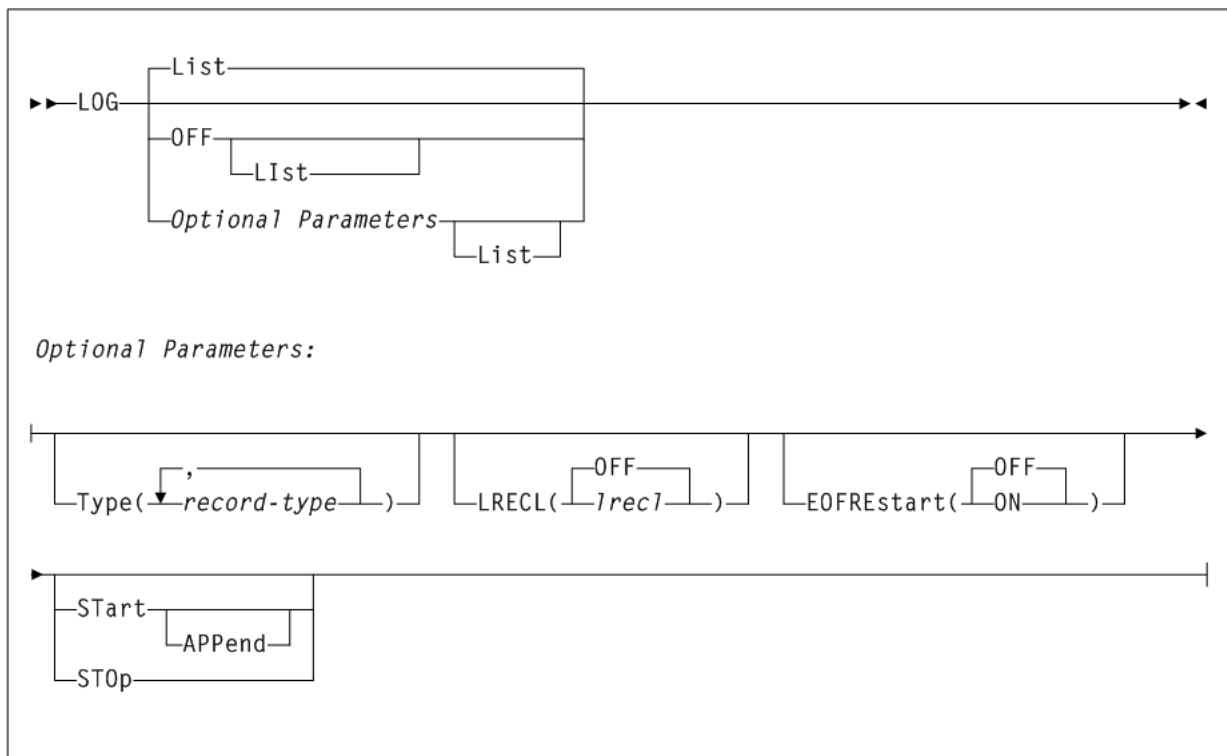
These communication transactions are logged to the SMC log file.

This command is intended to be used primarily as directed by Oracle StorageTek Software Support.

## Syntax

Figure 2–21 shows LOG command syntax.

**Figure 2–21 LOG command syntax**



## Parameters

As shown in Figure 2–21, the LOG command includes the following parameters:

### List

optionally, displays HTTP Server status information and interval statistics.

- LIst is the default when no parameters are specified on the LOG command.
- LIst may be specified with other parameters. In this case, the LIst is applied after the other parameters are processed.

**OFF**

optionally, resets all LOG settings to original SMC default settings. OFF closes the LOG, turns off all record TYPES, and sets LRECL to 0. If a subsequent LOG START command is issued after the OFF command, no records are logged until specific LOG TYPES are specified.

**Optional Parameters****Type(record-type)**

Optionally, specifies what record types are to be logged.

*record-type* indicates one of the following record types:

**LLS**

For local LIBTRANS XML transactions sent from the SMC Library Interface Component

**LLR**

For local LIBTRANS XML responses received by the SMC Library Interface Component

**CLS**

For remote LIBTRANS XML transactions sent from the SMC Client Communication Component

**CLR**

For remote LIBTRANS XML responses received by the SMC Client Communication Component

**CUS**

For remote UUI transactions sent from the SMC Client Communication Component

**CUR**

For remote UUI responses received by the SMC Client Communication Component

**CER**

For all errors detected by the SMC Client Communications Component

**SAS**

For all data sent outbound by the SMC HTTP Server Component

**SAR**

For all data received inbound by the SMC HTTP Server Component

**SER**

For all errors detected by the SMC HTTP Server Component

**CMD**

For all operator commands received by the SMC Operator-Command Component

**ALL**

Sets all types ON.

**OFF**

Sets all types OFF.

**LRECL**

optionally, specifies the line length written to the LOG file.

**lrecl**

a value from 80 to 32767, used to limit the logical line length.

If the data sent or received exceeds the specified LRECL, then multiple lines are written. If LRECL is specified, then the log date and time stamp are written as separate LOG records. If this value exceeds the DCB LRECL, then the DCB LRECL is used.

**OFF**

The maximum line length is controlled by the DCB LRECL. The LOG record is appended to the log date and time stamp. This is the default.

**EOFREstart**

optionally, specifies the response when the LOG file reaches End of File (EOF).

**OFF**

Logging stops at EOF.

**ON**

The LOG file closes and re-opens at EOF, and logging continues.

**STArt**

optionally, starts logging.

**APPend**

optionally, specifies that data is appended to the current LOG file when logging is started. If not specified, then data is added to the LOG file from the beginning of the log file, and any data currently in the log file is lost. APPend can only be specified when STArt is also specified.

**STOp**

optionally, stops logging. Current LOG settings are preserved.

## METAdata

**Interfaces:**

- Utility only
- UUI Support: Yes

**Subsystem Requirements:**

Active SMC required

## Description

The SMC METAdata command displays XML tags associated with a function that produces XML output.

---



---

**Note:** No text output is produced by the METAdata command. If neither XML nor CSV output is requested, the command will not produce any output.

---

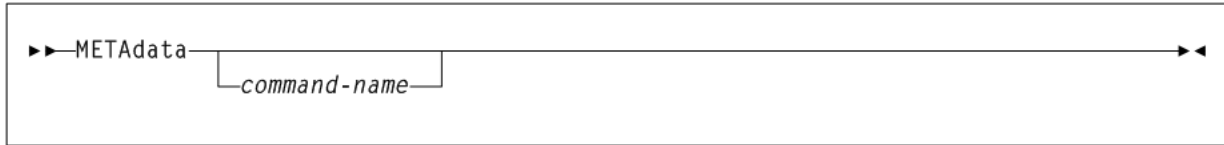


---

## Syntax

Figure 2–22 shows METAdata command syntax.

Figure 2–22 METAdata command syntax



## Parameters

As shown in Figure 2–22, the METAdata command includes the following parameters:

### **command-name**

The command for which metadata is to be produced.

When the command contains "two parts" such as `Display Volume`, both portions should be entered. The METAdata command uses the same abbreviations for the *command-name* as the command itself, for example:

```
METAdata D V
```

The METAdata command itself does not support metadata.

The following tags are produced by the METAdata command:

#### **<command\_name>**

The full name of the command.

#### **<security\_level>**

Required security level if command authorization is in effect. Values are `QUERY`, `SET`, and `ADMIN`.

#### **<command\_tags>**

Header tag for all XML tags.

The following tags are produced for each XML tag in the requested command:

#### **<tag\_data>**

Header tag for each XML tag.

#### **<tag\_name>**

The XML tag name, for example, `volser`.

#### **<tag\_type>**

Values are header (header XML tag), data (XML data tag, normally associated with a value), and trailer (indicates the position of the trailer tag associated with a header tag).

#### **<occurrences>**

The maximum number of expected occurrences, for tags that may occur multiple times. This may be either a numeric value or `unlimited`.

The following tags are produced only for data-type tags:

**<data\_type>**

Indicates the type of data expected in the tag value. Produced only for data tags. Values include: char, numeric, flag, hex, date, time.

**<maximum\_size>**

Indicates the maximum output data size.

## MONitor

**Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

- Active SMC required
- Cannot be input to the SMCUSIM utility

## Description

The SMC MONitor command defines parameters associated with SMC monitoring.

---

---

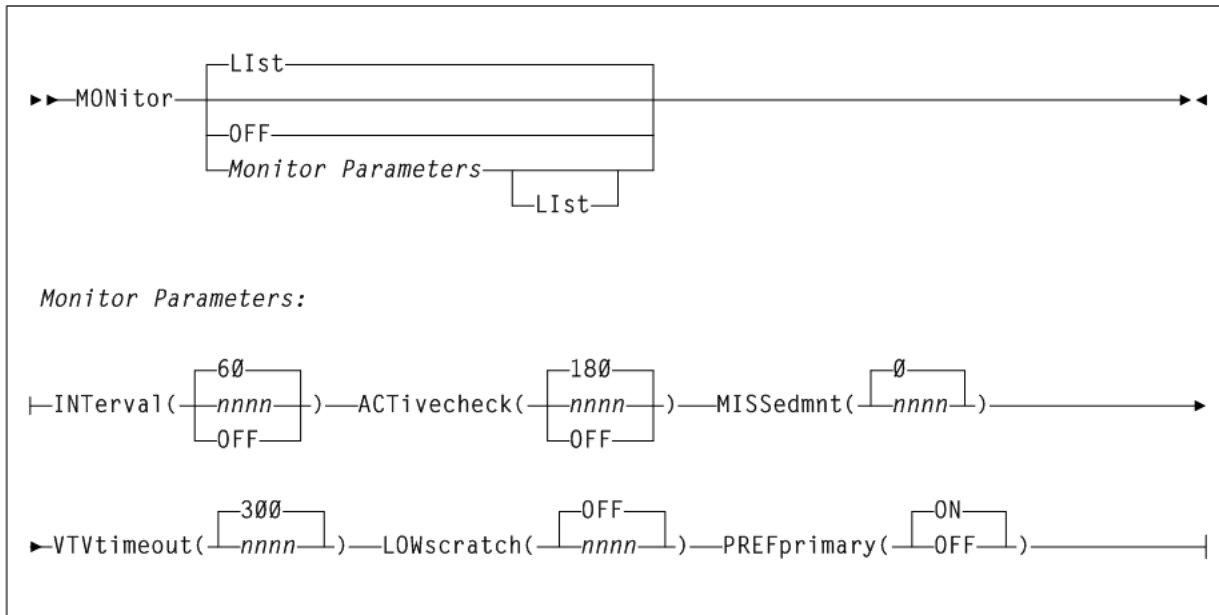
**Note:** SMC cannot detect pending mounts when either of the following conditions are present:

- The job entry subsystem is JES3,
  - The mount is outstanding on a JES3 LOCAL processor,
  - ALLOCDEF DEFER(OFF) has been specified.
  - The mount was requested before SMC had initialized and the mount request did not request the DEFER option.
- 
- 

## Syntax

Figure 2-23 shows MONitor command syntax.

Figure 2–23 MONitor command syntax



## Parameters

As shown in Figure 2–23, the MONitor command includes the following parameters:

### List

Optionally, lists the current MONitor command settings.

### OFF

Optionally, resets all MONitor parameters to their default values.

### Monitor Parameters

#### INTerval

Optionally, specifies the base SMC monitor scan interval in seconds. The default value is 60.

At each monitor interval, SMC performs the following actions:

1. SMC attempts to find an active communication path for any inactive TapePlex or Storage Manager.
2. If PREFprimary(ON) is set, SMC attempts to change the communication path from a non-primary path to the primary server.
3. SMC checks to ensure that the SMC IEFJFRQ exit is active, so that SMC is influencing tape allocation.
4. SMC clears orphaned task tokens.
5. SMC attempts to redrive pending mounts.

---

**Note:** It is strongly recommended that you do not set the MONitor INTerval to 0 or OFF. With these values set, SMC is not able to re-drive pending mounts, honor the PREFprimary setting automatically, or verify that the IEFJFRQ exit is active.

---



**ACTivecheck(nnnn)**

Optionally, specifies the interval in seconds after which SMC should verify that communication is still accessible on an active communication path that has had no activity since the prior active check.

*nnn* indicates the interval in seconds. The default value is 180, or 3 times the `INTerval` value.

If `ACTivecheck` is set to 0 or `OFF`, SMC does not monitor active communication paths. If the `ACTivecheck` interval is not a multiple of the base `INTerval`, it is rounded up to a number of seconds that is a multiple of the `INTerval` value.

**MISSEdmt(nnnn)**

Optionally, specifies the interval in seconds after which SMC should issue a message indicating that an outstanding mount is pending. The message is repeated after each interval as long as the mount remains unsatisfied.

*nnnn* indicates the interval in seconds. The default value is 0, meaning that the message does not appear.

If the `MISSEdmt` interval is not a multiple of the base `INTerval`, it is rounded up to a number of seconds that is a multiple of the `INTerval` value. It is recommended that you set this interval to at least 600 (10 minutes) to allow the SMC mount monitor processing to attempt to redrive missing mounts before producing the periodic missed mount message.

**VTVtimeout(nnnn)**

Optionally, specifies the time in seconds (*nnnn*) after which SMC will issue the `SMC0231` message and attempt to re-drive a mount request for a VTV.

*nnnn* indicates the time in seconds. The default value is 300 seconds (5 minutes).

If the `VTVtimeout` value is not a multiple of the base `INTerval`, it is rounded up to a number of seconds that is a multiple of the interval value.

The minimum value of `VTVtimeout` is 300 seconds; the maximum is 1800 seconds.

**LOWscratch(nnnn)**

Optionally, specifies the interval in seconds after which SMC should report on server scratch subpools that are below their threshold value.

*nnnn* indicates the interval in seconds. The default value is `OFF`.

`LOWSCRatch` does not report on scratch shortages that are not associated with a subpool name, except for scratch subpool zero.

If `LOWSCRatch` is set to 0 or `OFF`, SMC does not perform scratch threshold checking.

If the `LOWSCRatch` interval value is not a multiple of the base `INTerval`, it is rounded up to a number of seconds that is a multiple of the `INTerval` value.

**PREFprimary**

Optionally, indicates whether SMC attempts to establish communication on a higher priority server path when communication is currently active on a server that is not the first defined server for a TapePlex. The default value of `ON` indicates that at each `MONitor INTerval`, SMC attempts to communicate on the first defined server path if that is not the current active path.

If `PREFprimary` is set to `OFF`, then SMC continues to communicate on an active path even if one defined earlier becomes available.

The `PREFprimary` setting also controls the behavior of the `RESYNChronize` command. When `PREFprimary(ON)` is set, a `RESYNChronize` attempts to communicate to the first defined server. When `PREFprimary(OFF)` is set, a `RESYNChronize` does not change the current active communication path, if one exists.

## M0unt

### Interfaces:

- Console or utility
- UI Support: Yes (No XML/CSV output)

### Subsystem Requirements:

Active SMC required, or may be input to the `SMCUSIM` utility

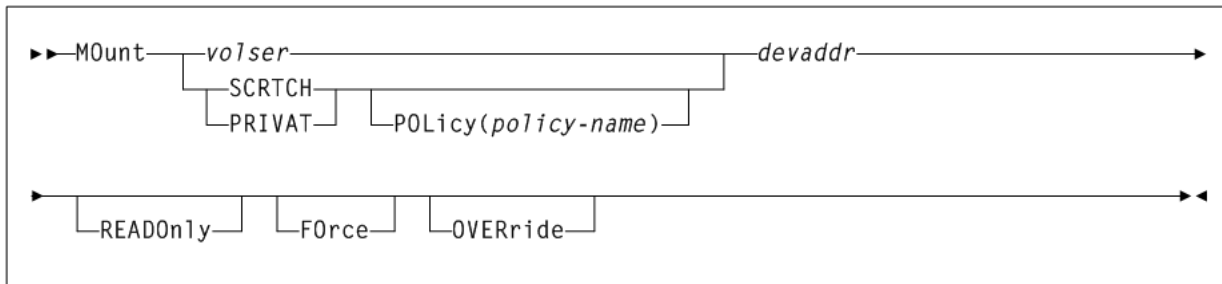
## Description

The SMC `M0unt` command requests a mount of a specific or scratch volume on a tape device.

## Syntax

Figure 2–24 shows `M0unt` command syntax.

Figure 2–24 *M0unt* command syntax



## Parameters

As shown in Figure 2–24, the `M0unt` command includes the following parameters:

### **volser**

the volume serial number of the volume to be mounted.

### **SCRTCH or PRIVAT**

indicates a scratch mount.

### **POLicy(policy-name)**

optionally, when the volume is specified as `SCRTCH` (or `PRIVAT`), specifies the name of an SMC policy that defines the subpool name and management class name (if applicable) to be applied for the scratch mount.

*policy-name* indicates the policy name.

### **devaddr**

the MVS device address for which the volume is to be mounted. The device must be defined by a TapePlex known to SMC. This is a required value.

**READOnly**

optionally, indicates that the volume is to be mounted with write protect enabled.

**FOrce**

optionally, indicates that if a loaded volume is found on the drive, it should be rewound and unloaded prior to attempting the requested mount. **FOrce** is not supported for virtual drives.

**OVERride**

optionally, indicates that if SMC detects a pending mount for the drive, indicates that the volume specified in the **MOunt** command should override the volume indicated in the pending mount.

## MOUNTDef

**Interfaces:**

- Console, utility, *SMCCMDS* data set, or *SMCPARMS* data set
- UI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

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

## Description

The SMC **MOUNTDef** command enables you to control message processing options (mounts or dismounts) previously controlled by the HSC **MNTD** command, HSC input parameters and **LIBGEN** options. The **MOUNTDef** command also applies to VTVs.

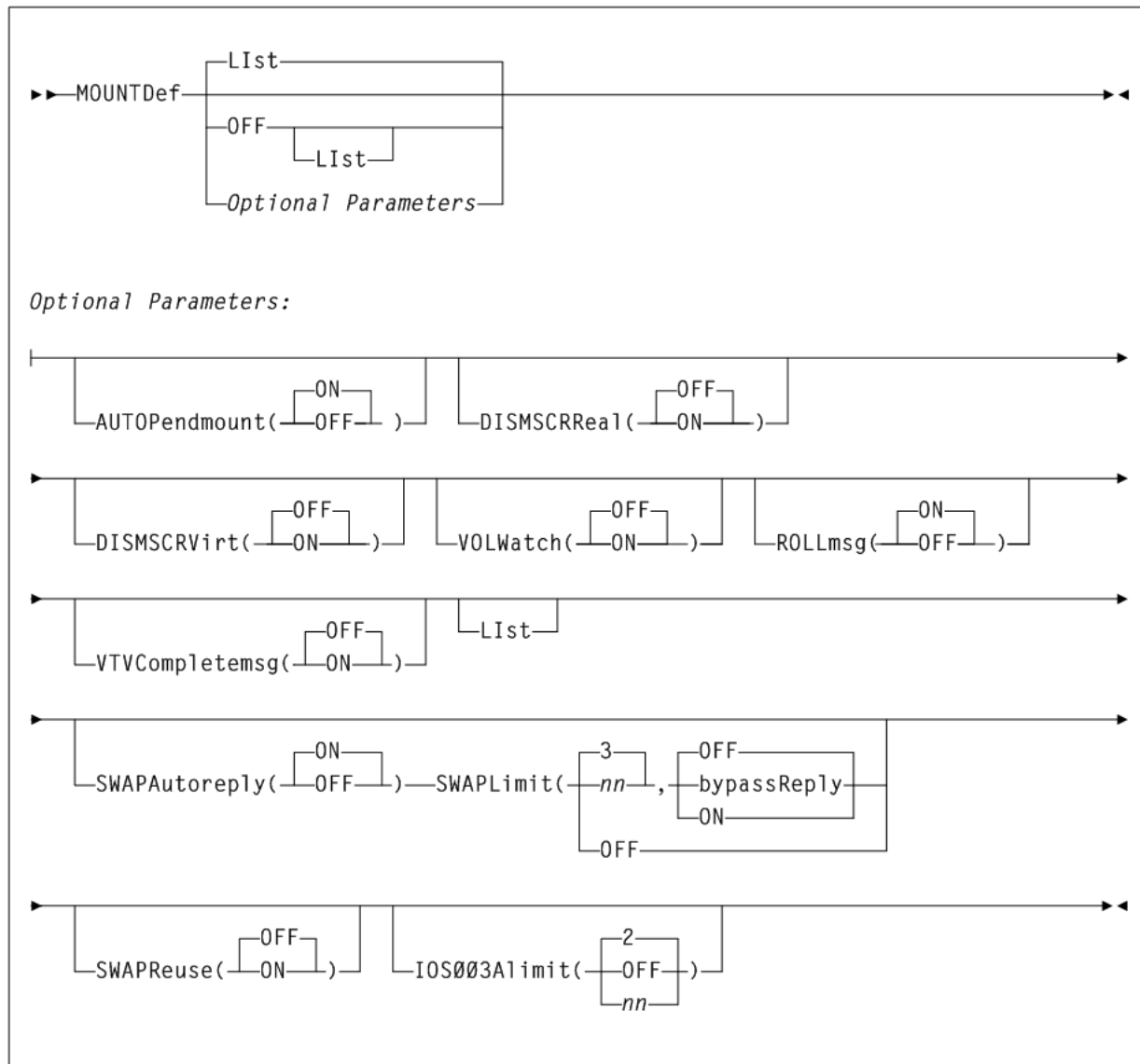
These options control the following:

- automation of pending mounts
- delete disposition at dismount time
- whether mount and dismount messages are issued
- when mount messages are rolled off of the console
- swap processing behavior

## Syntax

Figure 2–25 shows **MOUNTDef** command syntax.

Figure 2–25 MOUNTDef command syntax



## Parameters

As shown in Figure 2–25, the MOUNTDef command includes the following parameters:

### Lst

optionally, lists current SMC MOUNTDef settings.

- Lst is the default when no parameters are specified on the MOUNTDef command.
- Lst may be specified with other parameters. In this case, the Lst is generated after the other parameters are processed.

### OFF

optionally, resets all MOUNTDef settings to original SMC default settings. Specify Lst with this parameter to list these settings.

## Optional Parameters

### **AUTOPendmount**

optionally, specifies whether the pending mounts are automated for library or virtual drives found when SMC is first initialized and the owning TapePlex is first activated.

#### **ON**

Pending mounts are automated. This is the default.

#### **OFF**

Pending mounts are not automated.

---

---

**Note:** If `AUTOPendmount` is `OFF`, you must issue an `HSC MOUNT` command to complete any pending mounts found when a TapePlex is first activated.

---

---

### **DISMSCRReal**

optionally, specifies how the SMC interprets the delete disposition on a dismount message for real (nonvirtual) volumes.

---

---

**Note:** The `DISMSCRReal` and `DISMSCRVirt` parameters replace the `DELDISP=` parameter of the `HSC LIBGEN SLILIBRY` macro.

---

---

#### **OFF**

The real cartridge is retained as nonscratch when MVS indicates delete disposition in the dismount message. This is the default, and is recommended for CA-1 or CA-TLMS users to leave volumes in a consistent nonscratch status during the CA-1 and CA-TLMS grace period.

#### **ON**

The real cartridge is scratched when MVS indicates delete disposition in the dismount message.

### **DISMSCRVirt**

optionally, specifies how the SMC interprets the delete disposition on a dismount message for VTVs.

---

---

**Note:** The `DISMSCRReal` and `DISMSCRVirt` parameters replace the `DELDISP=` parameter of the `HSC LIBGEN SLILIBRY` macro.

---

---

#### **OFF**

The VTV is retained as nonscratch when MVS indicates delete disposition in the dismount message. This is the default.

#### **ON**

The VTV is scratched when MVS indicates delete disposition in the dismount message.

### **VOLWatch**

optionally, specifies whether an SMC message is issued when a library volume is to be mounted on a nonlibrary drive.

**OFF**

No message is issued when a library volume is to be mounted on a nonlibrary drive.

**ON**

A console message is issued when a library volume is to be mounted on a nonlibrary drive.

**ROLLmsg**

optionally, specifies whether mount messages handled by the SMC are allowed to roll off the operator console before mount requests are satisfied.

**ON**

Mount messages are allowed to roll off the console screen before mounts are satisfied. This is the default.

You can limit the rolling of specific mount messages (for example, TMS007) by presetting the messages xxxxRCDC (Request to Change Descriptor Codes) flag word xxxxBFB1. The SAMPLIB includes a sample MPF USEREXIT to preset the xxxxRCDC flag.

**OFF**

Mount messages remain on the console screen until mount requests are satisfied.

**VTVCompletemsg**

optionally, specifies whether VTV mount and dismount messages are issued.

**OFF**

VTV mount (SLS5075I) and VTV dismount (SLS5076I) messages are not issued on the host where the server is running. This is the default.

**ON**

The VTV mount (SLS5075I) and VTV dismount (SLS5076I) messages are issued on the host where the server is running.

**SWAPAutoreply**

optionally, specifies whether the SMC enhanced swap process automatically responds NO to the IBM IGF500D - REPLY 'YES' DEVICE OR 'NO' message when no compatible swap-to device is found.

**ON**

SMC automatically replies NO to the IBM IGF500D - REPLY 'YES' DEVICE OR 'NO' message after the SMC fails to find compatible devices and issues its own SMC0108 No compatible drive found for SWAP processing message. This is the default.

**OFF**

SMC does not automatically reply to the IBM IGF500D - REPLY 'YES' DEVICE OR 'NO' message after the SMC fails to find compatible devices.

**SWAPLimit**

optionally, specifies the maximum number of system initiated swaps that SMC influences for the same job step and volume serial.

**nn**

a swap value of 0-99, or OFF.

- If the value specified is 0 or OFF, then the SMC continues to attempt to influence the swap indefinitely.
- If the value is not specified, the default value is 3.
- For any SWAPLimit value other than 0 or OFF, SMC stops attempting to influence the swap after *nn* attempts for the same job step and volume serial.

---

**Note:** Any operator-initiated SWAP commands are not applied to the SWAPLimit value.

---

### bypassReply

Substitute one of the following values for *bypassReply*:

- ON optionally, indicates that SMC should not reply to the system message IGF509D if another swap attempt is detected after the SWAPLIMIT count is exhausted.
- OFF optionally, SMC generates a response of NO to an IGF509D message after *nn* swap attempts. This is the default.

### SWAPReuse

optionally, specifies whether previously used swap-to devices are to be excluded as possible swap-to devices for the current job step and volume serial.

#### OFF

Previously used swap-to devices are excluded as possible swap-to devices for the current job step and volume serial. If no eligible device is found then SMC issues message SMC0108 and if the MOUNTDef SWAPAutoreply is set to ON then the IGF500D or IGF509D message is replied to with NO. This is the default.

#### ON

Previously used swap-to devices are eligible to be used for the current job step and volume serial.

### IOS003Alimit

optionally, specifies the maximum number of attempts to re-drive a mount request that has failed with an IOS003A message.

#### OFF

perform an unlimited number of attempts to re-drive the mount request for a given drive following an IOS003A message for a given drive.

#### nn

the maximum number of attempts to re-drive the mount request. Valid values are 0-99. The default is 2. A value of 0 is equivalent to IOS003Alimit (OFF).

## MSGDef

### Interfaces:

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UI Support: Yes (No XML/CSV output)

### Subsystem Requirements:

Active SMC required, or may be input to the SMCUSIM utility

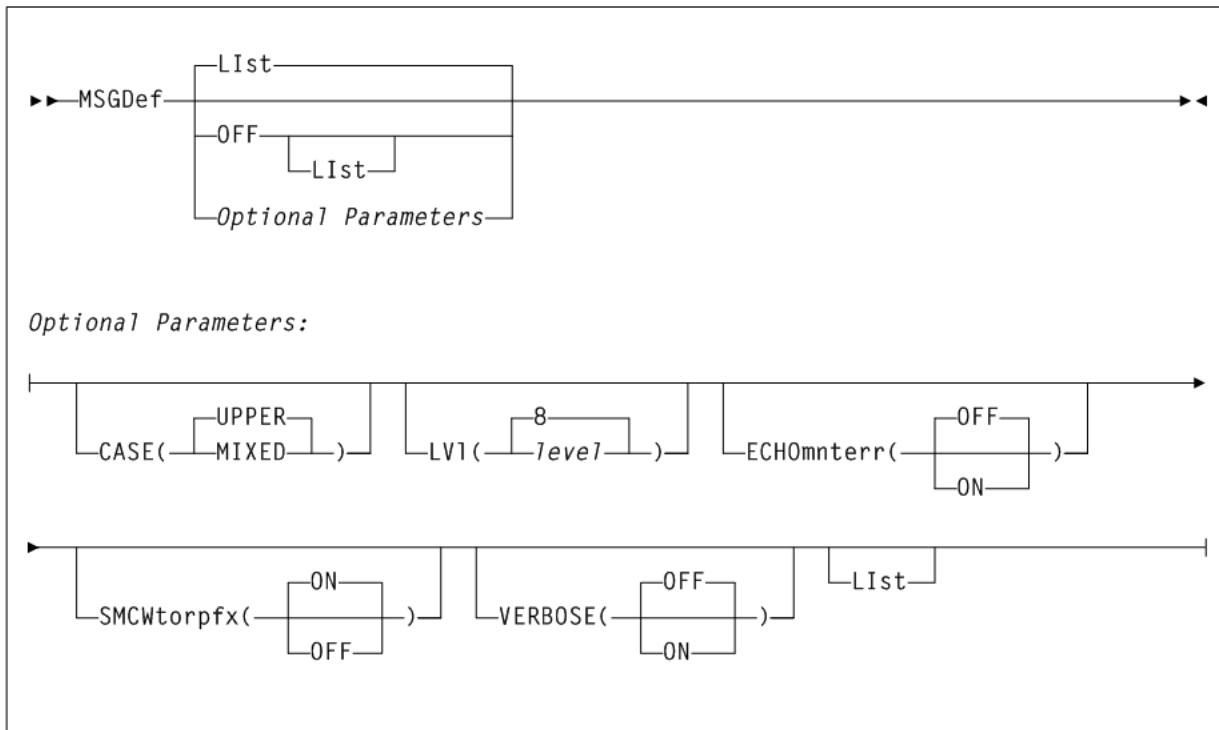
## Description

The SMC `MSGDef` command defines the appearance of SMC system messages, and controls which messages are displayed and suppressed.

## Syntax

Figure 2–26 shows `MSGDef` command syntax.

Figure 2–26 *MSGDef* command syntax



## Parameters

As shown in Figure 2–26, the `MSGDef` command includes the following parameters:

### LIsT

optionally, lists current default SMC message settings.

- `LIsT` is the default when no parameters are specified on the `MSGDef` command.
- `LIsT` may be specified with other parameters. In this case, the `LIsT` is generated after the other parameters are processed.

### OFF

optionally, resets all `MSGDef` values to original SMC default settings. Specify `LIsT` with this parameter to list these settings. `MSGDef` cannot be specified with any other parameter.

### Optional Parameters

#### CASE

optionally, specifies the message case. Valid values are `UPPER` or `MIXED`.



**UPPER**

specifies upper case. This is the default.

**MIXED**

specifies mixed case.

**LVI**

optionally, specifies the default level used to control which SMC messages are displayed and suppressed.

**level**

the default level. Valid values include the following:

- 0 - Display error messages only.
- 4 - Display error and warning messages from the SMC subsystem.
- 8 - Display all SMC subsystem messages and allocation job log warning messages. This is the default if the MSGDef parameter is not specified.

---

---

**Note:** Levels higher than 8 are used for diagnostic purposes and should only be specified as directed by Oracle StorageTek Software Support.

---

---

**ECHOmnterr**

optionally, specifies whether mount errors generated by the HSC are echoed directly to the console for the SMC client.

**ON**

Mount errors generated by the HSC are echoed to the console for the SMC client. This parameter is valid for both local and remote servers.

**OFF**

Mount errors generated by the HSC are not to the console for the SMC client. This is the default.

**SMCWtorpfx**

optionally, specifies whether the SMC0137 prefix is displayed for HSC WTOR messages generated by WTORDEST (CLIENT) on the TAPEplex command.

**ON**

The SMC0137 prefix is displayed. This is the default.

**OFF**

The SMC0137 prefix is not displayed.

**VERBOSE**

optionally, specifies whether SMC0190 and SMC0191 messages are displayed whenever any SMC settings are altered.

**ON**

SMC0190 and SMC0191 messages are displayed.

**OFF**

SMC0190 and SMC0191 messages are not displayed.

## MSGJob

### Interfaces:

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (No XML/CSV output)

### Subsystem Requirements:

Active SMC required, or may be input to the SMCUSIM utility

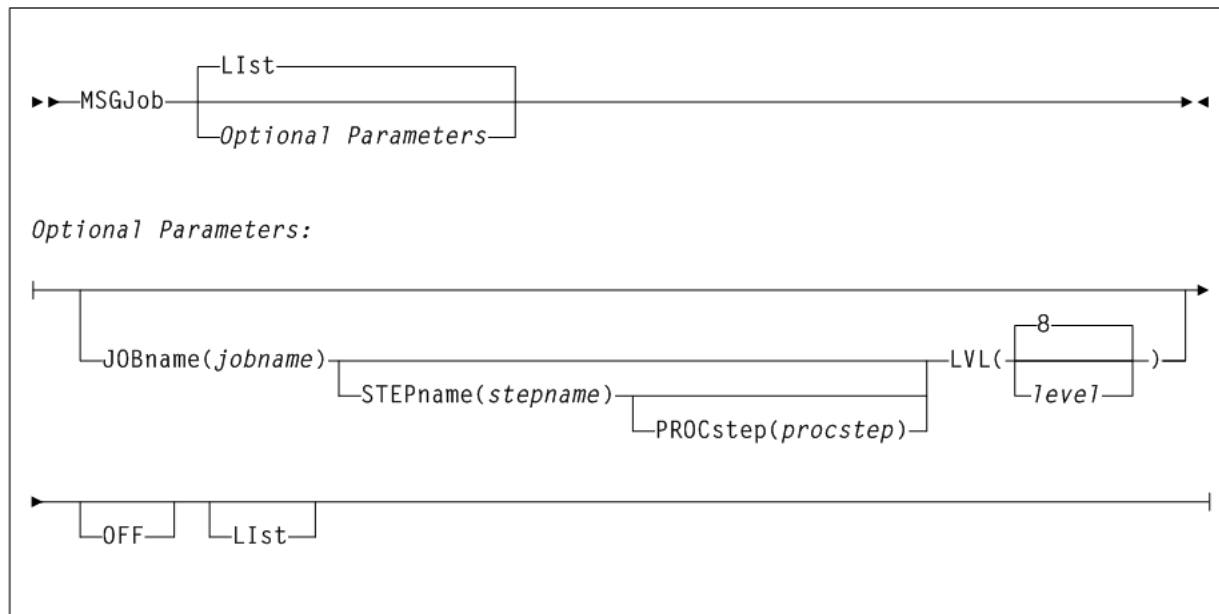
## Description

The SMC MSGJob command enables you to override of the default message level by job name, step name, and PROC step.

## Syntax

Figure 2–27 shows MSGJob command syntax.

Figure 2–27 MSGJob command syntax



## Parameters

As shown in Figure 2–27, the MSGJob command includes the following parameters:

### LIst

optionally, lists current default SMC message settings and job name, step name, and PROC step exceptions (in the order of evaluation).

- LIst is the default when no parameters are specified on the MSGJob command.
- LIst may be specified with other parameters. In this case, the LIst is generated after the other parameters are processed.

## Optional Parameters

### **JOBname(jobname)**

optionally, specifies a job name.

*jobname* indicates the job name. The value entered must be one to eight characters in length. An asterisk (\*) can be used as a wildcard when included as the last character. In this case, the message level is set for all jobs whose job names match the characters preceding the asterisk.

### **STEPname(stepname)**

optionally, specifies a step name. It is only valid when **JOBname** is also specified (may be **JOB=\*).**

*stepname* indicates the step name. The value entered must be one to eight characters in length. This parameter is only valid if **JOBname** is also specified.

### **PROCstep(procstep)**

optionally, specifies a PROC step. It is only valid when **JOBname** and **STEPname** are also specified.

*procstep* indicates the PROC step. The value entered must be one to eight characters in length. This parameter is only valid if **JOBname** and **STEPname** are also specified.

### **LVL(level)**

optionally, specifies the default level used to control which SMC messages are displayed. This parameter is required when **JOBname** is specified.

level indicates the default level. Valid values include the following:

- 0 - Display error messages only.
- 4 - Display error and warning messages from the SMC subsystem.
- 8 - Display all SMC subsystem messages and allocation job log warning messages. This is the default if the **MSGDef** parameter is not specified.

---



---

**Note:** Levels higher than 8 are used for diagnostic purposes and should only be specified as directed by Oracle StorageTek Software Support.

---



---

### **OFF**

optionally, removes **MSGJob** overrides.

- If no other parameters are specified, **MSGJob** overrides are removed for all jobs.
- If only **JOBname** is specified, **MSGJob** overrides are removed for all **MSGJob** entries for that job name.
- If **JOBname**, **STEPname** and **PROCstep** are specified, **MSGJob** overrides are removed only for the specified entry.

The global **MSGJob** is not affected by this parameter.

## POLicy

### **Interfaces:**

- Console, utility, **SMCCMDS** data set, or **SMCPARMS** data set
- UI Support: Yes (supports XML and CSV)

**Subsystem Requirements:**

Active SMC required, or may be input to the SMCUSIM utility

**Description**

The SMC POLicy command specifies all tape policy specifications in a single named policy object. It can be used with SMC TAPEREQ statements or SMC DFSMS routines to associate a named policy with an allocation or mount event. Additionally, the POLicy VALidate option can be used to validate policy objects.

---

---

**Note:** Refer to Oracle's ELS publication *Configuring and Managing SMC* for more information about the use of SMC policies.

---

---

Policies defined using the SMC POLicy command create SMC policy objects that can be referenced by name from either SMC TAPEREQ control statements using the TAPEREQ POLicy parameter. A policy name may be associated with two policy objects: a scratch policy with the specified name, and a specific policy with the specified name. Policies referenced from SMC TAPEREQ control statements must be defined prior to executing the TREQDEF command that processes the TAPEREQ control statements.

By default, and SMC POLICY command applies ONLY to SCRATCH volumes. If you want to set a policy for specify volumes, you must specify VOLTYPE (SPECIFIC). If you want to specify a policy to apply to both SCRATCH and SPECIFIC volumes, you must specify two separate POLICY commands with the same POLICY name.

When a policy to be added to the SMC contains a policy specification error, for example, specifying a scratch subpool name with VOLTYPE (SPECIFIC), the SMC still creates the policy object to find it. When a policy is marked as in error, the policy is empty, indicating that the default tape policy attributes will be applied to an event that specifies this policy name.

To update an SMC policy, enter a new SMC POLicy command for the given policy Name and VOLType. Be aware that the new POLicy command replaces the existing scratch or specific policy object. Therefore, when updating a policy, you must enter all values you wish to preserve.

For example, if the original scratch policy object is defined as:

```
POLICY NAME (POL1) VOLTYPE (SCRATCH) MEDIA (VIRTUAL) MGMTCLAS (MGMT1)
```

and you wish to add a subpool to the policy, you must enter:

```
POLICY NAME (POL1) VOLTYPE (SCRATCH) MEDIA (VIRTUAL) MGMTCLAS (MGMT1)  
SUBPOOL (SUBP1)
```

This behavior enables you to modify one or more POLicy commands in a file and reload all policies using the SMC READ command. Note that when using the SMC READ command to reload the SMC POLICY file, it is recommended that you specify the option REPlacepolicy to receive notifications about POLICY commands present in the original definitions but missing in the new definitions.

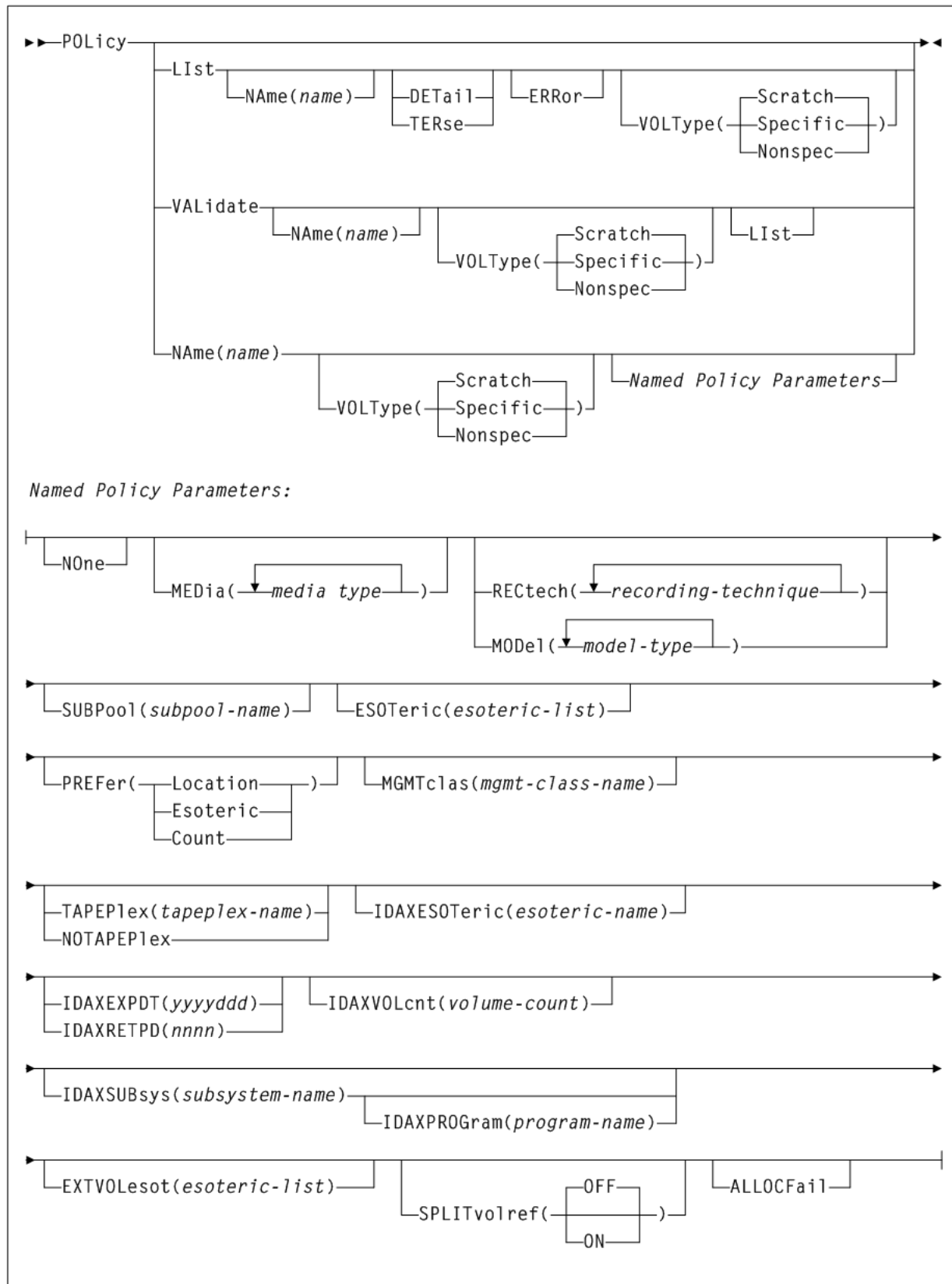
When an SMC policy object to be updated contains a policy specification error, for example, specifying a scratch subpool name with VOLType (SPecific), the SMC will not update any part of the existing policy object. The policy object is not marked in error, and the complete update is discarded.

DEVTYPREF is not supported on this command. However, you can use an esoteric list to specify preferencing for different device types.

## Syntax

Figure 2-28 shows POLicy command syntax.

Figure 2–28 POLicy command syntax



## Parameters

As shown in [Figure 2-28](#), the POLicy command includes the following parameters:

### List

optionally, displays all policy information. Additional parameters allow you to limit which policies are listed, and how much detail is displayed.

If name is also specified, only policies that match the specified name or generic name are listed. If name is not specified, then all policies are listed.

---

**Note:** Matching policies are listed in alphabetic order, regardless of the order in which they were defined to the SMC. When specified with other parameters, the LIst is performed after the other parameters have been processed.

---

### NAme(name)

optionally, specifies the name of a policy that matches the name specified in a TAPEREQ or DFSMS ACS routine management class. If the keyword NAME is omitted, the policy name is assumed to be the first parameter following the POLicy command (if that parameter is not a valid keyword).

*name* indicates the policy name.

### DETail

optionally, displays audit detail information about the policy, including last update date, time, and source. DETail and TERse are mutually exclusive.

### TERse

optionally, lists only policy names and whether each policy name applies to scratch, specific, or all allocations. TERse and DETail are mutually exclusive.

### ERRor

optionally, displays each policy in error status.

When a policy to be added to the SMC contains a policy specification error, for example, specifying a scratch subpool name with VOLTYPE(SPECIFIC), the SMC still creates the policy object. However, the SMC marks the policy object in error, allowing any reference to the named policy object to find it. However, the returned policy will be empty, indicating that the default tape policy attributes will be applied to the event. ERRor lists only those policies that contain an error.

### VOLType

optionally, specifies whether the list applies to scratch or specific policy types.

#### Scratch or Nonspec

List only scratch (nonspecific) policies. This is the default.

#### Specific

List only specific policies.

### VALidate

optionally, validates and lists policy information against the current TapePlex environment. Additional parameters allow you to limit which policies are validated, and whether all results or just warnings and errors are displayed.

**NAme(name)**

optionally, specifies the name of a policy to be validated.

If the last character is an asterisk (\*), all policies that begin with the prefix are displayed. For example, POL VAL NA V\* validates all policies that begin with "V". If the NAme parameter is omitted, a non-keyword parameter following the POLicy command is interpreted as the NAME.

*name* indicates the policy name.

**VOLType**

optionally, specifies whether the list applies to scratch or specific policy types.

**SCrash or Nonspec**

List only scratch (nonspecific) policies. This is the default.

**SPecific**

List only specific policies.

**List**

optionally, specifies whether only warnings and errors are listed, or all tested policies are listed. If not specified, then only a single line for each policy that generates a warning or error message is listed.

**NAme(name)**

optionally, specifies the name of a policy. *name* may be specified with or without the keyword NAme. If the keyword NAme is not specified, the name, if specified, must be the first parameter.

The policy name must match the name specified in a TAPEREQ or DFSMS ACS routine management class name.

*name* indicates the policy name.

When a policy is not being added or updated, the policy name can be a generic name, for example POLICY V\* lists information for all policies that begin with V.

**VOLType**

optionally, specifies whether the policy applies to scratch or specific allocation.

**SCrash or Nonspec**

The policy applies only to scratch (nonspecific) allocations. This is the default.

**SPecific**

The policy applies only to specific allocations.

See individual parameters for restrictions on the VOLType that is permitted or not allowed.

---

---

**Note:** If you want to specify a policy that applies to both scratch and specific volumes, you must use two separate POLicy commands with the same POLicy NAME, one with VOLType (SCrash) (or not specified) and one with VOLType (SPecific).

---

---

**Named Policy Parameters**

When a new policy is added, the default VOLType is SCrash (or Nonspec). You can add the same policy name for both SCrash and SPecific VOLTypes.



The following parameters require a *name* or NAME (*name*) parameter:

### **NOne**

optionally, removes all policy parameters associated with the named policy. Specifying NOne for a policy implies that the default behavior is applied, for example, any media or rectech, subpool 0, etc.

### **MEDia(media-type)**

optionally, specifies the desired media type for a data set being created. You can enter a list of media types, separated by commas. A list specifies a generic pool from which a selection is made. There is no implied priority.

*media-type* indicates the media type. See [Appendix A, "MEDia, RECtech, and MODel Values"](#) for a list of valid *media-type* values.

If this parameter is not specified, a default is chosen based on the value of the RECtech parameter. See [Appendix A, "MEDia, RECtech, and MODel Values"](#) for default values used if the MEDia parameter is omitted.

MEDia and VOLType(SPECIFIC) are mutually exclusive unless the NOTAPEplex parameter is specified. If both media type and recording technique are omitted, all available device types are eligible for allocation.

### **RECtech(recording-technique)**

optionally, specifies the method used to record data tracks on the tape surface for the desired data set. You can enter a list of recording techniques, but they must be separated by commas. A list specifies a generic pool from which a selection is made. There is no implied priority.

*recording-technique* indicates the recording technique. See [Appendix A, "MEDia, RECtech, and MODel Values"](#) for a list of valid *recording-technique* values.

If this parameter is not specified, a default is chosen based on the value of the MEDia parameter. See [Appendix A, "MEDia, RECtech, and MODel Values"](#) for default values used if the MEDia parameter is omitted. If both media type and recording technique are omitted, all available device types are eligible for allocation.

RECtech and MODel are mutually exclusive.

### **MODel(model-type)**

optionally, specifies the model number of a transport. You can enter a list of models, separated by commas. A list specifies a generic pool from which a selection is made. There is no implied priority.

MODel provides the same type of information as RECtech, but a user may find it more convenient to specify a transport model rather than a recording technique. MODel and RECtech are mutually exclusive.

*model-type* indicates the model type. See [Appendix A, "MEDia, RECtech, and MODel Values"](#) for a list of valid *model-type* values.

### **SUBPool(subpool-name)**

optionally, specifies the subpool used to satisfy a scratch request. SUBPool and VOLType(SPECIFIC) are mutually exclusive.

*subpool-name* indicates the subpool name.

### **ESOTeric(esoteric-group-name or esoteric-list)**

optionally, specifies the name of a device esoteric group, or list of device groups defining the list of eligible transports to be used to satisfy a tape request.

*esoteric-group-name* or *esoteric-list* indicates the esoteric group name. The name specified can consist of one to eight alphanumeric or national (#, @, \$) characters. A slash (/) or hyphen (-) is also valid.

If the esoteric list 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. During drive exclusion, devices in any of the listed esoterics are included. During drive prioritization, devices are ordered according to their position in the esoteric list.

**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**

Drives are preferred based on LSM location of a specific volume. This is the default for specific volumes. **Location** is only valid for **VOLType (SPecific)**.

**Esoteric**

Drives are preferred based on the order specified in the esoteric list. This is the default for scratch allocations.

**Count**

Drives are preferred based on scratch counts in the LSM. **Count** is mutually exclusive with **VOLTYPE(SPecific)**.

**MGMTclas(name)**

optionally, specifies a Management Class defined on the VTCS **MGMTclas** control statement. See "["MGMTclas Control Statement"](#)" on page 3-222 for more information.

*name* indicates the Management Class name.

**TAPEPlex(name)**

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* indicates the TapePlex name. This must be a TapePlex name specified on a **TAPEPlex** command. 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.

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. **NOTAPEPlex** and **TAPEPlex** are mutually exclusive.

**IDAXESOTeric(esoteric-name)**

optionally, specifies the esoteric name to be substituted during IDAX (MVS Interpreter/Dynamic Allocation Exit) processing.

*esoteric-name* indicates a valid esoteric name to be used for the allocation.

Specification of IDAXESOTeric is similar to changing the JCL unit name, and completely replaces the original esoteric.

Unlike the ESOTeric parameter, the IDAXESOTeric *esoteric-name* does not need to include any devices in the original JCL esoteric.

#### **IDAXEXPDT(yyyyddd)**

optionally, specifies the expiration date to be applied to the allocation during IDAX.

*yyyyddd* indicates the desired expiration date (4-digit year and Julian day).

---



---

**Note:** IDAXEXPDT and IDAXRETPD are mutually exclusive. IDAXEXPDT and VOLType(Specific) are mutually exclusive.

---



---

#### **IDAXRETPD(nnnn)**

optionally, specifies the retention period to be applied to the allocation during IDAX.

*nnnn* indicates the retention period in number of days.

---



---

**Note:** IDAXRETPD and IDAXEXPDT are mutually exclusive. IDAXRETPD and VOLType(Specific) are mutually exclusive.

---



---

#### **IDAXVOLcnt(volume-count)**

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* indicates the maximum volume count. Valid values are 1-255.

#### **IDAXSUBsys(subsystem-name)**

optionally, specifies that SMC direct an allocation job to another subsystem or program.

For example, SMC can direct any FDR or DFDSS job to Oracle's 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.

*subsystem-name* indicates the subsystem name.

---



---

**Note:** IDAXSUBsys and VOLType(Specific) are mutually exclusive.

---



---

#### **IDAXPROGram(program-name)**

optionally, alters the program name in the allocation job.

For example, this parameter can alter the program name for DFDSS (ADDRSSU) to the ExHPDM User Interface Module (SOVDSSU), and in conjunction with the IDAXSUBsys parameter, direct the backup DFDSS job to ExHPDM.

*program-name* indicates the program name.

---



---

**Note:** Refer to the *ExHPDM Operator and System Programmer's Guide* for more information about the ExHPDM backup and restore software.

---



---

**EXTVOLesot(esoteric-list)**

optionally, specifies the name of a device esoteric group, or list of device groups defining the list of eligible transports to be used to satisfy a tape request when the specific volume is found to be outside of the library. `EXTVOLesot` can only be specified if `VOLTYPE=SPECIFIC` is also specified.

*esoteric-list* indicates the esoteric-list. Each esoteric group name in the list can consist of one to eight alphanumeric or national (#, @, \$) characters. A slash (/) or hyphen (-) is also valid.

**SPLITvolref**

optionally, specifies whether `VOL=REF` chains are to be broken. `SPLITvolref` and `VOLType(SPECIFIC)` are mutually exclusive.

**OFF**

`VOL=REF` chains are not broken. This is the default.

**ON**

`VOL=REF` chains are broken and the `LABEL` data set sequence number is set to 1.

**ALLOCFail**

optionally, specifies that a job step matching the policy should be failed during SMC allocation. When this parameter is specified, any job step that matches the policy has all its devices marked ineligible by the SMC allocation process, which causes the job to fail. `ALLOCFail` is mutually exclusive with all other parameters except for `NAME`, `LIST`, `VALIDATE`, and `VOLType`.

## READ

**Interfaces:**

- Console, utility, `SMCCMDS` data set, or `SMCPARMS` data set
- UUI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required, or may be input to the `SMCUSIM` utility

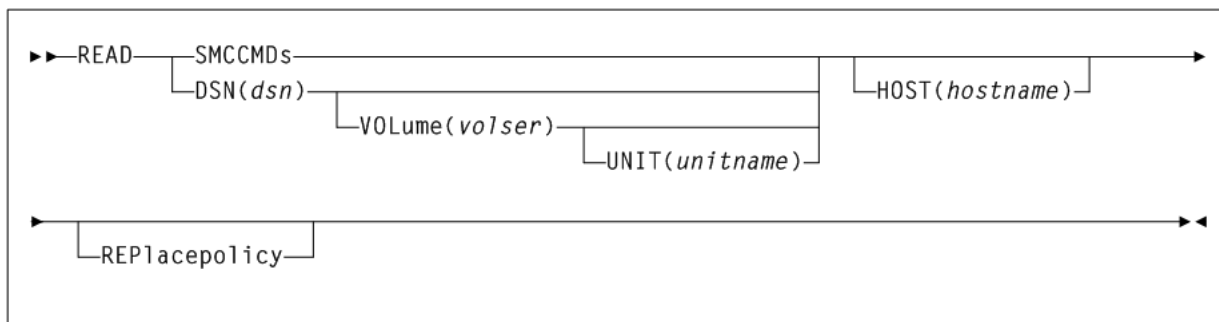
## Description

The SMC `READ` command enables you to enter a series of commands using an input data set instead of console commands.

## Syntax

Figure 2–29 shows `READ` command syntax.

Figure 2–29 READ command syntax



## Parameters

As shown in [Figure 2–29](#), the READ command includes the following parameters:

### SMCCMDS

optionally, re-processes commands contained in the data set specified in the SMCCMDS DD statement of the SMC START procedure.

### DSN(dsn)

optionally, processes commands contained in the specified data set.

*dsn* indicates the data set name.

---

**Note:** If the DSN parameter specifies a member name, the full data set must be enclosed in single quotes.

---

### VOLume(volser)

optionally, specifies the volume serial number of the DASD volume on which the data set resides. Specify this parameter if the data set is not catalogued.

*volser* indicates the volume serial number on which the data set resides.

### UNIT(unitname)

optionally, specifies the unit name of the specified VOLume for uncataloged data sets. Requires that VOLume also be specified.

*unitname* indicates the unit name. If not specified, then a unit name of SYSALLDA is the default.

### HOST(hostname)

optionally, specifies a host name for which this READ command is valid. If this parameter is specified, the READ command is only processed when the specified *hostname* matches the MVS host. If this parameter is not specified, the READ command is processed regardless of the MVS host.

*hostname* indicates the host name.

HOST is not valid when the READ command is entered from an MVS console.

Specification of the HOST parameter enables you to share a single SMC parameter file set across multiple MVS hosts as shown in the following example:

```
your.parmlib(COMMCMDs)
```

```
MSGDEF CASE(MIXED)
...other common SMC parameter settings...
```

```
READ DSN('your.parmlib(MVS1CMDS)') HOST(MVS1)
READ DSN('your.parmlib(MVS2CMDS)') HOST(MVS2)
```

```
your.parmlib(MVS1CMDS)
```

```
DRIVEMAP CL(AB0-AB7) SE(BB0-BB7)
```

```
your.parmlib(MVS2CMDS)
```

```
DRIVEMAP CL(CB0-CB7) SE(BB0-BB7)
```

The file `COMMCMS` contains SMC parameter settings applicable to all hosts, and the `READ` commands with `HOST` specifications for your MVS hosts. The file `MVS1CMDS` contains SMC specifications applicable only to MVS host `MVS1`, while the file `MVS2CMDS` contains SMC specifications applicable only to MVS host `MVS2`.

### **REPlacepolicy**

optionally, specifies that all SMC `POLICY` objects are to be replaced by any `POLICY` statements found in the specified data set (or in any data set subsequently read by the specified `READ` data set).

If `REPlacepolicy` is not specified, individual `POLICY` statements within a `READ` command update or add only the referenced `POLICY` statements and do not affect unreferenced `POLICY` statements.

---

---

#### **Note:**

- After the `READ REPlacepolicy` command is executed, any `TAPEREQ POLICY` references are re-validated. If any `TAPEREQ POLICY` references are subsequently found to be in error, the invalid `TAPEREQ` statements are left active, but reference a "null" `POLICY` (that is, the equivalent of specifying `POLICY NAME(name) NONE`).
  - It is recommended that any `TAPEREQ` statement(s) be reloaded by inclusion of the `TREQDEF` command whenever a `READ REPlacepolicy` command is executed, in order to re-validate all SMC tape policies at the same time.
- 
- 

## **RESYNChronize**

### **Interfaces:**

- Console, utility, `SMCCMS` data set, or `SMCPARMS` data set
- UUI Support: Yes (No XML/CSV output)

### **Subsystem Requirements:**

Active SMC required, or may be input to the `SMCUSIM` utility

## **Description**

The SMC `RESYNChronize` command re-establishes connections to all defined TapePlexes to acquire drive configuration information from all TapePlexes.

This action is automatically performed when the SMC first activates a new path to a TapePlex, when an HSC subsystem (local or remote) reports a configuration change, or when an active local path to a TapePlex is accessed during an MVS allocation or mount event. However, the operator must issue the `RESYNChronize` command in the following instances:

- When the MVS/CSC RESYNC command is issued and the MVS/CSC is active.
- When the user wants to automate pending mounts for a newly activated TapePlex, and does not want to wait for an MVS allocation or mount event to trigger the reconnection.
- To attempt to redrive outstanding mounts that may have been lost due to hardware or communication errors.

---

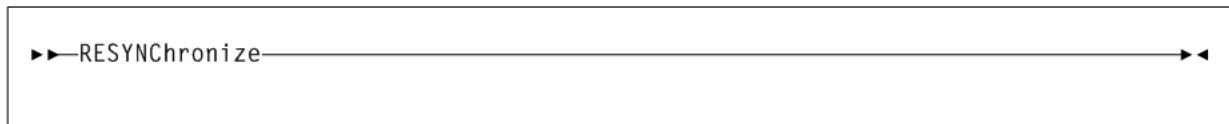
**Note:** It is recommended that you include a RESYNChronize command at the end of your SMCCMDS file (after all TAPEPlex and SERVer commands) to initialize communication with TapePlexes prior to the first intercepted allocation or mount.

---

## Syntax

Figure 2–30 shows RESYNChronize command syntax.

**Figure 2–30** RESYNChronize command syntax



## Parameters

None.

## Route

### Interfaces:

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (No XML/CSV output)

### Subsystem Requirements:

Active SMC required

## Description

The SMC `Route` command routes command 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.

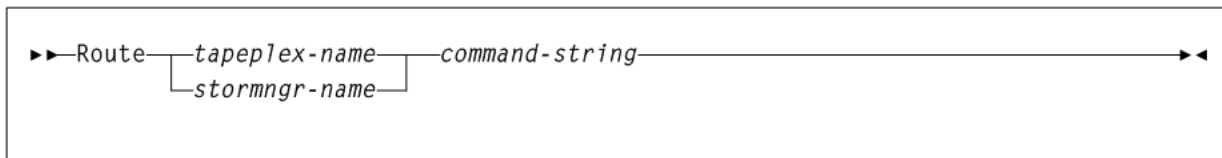
**Note:**

- The HSC MOUNT, DISMOUNT, ENTER and MOVE commands can now be issued from SMC via the ROUTE command.
- Currently, the only command that will be accepted and processed by a VLE is the QUERY SERVER command.
- You can also use the Route command to issue various commands from a VM Client to an ACSLS XAPI server. Refer to Oracle's ELS publication *XAPI Client Interface to ACSLS Server Reference* for more information.

**Syntax**

Figure 2–31 shows Route command syntax.

**Figure 2–31 Route command syntax**

**Parameters**

As shown in Figure 2–31, the Route command includes the following 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. VTCS commands should not be prefixed with VT; the HSC UI interface routes VTCS commands to the correct functional processor without the VT prefix. The command string must be a command supported by the HSC UI or any VTCS command (except DISPLAY MSG and DISPLAY CMD). This is a required value.

**SERVer****Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**



---

Active SMC required, or may be input to the SMCUSIM utility

## Description

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. See "TAPEPLEX" on page 2-79 for more information.
- Similarly, before a `SERVer` is defined, the VLE that it references must be defined using a `STORMNGR` command. The VLE name associated with a `SERVer` cannot be changed. See "STORMNGR" on page 2-77 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)
```

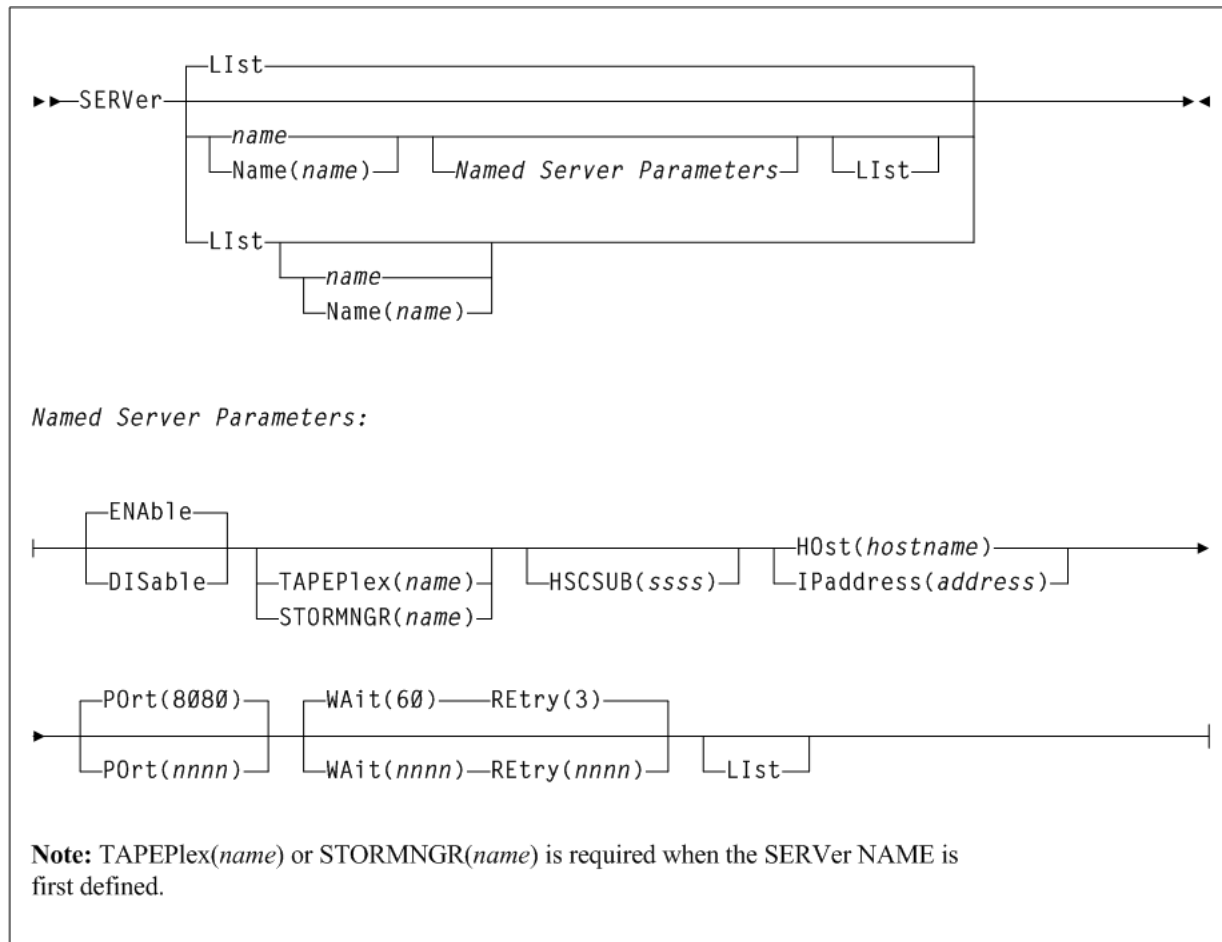
---

---

## Syntax

Figure 2–32 shows `SERVer` command syntax.

Figure 2–32 SERVer command syntax



## Parameters

As shown in Figure 2–32, the SERVer command includes the following parameters:

### name or NAmE(name)

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

*name* may be specified with or without the keyword NAmE. If the keyword NAmE is not specified, the name, if specified, must be the first parameter.

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 SMC HTTP server component, there can be multiple NAMED server paths for a single TapePlex configuration. Refer to Oracle's ELS publication *Configuring and Managing SMC* for more information.

---

**List**

optionally, displays status information for TapePlex server paths.

If *name* is also specified, only the named TapePlex server path is listed. If *name* is not specified, then all TapePlex server paths are listed.

**Note:**

- `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.

**Named SERVer Parameters**

The following parameters require a name or `NAme` (*name*) parameter:

**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(name)**

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 first defined.

*name* indicates the TapePlex name. This name is reported in any TapePlex error message. 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.

**STORMNGR(name)**

optionally, specifies a VLE name as defined on an SMC `STORMNGR` command. The `STORMNGR` parameter must be specified when a new server is first defined. `TAPEPlex` and `STORMNGR` are mutually exclusive.

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

**HSCSUB(ssss)**

optionally, specifies the name of the HSC subsystem that represents the Tapeplex associated with the server. This parameter is required only when there is more than one HSC subsystem executing on the server host (HSC running in `MULT` mode).

*ssss* indicates the HSC subsystem name.

**H0st(hostname)**

optionally, specifies the IP resolver host name on which the remote HSC subsystem resides. `H0st` and `IPaddress` are mutually exclusive.

*hostname* indicates 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 or hyphen.

**IPaddress(address)**

optionally, specifies the subsystem IP address for the remote HSC subsystem. `IPaddress` and `H0st` are mutually exclusive.

*address* indicates the IP address. 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.

**POrt(nnnn)**

optionally, specifies the server port. *nnnn* indicates the server port. The default is 8080.

**WAIt(nnnn)**

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* indicates the wait time in seconds. The default is 60.

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(nnnn)**

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

*nnnn* indicates the number of retries. The default is 3.

## SIMulate

**Interfaces:**

- Console, utility, `SMCCMDS` data set, or `SMCPARMS` data set
- UUI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required, or may be input to the `SMCUSIM` utility

## Description

The SMC `SIMulate` command simulates how SMC would process a single DD job step with a given set of jobstep characteristics. The `SIMulate` command simulates the allocation influence (SSI24) portion of SMC processing, but not mount processing.

You provide the keyword jobstep parameters to simulate, and the command invokes SMC allocation, policy lookup, and volume lookup. The `SIMulate` output displays the simulation results.

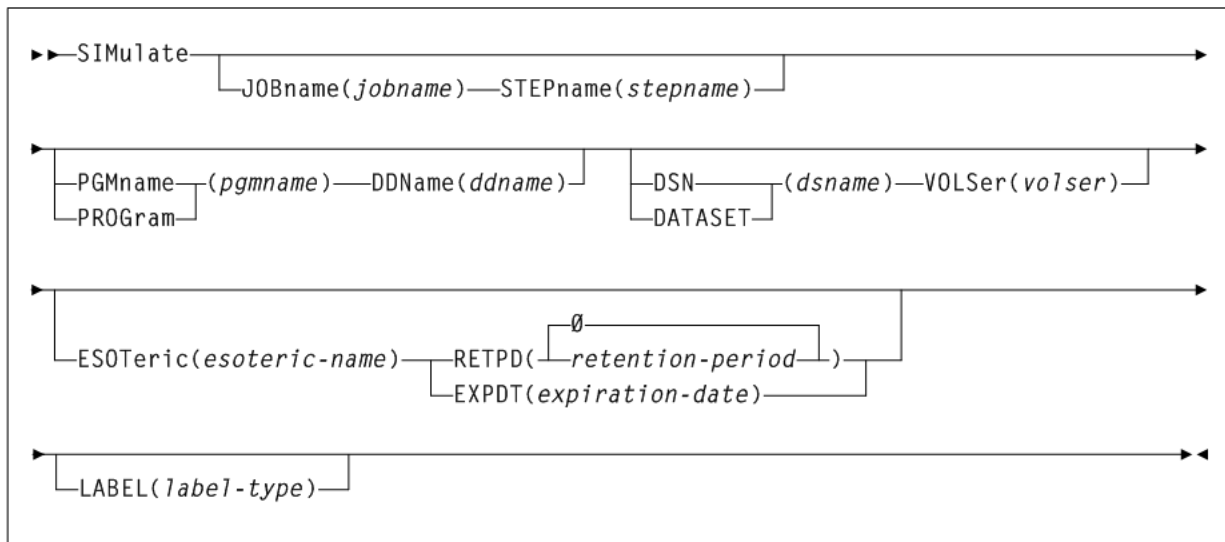
While `SIMulate` does not simulate certain jobstep characteristics (such as multiple units, affinity separation, or volref attributes), it does provide a powerful tool to test how changes in SMC policies, including `TAPEREQ` and `POLICY` commands, user exits, and `TAPEREQ` and `SERVER` changes will affect your production jobs.

You can issue `SIMulate` command to an active SMC subsystem or via the `SMCUSIM` utility. The `SMCUSIM` utility, working with the `SIMulate` command, produces XML output that can be manipulated as CSV to produce before and after reports to compare how jobs would be processed. See "[SMC UUI Simulation Utility \(SMCUSIM\)](#)" on page 4-27 for more information about the `SMCUSIM` utility.

## Syntax

Figure 2-33 shows `SIMulate` command syntax.

**Figure 2-33** *SIMulate command syntax*



## Parameters

As shown in Figure 2-33, the `SIMulate` command includes the following parameters:

### **JOBname(jobname)**

optionally, specifies the job name of the simulated job step to test. The format must conform to MVS standards. If not specified, "(any)" is used, which prevents a match to non-generic job names specified in SMS ACS routines or `TAPEREQ` control statements.

*jobname* indicates the job name.

**STEPname(stepname)**

optionally, specifies the step name of the simulated job step to test. The format must conform to MVS standards. If not specified, "(any)" is used, which prevents a match to non-generic step names specified in SMS ACS routines or TAPEREQ control statements.

*stepname* indicates the step name.

**PGMname(pgmname) or PROGRAM (pgmname)**

optionally, specifies the program name of the simulated job step to test. The format must conform to MVS standards. If not specified, "(any)" is used, which prevents a match to non-generic program names specified in TAPEREQ control statements.

*pgmname* indicates the program name.

**DDName(ddname)**

optionally, specifies the DD name of the simulated data set to test. The format must conform to MVS standards. If not specified, "(any)" is used, which prevents a match to non-generic DD names specified in TAPEREQ control statements.

*ddname* indicates the DD name.

**DSN(dsname) or DATASET(dsname)**

optionally, specifies the data set name of the simulated data set to test. The format must conform to non-PDS member MVS standards. If not specified, "(any)" is used, which prevents a match to non-generic program names specified in TAPEREQ control statements.

*dsname* indicates the data set name.

**VOLSer(volser)**

optionally, specifies the volume serial number of the simulated data set to test. The format must conform to MVS standards. If not specified, "SCRATCH" is used, which simulates a scratch mount.

*volser* indicates the volume serial number.

**ESOTeric(esoteric-name)**

optionally, specifies the name of a device esoteric group defining the list of eligible transports used to satisfy the tape request. The specified esoteric must be a defined MVS esoteric. If not specified, the SIMulate command simulates an original esoteric consisting of all known library, non-library, and virtual devices for all TapePlex(es). If the default esoteric is used, it may not correspond to a defined MVS esoteric.

*esoteric-name* indicates the esoteric name.

**RETPD(retention-period)**

optionally, specifies the retention period of the simulated data set to test. The format must conform to MVS standards. If not specified, 0 is used.

*retention-period* indicates the retention period.

**EXPDT(expiration-period)**

optionally, specifies the expiration date of the simulated data set to test. The format must conform to MVS standards. If not specified, the current date is used.

*expiration-period* indicates the expiration date.

**LABEL(label-type)**

optionally, specifies the label type to be used to select the policy.

*label-type* indicates the label type. Specify one of the following:

- SL indicates standard-labeled tape. This is the default if the parameter is omitted.
- AL indicates ANSI-labeled tape.
- NL indicates nonlabeled tape.
- NSL indicates nonstandard labeled tape.

## Examples

SMC SIMulate command output displays the simulated job characteristics and policy sections displaying the SMS, TAPEREQ and SMC policy applied.

The following example shows the "minimum" output, where no SMC tape policies were applied:

### **Example 2-1 SIMulate Command Output - No Tape Policies Applied**

```
SMC0222 SIMULATE command results:
---- Jobstep characteristics ----
JOBNAME=(any), STEPNAME=(any), PROGRAM=(any), DDNAME=(any)
DSN=SMCINT.BT1.TEST, RETPD=0
VOLSER=SCRATCH, ESOTERIC=(any)
---- Volume lookup results ----
Owning TAPEPLEX=HSCQ
MEDIA=STANDARD, ECART, STK2P
RECTECH=LONGITUD, STK2P
---- List of eligible drives ----
0AA0 TAPEPLEX=HSCQ, ACS=00, LSM=00, 18TRACK, pref=1
0AA1 TAPEPLEX=HSCQ, ACS=00, LSM=00, 18TRACK, pref=1
remainder of drives
```

The following figure shows output when tape policies are applied at IDAX time from a matched SMC policy.

### **Example 2-2 SIMulate output - Tape Policies Applied at IDAX**

```
SMC0222 SIMULATE command results:
---- Jobstep characteristics ----
JOBNAME=(any), STEPNAME=(any), PROGRAM=(any), DDNAME=(any)
DSN=SMCINT.MGMTCLAS.TEST, RETPD=0
VOLSER=SCRATCH, ESOTERIC=(any)
---- SMC SMS policy results ----
*SMC POLICY applied at IDAX
ESOTERIC=CART from POLICY IDAXESOTERIC
*RETPD updated at IDAX
*VOLCNT updated at IDAX
MGMTCLAS=TEST matches an SMC POLICY name
---- POLICY lookup results ----
POLICY=TEST
ESOTERIC=(SVTSS1, SVTSS2)
MEDIA=VIRTUAL
RECTECH=VIRTUAL
IDAXESOTERIC=CART
IDAXRETPD=120
IDAXVOLCNT=25
---- Volume lookup results ----
Owning TAPEPLEX=HSCQ
*No scratch volume for request
MEDIA=STANDARD, ECART, STK2P, VIRTUAL
RECTECH=LONGITUD, STK2P, VIRTUAL
```

```

---- List of eligible drives ----
9000 TAPEPLEX=HSCQ, VTSS=SVTSS1, VIRTUAL, pref=1
9001 TAPEPLEX=HSCQ, VTSS=SVTSS1, VIRTUAL, pref=1
remainder of drives

```

## SMSDef

### Interfaces:

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UI Support: Yes (No XML/CSV output)

### Subsystem Requirements:

Active SMC required, or may be input to the SMCUSIM utility

## Description

The SMC `SMSDef` command defines default SMC DFSMS support. This command takes effect when SMC DFSMS support is activated by setting the `ALLOCDef SMS` parameter value to `ON`.

During an allocation resulting in a tape mount, the SMC makes up to four calls to the DFSMS ACS routines. The `SMSDef` command allows you to bypass certain routine calls.

- The `MGMPol` parameter specifies whether the management class returned from a DFSMS ACS routine may represent an SMC policy. `MGMPol` allows the user's SMC DFSMS routines to point to an SMC `POLICY` with a scratch allocation, and apply all policy values, including media, subpool, and VTCS management class esoteric.

Using `MGMPol (ALL)` or `MGMPol (IDAX)` allows `POLICY` parameter names that start with `IDAX*` (including `IDAXESOT`) to be applied during JCL interpretation, allowing override of JCL values by SMC DFSMS routines.

Oracle recommends that you specify `SMSDef MGMPol (ALL)` to direct SMC to process all DFSMS returned management class names as policy names.

- If `MGMPol` is `OFF`, or if `MGMPol` is `ON` but no matching SMC policy name is found, the `MGMTCLAS` name returned by the DFSMS routine is passed as the VTCS `MGMTCLAS` name during volume lookup and volume mount processing.
- If VTCS is installed in your environment, Oracle recommends ensuring that all `MGMTCLAS` names returned by your DFSMS routines match a defined SMC `POLICY` name. If that is the case, specify `VTVmgmt (OFF)` to avoid inadvertently passing an invalid `MGMTCLAS` name to VTCS.

---

**Note:** `VTVmgmt (ON)` lets you use a `MGMTCLAS` from a DFSMS routine directly as a VTCS `MGMTCLAS` if `MGMPol` is `OFF`, or if `MGMPol` is `ON` but no matching SMC policy name is found. This setting applies to the old DFSMS interface format.

---

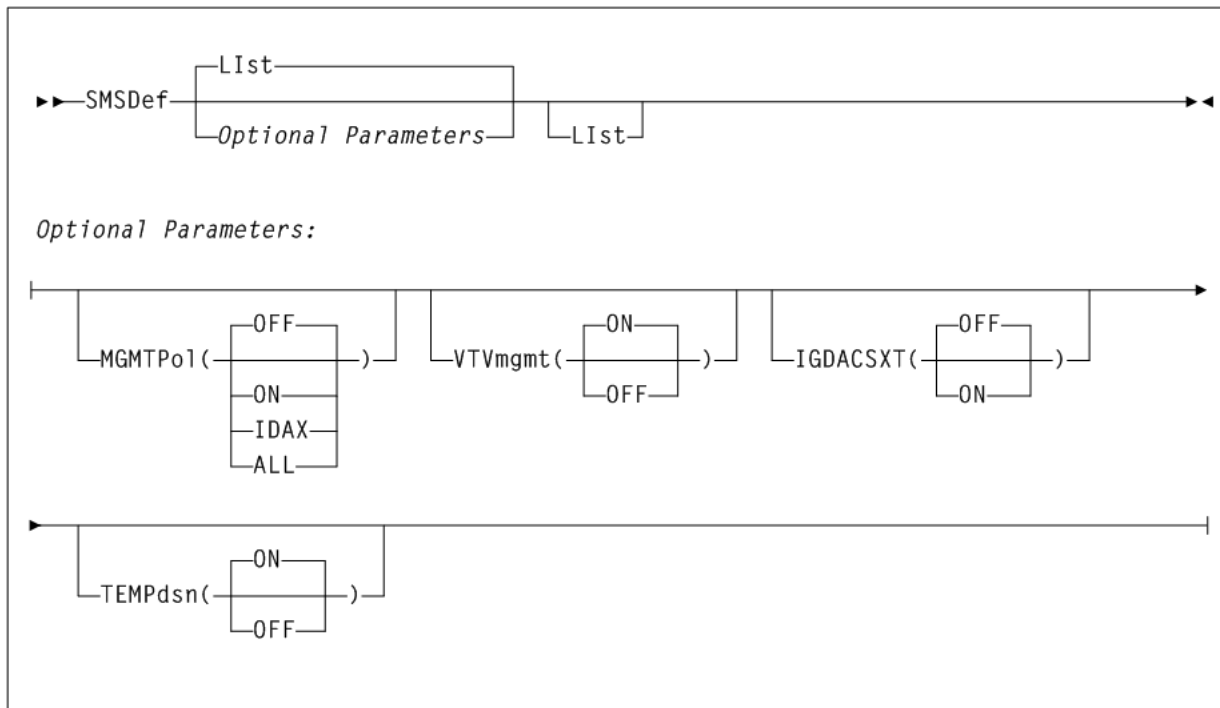
- If your environment does not support DFSMS functions, set the `ALLOCDef SMS` parameter to `OFF` (the default). The SMC bypasses all DFSMS ACS routine calls.

## Syntax

Figure 2–16 shows `SMSDef` command syntax.



Figure 2-34 SMSDef command syntax



## Parameters

As shown in [Figure 2-16](#), the SMSDef command includes the following parameters:

---

**Note:** This command includes additional parameters that apply to the previous DFSMS interface format, which has been superseded by the SMC POLICY command. Refer to the *ELS Legacy Interfaces Reference* for information about these legacy parameters.

---

### List

optionally, lists the current DFSMS settings.

- List is the default when no parameters are specified on the SMSDef command.
- List may be specified with other parameters. In this case, the List is generated after the other parameters are processed.

### Optional Parameters

#### MGMTPol

optionally, specifies whether the management class returned from a DFSMS ACS routine may represent an SMC policy.

#### OFF

A management class name returned from a DFSMS ACS routine is either ignored (if VTVmgmt if OFF), or is interpreted as a VTCS management class name (if VTVmgmt is ON). This is the default value.

**ON**

A management class name returned from an SMC DFSMS ACS routine is interpreted as an SMC policy name during allocation and mount message processing, but not during IDAX processing. If a matching SMC policy name is found, all policy information applies to the request. If the policy name is not found, other SMC DFSMS processing is applied as specified.

**IDAX**

A management class name returned from an SMC DFSMS ACS routine is interpreted as an SMC policy name during IDAX processing, but not during allocation or mount processing. If the policy name is not found, other SMC DFSMS processing is applied as specified. For example, if `ESOT(ON)` is specified, the returned storage group is used as a substitution esoteric.

**ALL**

A management class name returned from an SMC DFSMS ACS routine is always interpreted as a policy name. This is the recommended setting.

- If a matching SMC policy name is found, all policy information applies to the request.
- If the policy name is not found, other SMC DFSMS processing is applied as specified. For example, if `VTVMGMT` is `ON`, the DFSMS management class name is interpreted as a VTV management class.

---

---

**Note:** If `MGMTPol` is `ON`, and the matching policy name contains an esoteric, the esoteric name in the policy is used during common allocation to exclude devices not contained in the specified esoteric. It is not used as a “true” esoteric substitution parameter.

---

---

**VTVmgmt**

optionally, specifies whether the SMC invokes the storage class and management class DFSMS ACS routines during virtual volume allocation and virtual volume mount message processing.

**ON**

If `MGMTPol` is `OFF`, or if `MGMTPol` is `ON` but no matching SMC policy name is found, the management class name is provided to VTCS during volume lookup and volume mount processing.

**OFF**

If `MGMTPol` is `ON`, the management class name returned by the DFSMS ACS routine is used to look up policy values from the named SMC policy. If no matching SMC policy name is found, no policy is applied to the request.

If `MGMTPol` is `OFF`, SMC does not invoke the DFSMS ACS storage class or management class routines during allocation or mount processing.

**IGDACSXT**

optionally, specifies whether the SMC invokes the Pre-ACS Installation Exit, `IGDACSXT`, before invoking the DFSMS ACS routines during SMC DFSMS processing. Refer to the IBM publication *DFSMS Installation Exits* for more information about this exit.

**ON**

The `IGDACSXT` exit is invoked before the SMC calls the DFSMS ACS routines. The exit receives the `ACERO` control block that contains the read only variable settings.

The exit has the opportunity to change these settings before SMC invokes the DFSMS ACS routines.

**OFF**

The IGDACSXT exit is not invoked before the SMC calls the DFSMS ACS routines.

**TEMPdsn**

optionally, specifies whether temporary data sets are presented to the DFSMS ACS routines as permanent data sets so that a MGMTCLAS may be returned.

**ON**

The &DSTYPE read only variable is set to 2 (indicating a permanent data set) if the original data set is a temporary data set.

The &LIBNAME read only variable will contain the character 3 if the &DSTYPE read only variable is changed.

ON is the default setting.

**OFF**

The &DSTYPE read only variable is set to the actual disposition of the data set.

## STORMNGR

**Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required, or may be input to the SMCUSIM utility

### Description

The SMC STORMNGR command defines a VLE to SMC.

---



---

**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.

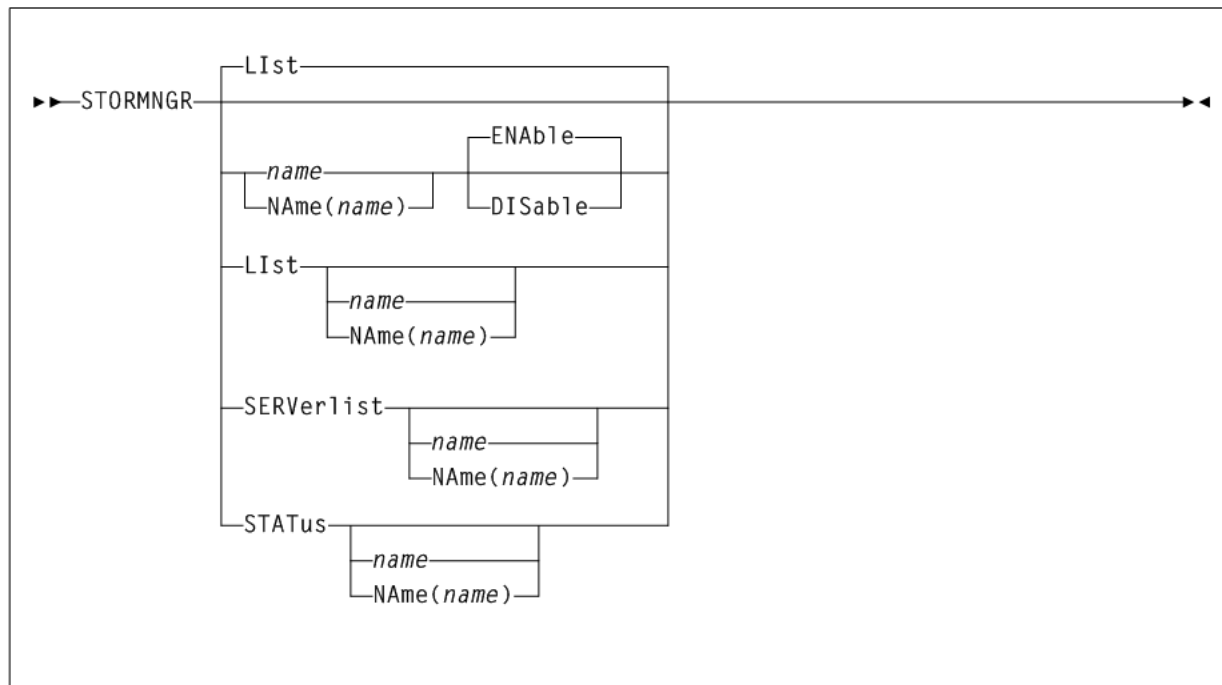
---



---

### Syntax

Figure 2–35 shows STORMNGR command syntax.

**Figure 2–35** STORMNGR command syntax

## Parameters

As shown in [Figure 2–35](#), the STORMNGR command includes the following parameters:

### **name or NAmE(name)**

Specifies VLE to be defined, modified or listed. This name is also defined to VTCS, and is initially defined by the VLE GUI. If *name* is not specified, then all VLEs are listed.

*name* may be specified with or without the keyword NAmE. If the keyword NAmE is not specified, the *name*, if specified, must be the first parameter.

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.

### **List**

optionally, lists VLE information.

If *name* is also specified, only the named VLE is listed. If *name* is not specified, then all VLEs are listed.

**SERVerlist**

optionally, lists defined VLEs, their attributes and associated servers.

If *name* is also specified, only the named VLE is listed. If *name* is not specified, then all VLEs are listed.

**STATus**

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

If *name* is also specified, only the named VLE is listed. If *name* is not specified, then all VLEs are listed.

## TAPEPlex

**Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required, or may be input to the SMCUSIM utility

## Description

The SMC `TAPEPlex` command defines a TapePlex; a specific Oracle StorageTek tape hardware configuration normally represented by a single CDS. The `TAPEPlex` command can also list TapePlexes that the SMC tries to communicate with, and report their status.

---



---

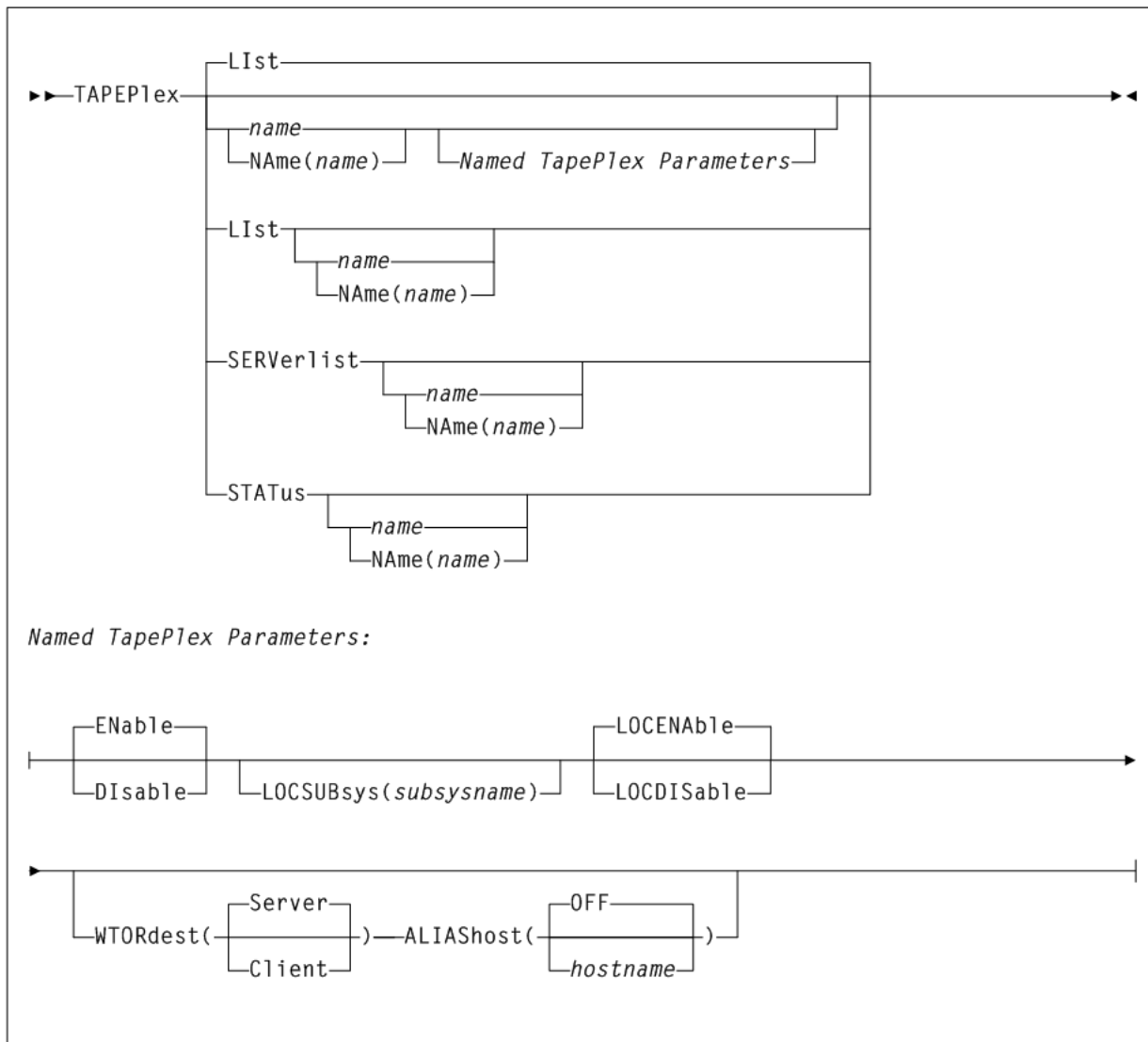
**Note:**

- `TAPEPlex` and `SERVer` commands are required to access remote HSC TapePlexes.
  - At least one TapePlex must be defined to allow SMC to initialize.
- 
- 

## Syntax

Figure 2–36 shows `TAPEPlex` command syntax.

Figure 2–36 TAPEPlex command syntax



## Parameters

As shown in [Figure 2–36](#), the TAPEPlex command includes the following parameters:

### **name or NAmE (name)**

specifies the TapePlex name to be added, modified, or listed.

*name* may be specified with or without the keyword **NAmE**. If the keyword **NAmE** is not specified, the name, if specified, must be the first parameter.

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.

**List**

optionally, lists TapePlex information.

If *name* is also specified, only the named TapePlex is listed. If *name* is not specified, then all TapePlexes are listed.

`LIst` is the default when no parameters are specified on the `TAPEPlex` command. In this case, all TapePlexes are listed. `LIst` may be specified with other parameters. In this case, the named TapePlex is listed after the other parameters are processed.

**SERVERlist**

optionally, lists defined TapePlexes, their attributes and associated servers.

If *name* is also specified, then only the named TapePlex servers are listed. If *name* is not specified, then all servers are listed.

**STATus**

optionally, lists current status of all TapePlexes, or a single named TapePlex. The TapePlex status indicates whether a TapePlex is active, inactive, or disabled. For an active TapePlex, the status lists the name of the current server or local subsystem.

If *name* is also specified, then only the named TapePlex status is listed. If *name* is not specified, then the status of all TapePlexes is listed.

**Named TAPEPlex Parameters**

The following parameters require a *name* or `NAmE (name)` parameter:

**ENABle**

optionally, enables the specified TapePlex to be selected for allocation or mount requests.

**DISable**

optionally, disables the specified TapePlex. The TapePlex is not used for any allocation or mount requests.

**LOCSUBSsys(subsysname)**

optionally, specifies a local HSC and/or MVS/CSC subsystem.

*subsysname* indicates the subsystem name.

**LOCENABle**

optionally, enables the specified HSC or MVS/CSC subsystem to be used as a local path to the TapePlex.

**LOCDISable**

optionally, disables the specified HSC or MVS/CSC subsystem. `LOCDISable` can be used to force the SMC to ignore the local subsystem and access the TapePlex via a remote server.

**WTORdest**

optionally, specifies the console to which HSC WTOR messages are directed. Refer to Oracle's ELS publication *Configuring and Managing SMC* for messages that apply.

**Client**

The remote HSC returns 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. This is the default.

**ALIAShost**

optionally, specifies a host alias to be used in communications with the HSC TapePlex.

**hostname**

a 1-8 character host name.

**OFF**

The native SMF host name is used in communications with the HSC library. This is the default.

## TCPip

**Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UII Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required, or may be input to the SMCUSIM utility

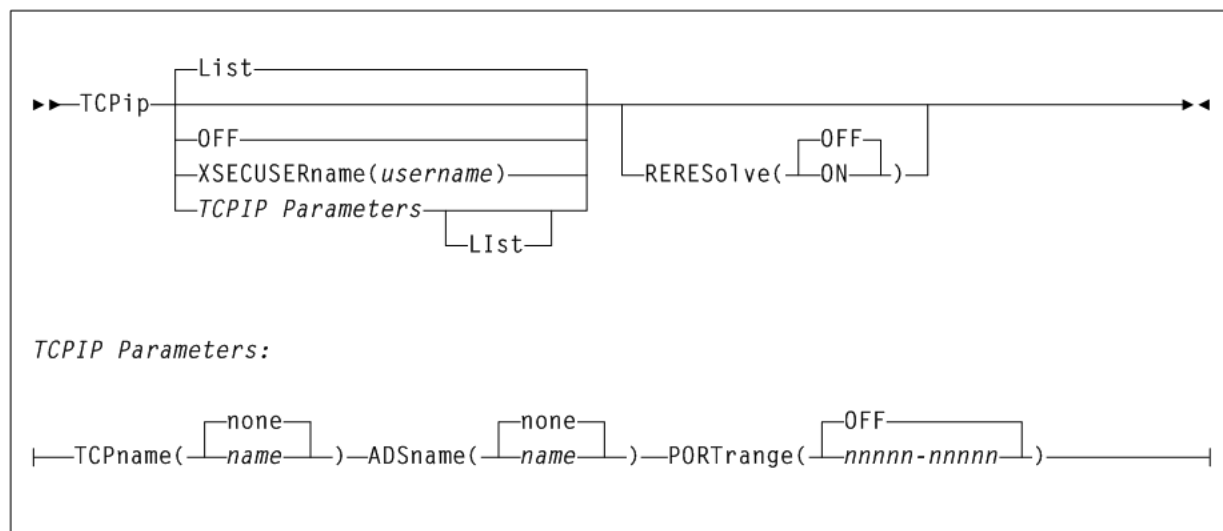
## Description

The SMC TCPip command enables you to alter and/or lists current TCPNAME and ADSNAME defaults for your TCP/IP communications environment. You can use this command to direct TCP/IP requests to a specific TCP/IP stack on an MVS host.

## Syntax

Figure 2–37 shows TCPip command syntax.

**Figure 2–37 TCPip command syntax**



## Parameters

As shown in Figure 2–37, the TCPip command includes the following parameters:



**List**

optionally, lists current TCPNAME and ADSNAME settings.

- LIst is the default when no parameters are specified on the TCPip command.
- LIst may be specified with other parameters. In this case, the LIst is generated after the other parameters are processed.

**OFF**

optionally, specifies that system defaults are used for client TCP/IP communication.

**RERESolve**

optionally, specifies whether SMC attempts to re-resolve the SERVER HOST name after a communication failure.

**OFF**

SMC does not attempt to re-resolve the server host name. This is the default.

**ON**

SMC attempts to re-resolve the server host name.

SMC first attempts a retry as specified by the SERVER RETRY parameter. If the error indicates that the IP address may be unavailable, and the RERESolve parameter is ON, SMC then attempts to resolve the HOST name again, and if a new IP address is generated, retries the transaction.

**XSECusername(username)**

optionally, specifies the XAPI user name associated with all transactions originating from this client. This user name will only be used when a login challenge is issued by the targeted server.

*username* indicates the XAPI user name.

---



---

**Note:** The same USER must be defined on both the client and server using the XUDB command. See "XUDB" on page 2-106 for more information.

---



---

**TCP/IP Parameters**

These parameters are mutually exclusive with OFF or RERESolve.

**TCPname(name)**

optionally, specifies the TCPNAME value for a specific TCP/IP stack on an MVS host. TCPNAME is the name of the MVS TCP/IP stack. Refer to the IBM publication *TCP/IP for MVS Application Programming Interface Reference* for more information about TCPNAME. The default is none.

*name* indicates the TCPNAME value. The default is none.

- If a subsystem name is specified, it must consist of one to four alphanumeric or national (#,@,\$) characters, the first character being alphabetic or national.
- If an address name is specified, it must consist of one to eight alphanumeric or national characters.

**ADSname(name)**

optionally, specifies the ADSNAME value for a specific TCP/IP stack on an MVS host. ADSNAME is the jobname of the TCP/IP startup job. Refer to the IBM publication *TCP/IP*

for *MVS Application Programming Interface Reference* for more information about ADSNAME.

*name* indicates the ADSNAME value.

### **PORTrange**

optionally, specifies a range of ports to be used by the SMC to bind() sockets on the client when communicating on remote server paths.

When PORTrange is defined, the SMC binds() client sockets to one of the ports within the specified PORTrange and will not use client ports outside the PORTrange. This allows the SMC to operate behind a firewall that restricts communication to known ports. A unique port is required for each concurrent subtask requiring communication services for a volume lookup, mount, etc. If a PORTrange is not defined, then any ephemeral port is used by the SMC.

Only one PORTrange can be active at a time, but you can dynamically re-define the PORTrange even if the new PORTrange overlaps with the old PORTrange.

#### **nnnnn-nnnnn**

the port number range. Each port number can have a value of 1-65535. However:

- The minimum port number range that can be specified is 10 (for example, 6400-6410).
- The maximum port number range that can be specified is 1000 (for example, 6400-6500).

#### **OFF**

PORTrange logic is disabled. As a result, any ephemeral port is used.

---

---

#### **Note:**

- It is recommended that you specify a PORTrange that does not conflict with TCP/IP well-known ports.
  - It is recommended that you specify a PORTrange greater than the anticipated number of concurrent subtasks requesting remote communication services. For most installations, a PORTrange of 40 ports is sufficient. However, if SMC0128 messages are produced with a return code indicating no free port then a larger PORTrange is required.
  - When an SMC mount request to the HSC server results in an outstanding WTOR on the HSC, the used port number is unavailable until a reply is supplied for the outstanding WTOR. WTOR messages for mount requests are produced on the HSC when TAPEplex WTORdest(SERVER) is specified, or when TAPEplex WTORdest(CLIENT) is specified and the HSC mount message is not one returned to the SMC for processing. Refer to the publication *Configuring and Managing SMC* for messages that apply.
  - The TCPip LIST command may be used to display the high-water port number, indicating the largest number of concurrent communication subtasks executing at one time.
- 
- 

## **TRace**

### **Interfaces:**

- Console, utility, *SMCCMDS* data set, or *SMCPARMS* data set
- UUI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

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

## Description

The SMC *TRace* command enables SMC tracing using the GTF trace facility and optionally provides a snap dump of the address space before and after control block modification.

---

---

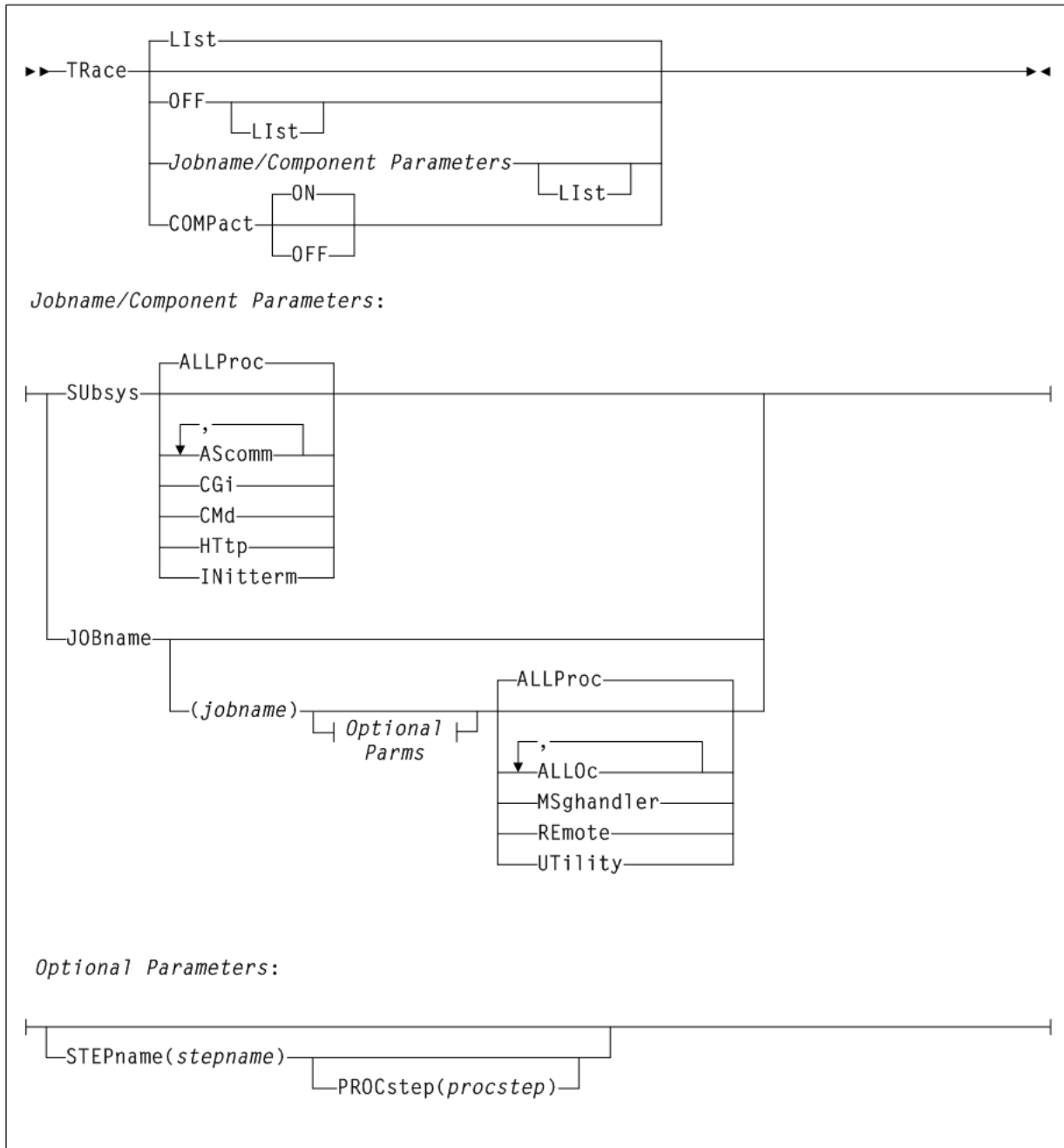
**Note:**

- SMC *TRace* processing may impact system performance and should be used only as directed by Oracle StorageTek Software Support.
  - *TRace JOBname (jobname)* causes SMC traces to be produced for UUI commands issued by the HSC or VTCS subsystem address space when *jobname* matches the HSC subsystem name. UUI tracing is extremely CPU intensive and should only be used as directed by Oracle StorageTek Software Support.
- 
- 

## Syntax

Figure 2–38 shows *TRace* command syntax.

**Figure 2–38** TRace command syntax



## Parameters

As shown in [Figure 2–38](#), the TRace command includes the following parameters:

### LIsT

optionally, lists current SMC trace settings for both the SMC subsystems and SMC components by job name, step name, and PROC step.

- LIsT is the default when no parameters are specified on the TRace command.
- LIsT may be specified with other parameters. In this case, the LIsT is generated after the other parameters are processed.

- If no `JOBname` is specified, all jobs with SMC components being traced are listed.
- If `JOBname` is specified, only those jobs with SMC components being traced whose job name matches the specified job `JOBname` are listed.

**OFF**

optionally, disables SMC tracing.

- If no other parameters are specified, tracing is disabled for all jobs.
- If only `JOBname` is specified, tracing is disabled for all entries for that job name.
- If `JOBname`, `STEPname` and `PROCstep` are specified, tracing is disabled only for the specified entry.

**COMPact**

Optionally, indicates whether SMC trace output should be produced in "compact" format.

When GTF trace output is produced in this format, you can use the SMC utility program `SMCUGTF` to reformat the trace to the original SMC trace format. See "[Trace Format Utility \(SMCUGTF\)](#)" on page 4-32.

**ON**

SMC trace output is produced in a format that conserves GTF data set space. This is the default.

**OFF**

SMC trace output is produced in human readable format.

**Jobname/Component Parameters****SUbsys**

optionally, enables or disables SMC tracing for one or more SMC subsystem components or subtasks.

**ALLProc**

optionally, enables tracing for all SMC subsystem components and tasks. This is the default if no parameters are specified.

**AScomm**

optionally, enables tracing for all SMC subsystem `ASCOMM` Component tasks.

**CGi**

optionally, enables tracing for all SMC subsystem CGI module processing for all connected client requests. To trace CGI module execution for individual client requests, issue a `TRace JOBname REMOTE` command on the appropriate SMC client.

**CMd**

optionally, enables tracing for all SMC subsystem Operator Command Component tasks.

**HTtp**

Optionally, enables tracing for all SMC subsystem HTTP Server Component tasks.

**INit**

Optionally, enables tracing for all SMC subsystem Initiation/Termination Component tasks.

**JOBname(jobname)**

optionally, enables or disables SMC tracing for one or more initiator components in initiators that match the specified *jobname*, *stepname*, and *procstep* name specified. If not specified, tracing information for all jobs is displayed.

*jobname* is a specific job name. The value entered must be one to eight characters in length. An asterisk (\*) can be used as a wildcard when included as the last character. In this case, all jobs whose job names match the characters preceding the asterisk are traced or listed. To trace all jobs, specify `TRACE JOB (*)`.

**ALLProc**

optionally, enables tracing for all SMC initiator components executing in initiators that match the specified *jobname*, *stepname*, and *procstep* name specified. This is the default if no parameters are specified.

**ALLOc**

optionally, enables tracing for all SMC Allocation tasks executing in initiators that match the specified *jobname*, *stepname*, and *procstep* name specified.

**MSghandler**

optionally, enables tracing for all SMC Message Handler Component tasks executing in initiators that match the specified *jobname*, *stepname*, and *procstep* name specified.

**REmote**

optionally, enables tracing in the remote SMC HTTP Server for requests originating from the *jobname*, *stepname*, and *procstep* name specified.

Remote tracing does not rely on the GTF trace being active on the SMC client, but GTF tracing must be active on the corresponding SMC HTTP Server.

**UTility**

optionally, enables tracing for all SMC Utility Component tasks executing in initiators that match the specified *jobname*, *stepname*, and *procstep* name specified.

**Optional Parameters****STEPname(stepname)**

optionally, specifies a step name.

*stepname* indicates the step name. The value entered must be one to eight characters in length. This parameter is only valid if *JOBname* is also specified.

**PROCstep(procstep)**

optionally, specifies a PROC step.

*procstep* indicates the PROC step. The value entered must be one to eight characters in length. This parameter is only valid if *JOBname* and *STEPname* are also specified.

**TREQDef****Interfaces:**

- Console, utility, *SMCCMDS* data set, or *SMCPARMS* data set
- UI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required, or may be input to the SMCUSIM utility

## Description

The SMC TREQDef command specifies and loads the definition data set containing your tape request (TAPEREQ) parameter statements.

---

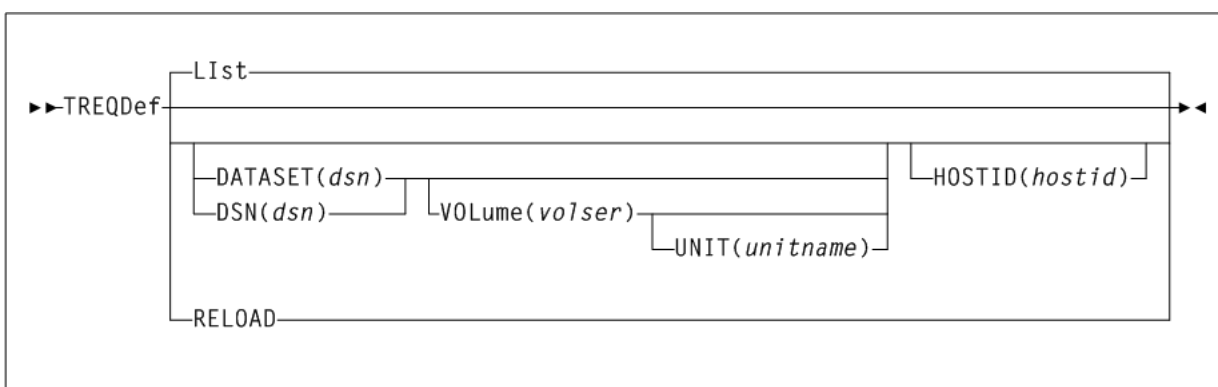
**Note:** Existing HSC TREQDEF control statements are not disabled, but are only processed for remote SMC 6.0 client systems.

---

## Syntax

Figure 2–39 shows TREQDef command syntax.

Figure 2–39 TREQDef command syntax



## Parameters

As shown in Figure 2–39, the TREQDef command includes the following parameters:

### List

lists information about the definition data set containing TAPEREQ control statements that are currently loaded. Information includes:

- Data set name
- Member name
- Identifying string (if the data set contains an OPTION TITLE statement)
- Date and time parameters were loaded

List is the default when no parameters are specified on the TREQDef command.

### DATASET(dsn) or DSN(dsn)

optionally, specifies the name of the data set containing the TAPEREQ control statements to be processed and, optionally, an OPTION TITLE control statement.

dsn indicates the data set name. This data set can be a fixed length 80-byte sequential data set, or a fixed length 80-byte member of a PDS. If the data set name includes a member name, you must enclose the data set name in quotes. For example:

```
DATASET('YOUR.PDS.DSN(MEMBER)')
```

**VOLume(volser)**

optionally, specifies the volume serial number of the DASD volume on which the data set resides. Specify this parameter if the data set is not catalogued.

*volser* indicates the volume serial number.

**UNIT(unitname)**

optionally, specifies the unit name of the specified VOLume for uncataloged data sets. Requires that VOLume also be specified.

*unitname* indicates the unit name. If not specified, then a unit name of SYSALLDA is the default.

**HOSTID(hostid)**

optionally, specifies the *hostid* from which to execute this control statement. This parameter is valid only for use in PARMLIB, to allow multiple systems to share a parmlib member containing multiple TAPEREQ control statement members.

*hostid* indicates the host name(s) from which to execute this control statement. Multiple host names must be separated by commas.

**RELOAD**

optionally, reloads the currently loaded TAPEREQ control statements from the same data set. This parameter can be used after a TREQDEF DSN(*dsm*) command is issued, and changes may then be made to the contents of that data set.

## TAPEREQ Control Statement

The SMC TAPEREQ control statement specifies a named POLicy to be applied to a particular job name, step name, program name, data set or group of data sets.

TAPEREQ statements are placed in the definition data set (DDname) specified by the SMC TREQDEF command. SMC allocation calls on these TAPEREQ statements to help ensure that the correct media type is used to satisfy a request, and that the cartridge is mounted on the appropriate device. TAPEREQ statements must be placed in the definition data set; they cannot be issued as operator commands.

The SMC POLicy command enables you to specify all tape policy specifications in a single named object. It is used with TAPEREQ statements or SMS to associate a named policy with an allocation or mount event.

The TAPEREQ POLicy parameter directs the SMC to reference an associated SMC policy defined by an SMC POLicy command. Refer to Oracle's ELS publication *Configuring and Managing SMC* for more information.

If any of the following conditions exist, then the MEDia and RECTech or MODEl parameters are set to undefined, which matches all requests. Any available device is eligible for allocation.

- There is no TAPEREQ control statement specified in the definition data set to match a request.
- There is no matching Policy command specifying the MEDia, RECTech, or MODEl parameters.
- There is no definition data loaded.



---

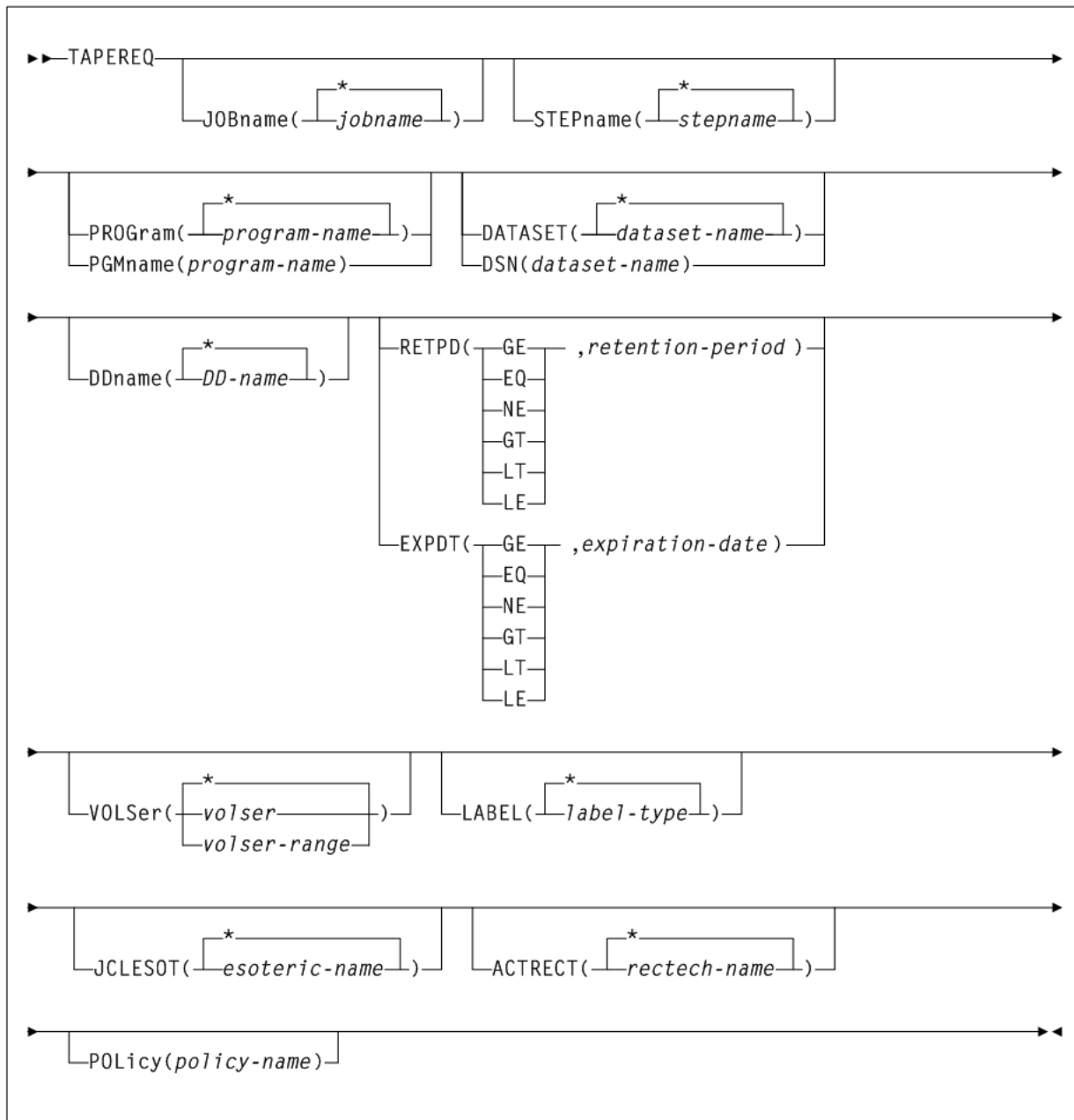
**Note:**

- In previous software releases, `TAPEREQ` parameters were used to define tape attributes. Refer to Oracle's ELS publication *ELS Legacy Interfaces Reference* for information about these legacy parameters.
  - See "[TREQDef](#)" on page 2-88 for information about the `TREQDEF` command.
- 

**Syntax**

[Figure 2-40](#) shows `TAPEREQ` control statement syntax.

Figure 2-40 TAPEREQ control statement syntax



### Parameters

As shown in Figure 2-40, the TAPEREQ control statement includes the following parameters:

---



---

**Note:** The `JOBname`, `STEPname`, `PROGname`, and `DATASET` parameter values can include the following wild card characters:

- % or ? indicates any single nonblank character.
  - \* indicates 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(jobname)**

optionally, identifies the job name. If `JOBname` is not specified, the default value is `*`.

*jobname* indicates the name of the job for which the `TAPEREQ` control statement is processed.

The `TAPEREQ` control statement is used only when the specified *jobname* matches the job name in the request.

---



---

**Note:** If you are coding a `TAPEREQ` statement for `DFHSM`, you must specify `JOBname` as a selection criteria. Using `DATASET (DSN)` can cause unpredictable results.

---



---

### **STEPname(stepname)**

optionally, specifies the step name. If the `STEPname` parameter is not specified, the default value is `*`.

*stepname* indicates the name of the step for which the `TAPEREQ` control statement is processed. The `TAPEREQ` control statement is used only when the specified *stepname* matches the step name in the request.

### **PROGname(program-name) or PGMname(program-name)**

optionally, specifies the program name.

The program name used is the job step program name. The default value is `*`.

*program-name* indicates the program name for which the `TAPEREQ` control statement is processed.

The `TAPEREQ` control statement is used only when the specified *program-name* matches the program name in the request.

---



---

**Note:** This parameter is not supported in JES3 environments.

---



---

### **DATASET(dataset-name) or DSN(dataset-name)**

optionally, specifies the data set name. The default value is `**`.

*dataset-name* indicates the data set name for which the `TAPEREQ` control statement is processed.

The `TAPEREQ` control statement is used only when the specified *dataset.name* matches the data set name in the request. Rules for `RACF` and `SMS` enhanced generic naming apply.

If you are coding a TAPEREQ statement for DFHSM, you must specify JOBname as a selection criterion. Using DATASET (DSN) 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,

```
DSN=&&ABC
```

may be coded on a DD statement if a data set is temporary. 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.

**DDname**

optionally, specifies the DD name of the data set.

*DD-name* indicates the DD name.

**RETPD**

optionally, specifies the retention period and the relationship that must exist. 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* indicates the retention period in days for the data set. Specify the number of days as a 1- to 4-digit decimal number. This parameter is not supported in JES3 environments.

---

---

**Note:** This parameter is not supported in JES3 environments.

---

---

**EXPDT**

optionally, specifies the expiration date and the relationship that must exist.

- 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* indicates the expiration date of the data set in YYDDD or YYYY/DDD format.

---

---

**Note:** This parameter is not supported in JES3 environments.

---

---

**VOLser(volser or volser-range)**

optionally, identifies a specific volume id or range of volsers. If `VOLSer` is omitted, the default value is `*` (any volume).

Using a combination of `TAPEREQ` statements and `POLICY` commands, SMC enables you to specify allocation policy based on specific volume serial numbers. Refer to Oracle's ELS publication *Configuring and Managing SMC* for more information.

*volser* or *volser-range* indicates the volume id or a range of volumes.

`VOLSer` requires that the `POLICY` parameter is also specified.

**LABEL(label-type)**

optionally, specifies the label type to be used to select the policy.

*label-type* indicates the label type. Specify one of the following:

- `SL` indicates standard-labeled tape. This is the default if the parameter is omitted.
- `AL` indicates ANSI-labeled tape.
- `NL` indicates nonlabeled tape.
- `NSL` indicates nonstandard labeled tape.

`LABEL` is valid only if `POLICY` is also specified.

**JCLESOT(esoteric-name)**

optionally, specifies the name of a device esoteric group to be matched with the esoteric supplied in the `JCL` (or via `IDAXESOT`).

*esoteric-name* indicates the esoteric group name.

This can consist of one to eight alphanumeric or national (`#`, `@`, `$`) characters. A slash (`/`) or hyphen (`-`) is also valid.

`JCLESOT` is valid only if `POLICY` is also specified. `JCLESOT` only matches at `SSI55` (`IDAX`) or `SSI24` (allocation) time, such that a `POLICY` containing `SUBPOOL` and `MGMTCLAS` will not be selected by this parameter.

The following example illustrates the use of the `JCLESOT` and `ACTRECT` parameters:

```
TAPEREQ DSN(V047970.JCLESOT.*) JCLESOT(HVTSS11) POLICY(POL1)
TAPEREQ DSN(V047970.JCLESOT.*) ACTRECT(V) POLICY(POL1)
TAPEREQ DSN(**) POLICY(POL2)
```

```
POLICY NAME(POL1) MED(V) REC(V) MGMT(HSCX400) SUBP(SUBP1)
POLICY NAME(POL2) MED(ECART)
```

In this example, jobs that specify `UNIT=HVTSS11` (with the matching `DSN`) will be assigned a `VIRTUAL` media and `rectech` at `SSI55/SSI24` time. All other data sets will be assigned a media of `ECART`. At mount message time, if the recording technique of the allocated drive is virtual for the matching `DSN`, then the management class and subpool from `POL1` are assigned. Otherwise, no subpool or management class is assigned.

**ACTRECT(rectech-name)**

optionally, specifies the recording technique to be used to determine the policy at mount time if the drive where the mount is requested conflicts with the previously selected policy, or if there is no previously selected policy.

For example, if the `JCL` esoteric prevents selection of a policy recording technique or esoteric substitution, the selected policy might have a scratch subpool inconsistent with the drive where the mount is requested.

Also, if criteria that are not available at mount time, such as JCLESOT, are used for policy selection, SMC mount processing may not have a policy available to assign mount parameters such as SUBPOOL and MGMTCLAS. Specifying ACTRECT allows a default override.

*rectech-name* indicates the recording technique name.

ACTRECT is valid only if POLICY is also specified.

**POLICY(policy-name)**

optionally, specifies the name of a policy (defined by an SMC POLICY command) to be associated with the allocation or mount request.

*policy-name* indicates the policy name.

The SMC uses this policy name to determine all policy information associated with an allocation or mount request.

The named policy must be defined using a POLICY command before the TREQDEF command is processed.

See "POLICY" on page 2-53 for more information about the SMC POLICY command. Refer to Oracle's ELS publication *Configuring and Managing SMC* for information about creating new POLICY definitions based on existing TAPEREQ statements.

**Disabling a TAPEREQ Definition**

Use the following procedure to disable a TAPEREQ definition:

1. Edit the definition data set containing the current TAPEREQ control statements.
2. Delete the parameters of the TAPEREQ control statement you want to disable, or delete the TAPEREQ control statement.
3. Issue the TREQDEF operator command to reload the definition data set.

To disable all TAPEREQ definitions, load a definition data set that contains only one TAPEREQ control statement with no parameters.

---

---

**Note:** If any individual TAPEREQ control statement is in error, the entire set is discarded.

---

---

## UExit

**Interfaces:**

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required, or may be input to the SMCUSIM utility

## Description

The SMC UExit command defines the load module invoked for a specific user exit, and specifies whether the user exit is enabled or disabled.

User exits 1, 2, 4, 8, 9, 10, 11, 12, and 13 are invoked and managed by the SMC.

The SMC UExit command defines which load module is invoked for a specific user exit, and whether the user exit is enabled or disabled.

User exits 1, 2, 4, 8, 9, 10, 11, 12, and 13 are invoked and managed by the SMC. Both HSC and MVS/CSC user exit formats are supported. Use the UExit command to specify the user exit format.

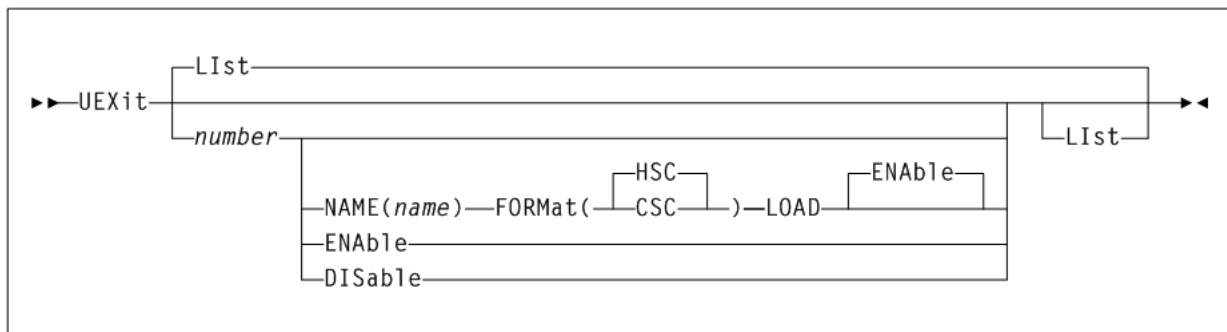
**Note:**

- Use this command to load each user exit you wish to use. If a user exit is not loaded, it will not be called.
- All user exits are executed on the host where the SMC resides, regardless of where the library server resides.
- Only one format of a given user exit can be invoked, regardless of the number of TapePlexes the SMC interacts with. The desired format (HSC or MVS/CSC) is specified using the FOrMat parameter.
- HSC User exit 7 (SLSUX07) is no longer supported.
- These user exits must reside in a library accessible to the subsystem.
- User exit information is not honored when an SMC POLiCy applies to an allocation, or a mount.
- Refer to Oracle’s ELS publication *ELS Legacy Interfaces Reference* for information about each user exit.

**Syntax**

Figure 2–41 shows UExit command syntax.

**Figure 2–41 UExit command syntax**



**Parameters**

As shown in Figure 2–41, the UExit command includes the following parameters:

**LIsT**

optionally, displays status information for all user exits.

- LIsT is the default when no parameters are specified on the UExit command.
- LIsT may be specified with other parameters. In this case, the LIsT is generated after the other parameters are processed.

**number**

optionally, specifies the user exit number. Valid values are 1, 2, 4, 8, 9, 10, 11, 12, and 13.

**NAME(name)**

optionally, specifies the user exit's load module name.

*name* indicates the load module name (for example, SLSUX01).

**FORMat**

optionally, specifies the format of user exit to be enabled or disabled.

**HSC**

the HSC user exit format.

**CSC**

the MVS/CSC user exit format.

**LOAD**

optionally, loads the specified user exit, making it available for use.

If a user exit is not loaded, it will not be called. A subsequent load of a module for the same user exit causes the currently active copy of the load module to be deleted when its use count reaches zero.

**ENable**

optionally, enables a user exit that was previously disabled due to a problem.

**DISable**

optionally, disables a user exit to allow for problem resolution.

## UNITAttr

**Interfaces:**

- Console or SMCCMDS data set
- UI Support: Yes (No XML/CSV output)

**Subsystem Requirements:**

Active SMC required, or may be input to the SMCUSIM utility

## Description

The SMC acquires drive type information from both the z/OS operating system and the ELS library control systems (TapePlexes). For the ELS library control systems, HSC data returned from configuration queries is used to augment information returned from the operating system for library-controlled devices.

The SMC `UNITAttr` command enables you to specify device addresses and attributes to augment operating system information for nonlibrary controlled devices, and effectively ignore configuration information from TapePlexes for specified devices when multiple TapePlexes define the same device address. The SMC `UNITAttr` command enables you to do the following:

- Set `MODEL=IGNORE` for device addresses not available for this host.
- Specify model types for nonlibrary devices on this host.



- Specify `NOTAPEplex` for a nonlibrary device address or range that is defined to multiple TapePlexes, but for this host the attached devices belong to the specified TapePlex.
- Specify TapePlex ownership and model for devices that may be referenced by a mount after the SMC is started but before the TapePlex is initialized.

Refer to Oracle's ELS publication *Configuring and Managing SMC* for information about configurations that may require or benefit from the use of `UNITAttr` commands.

---

---

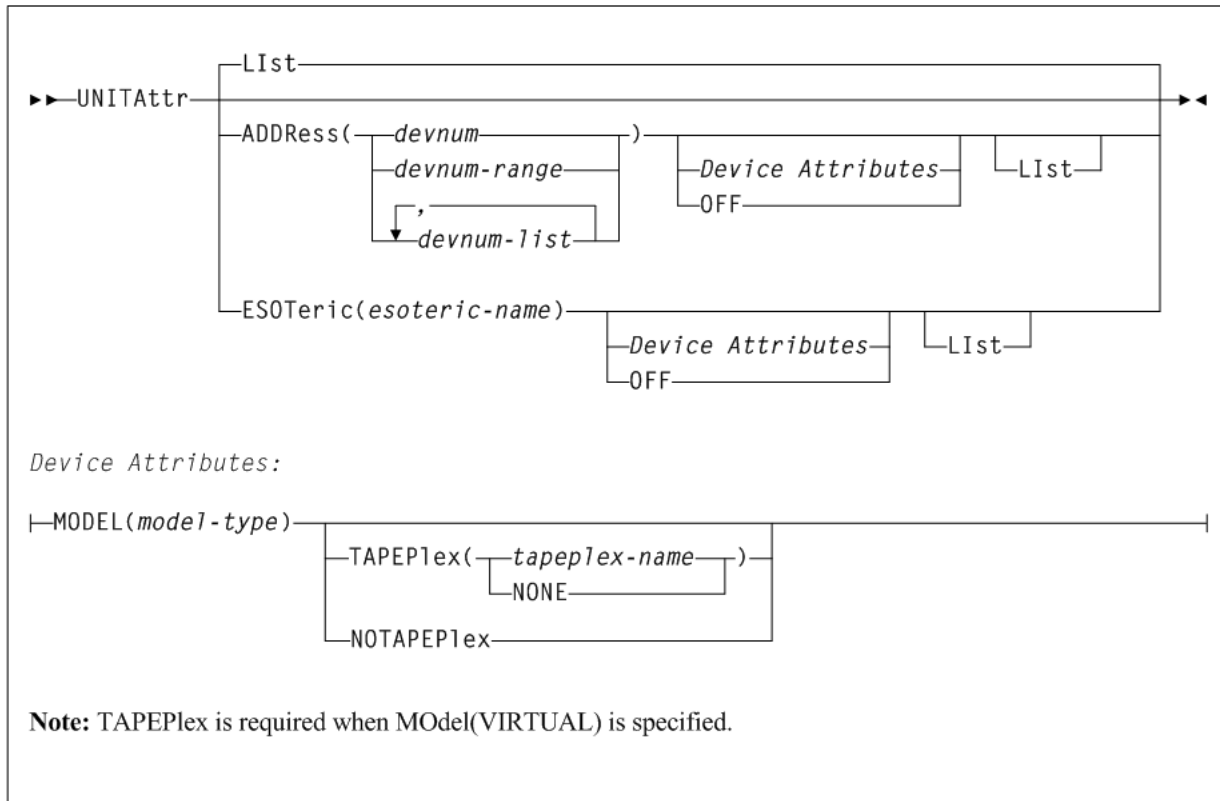
**Note:**

- Existing HSC `UNITATTR` control statements are not disabled. However, `UNITATTR` statements for nonlibrary and `MODEL (IGNORE)` devices are only processed for remote SMC 6.0 client systems. HSC `UNITATTR` statements for library and virtual drives are not used in any HSC processing after release 6.0.
  - SMC does not support a `UNITDEF` command. Each `UNITATTR` command is parsed and processed independently.
  - HSC nonlibrary and `MODEL=IGNORE` `UNITATTR` statements are no longer returned to SMC on a configuration query. Therefore, if you use `UNITAttr`s to bypass devices for allocation or to define the device type of nonlibrary drives, you must define these `UNITATTR` commands to the SMC subsystem.
- 
- 

## Syntax

Figure 2-42 shows `UNITAttr` command syntax.

Figure 2-42 UNITAttr command syntax



## Parameters

As shown in [Figure 2-42](#), the UNITAttr command includes the following parameters:

### ADDRESS(*devnum*, *devnum-range*, or *devnum-list*)

optionally, specifies device numbers for which UNITAttr device attributes are assigned.

*devnum*, *devnum-range*, or *devnum-list* indicates the device number, device number range, or device number list.

### ESOTeric(*esoteric-name*)

optionally, specifies an esoteric for which UNITAttr device attributes are assigned. When device characteristics are modified, all of the devices in the specified esoteric are modified.

*esoteric-name* indicates the MVS esoteric name. This name must be defined to MVS, and must contain only tape devices known to SMC.

### OFF

optionally, specifies that the model type for the device numbers specified by the ADDRESS parameter or ESOTeric parameter should be determined from the device UCB, or, for a device which was previously set to MODEL=IGNORE, that the MODEL information from the library server should be restored.

### List

lists the UNITAttr model information for the device numbers specified by the ADDRESS or ESOTeric parameter.

## Device Attributes

### MODEL(model-type)

optionally, specifies the model type of the transport to be assigned the device numbers specified by the ADDRESS parameter.

*model-type* indicates the model type.

See [Appendix A, "MEDia, RECTech, and MODel Values"](#) for a list of valid *model-type* values.

### TAPEPlex

optionally, specifies the name of a TapePlex to be associated with the drive before communication is established.

TAPEPlex is required when MODel (VIRTUAL) is specified. MODel (VIRTUAL) drives cannot exist as "non-library" devices.

#### tapeplex-name

the TapePlex name.

If specified, this name must match a TapePlex name defined in a TAPEPlex command.

### NONE

optionally, removes the TapePlex association from the specified device addresses.

### NOTAPEPlex

optionally, specifies that the specified address or address range are nonlibrary devices, even when configuration queries from ELS library control systems indicate that the device is library controlled. Use this parameter when a local nonlibrary device address overlaps a TapePlex owned device that is not accessible from his host.

---

---

#### Note:

- When a MODel is specified with TAPEPlex, the configuration query model information returned from the TapePlex is assumed to be the "real" or "correct" model, and it overrides any MODel specified in the UNITAttr command.
  - If TAPEPlex is specified for an address, but that address is not returned from the configuration query for that TapePlex, an error message is issued, but the erroneous TapePlex ownership remains with the specified device address. As a result, any mounts destined for the specified device address are not automated and are rejected by the specified TapePlex.
- 
- 

## USERMsg

### Interfaces:

- Console or SMCPARMS data set
- UUI Support: Yes (No XML/CSV output)

### Subsystem Requirements:

Active SMC required, or may be input to the SMCUSIM utility

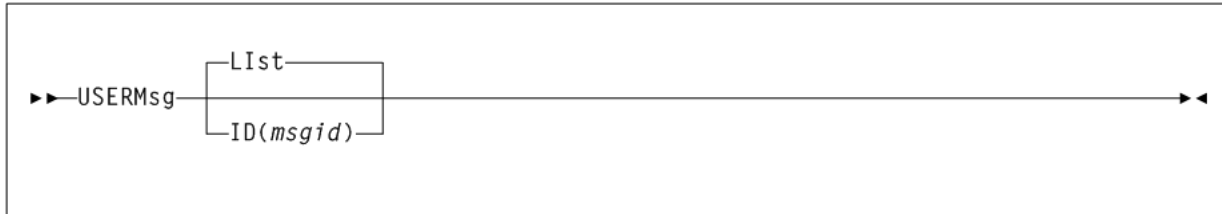
## Description

The SMC `USERMSG` command enables you to specify additional message identifiers for messages to be intercepted and passed to the library subsystem's user exit 01. You can also use this command to list the message identifiers for messages that have been defined using the `USERMSG` command.

## Syntax

Figure 2–43 shows `USERMSG` command syntax.

**Figure 2–43** *USERMSG command syntax*



## Parameters

As shown in Figure 2–43, the `USERMSG` command includes the following parameters:

### List

optionally, lists the message identifiers in the customer-defined message table. This is the default if no parameters are specified.

### ID(msgid)

optionally, specifies a message to be added to the customer-defined message table. This message is passed to user exit 01.

*msgid* indicates the message identifier for the message to be added.

---

---

**Note:** This parameter may only be specified in the `SMCPARMS` data set.

---

---

## VMSG

### Interfaces:

- Console, utility, `SMCCMDS` or `SMCPARMS` data set
- UI Support: Yes (no XML/CSV output)

### Subsystem Requirements:

Active SMC required

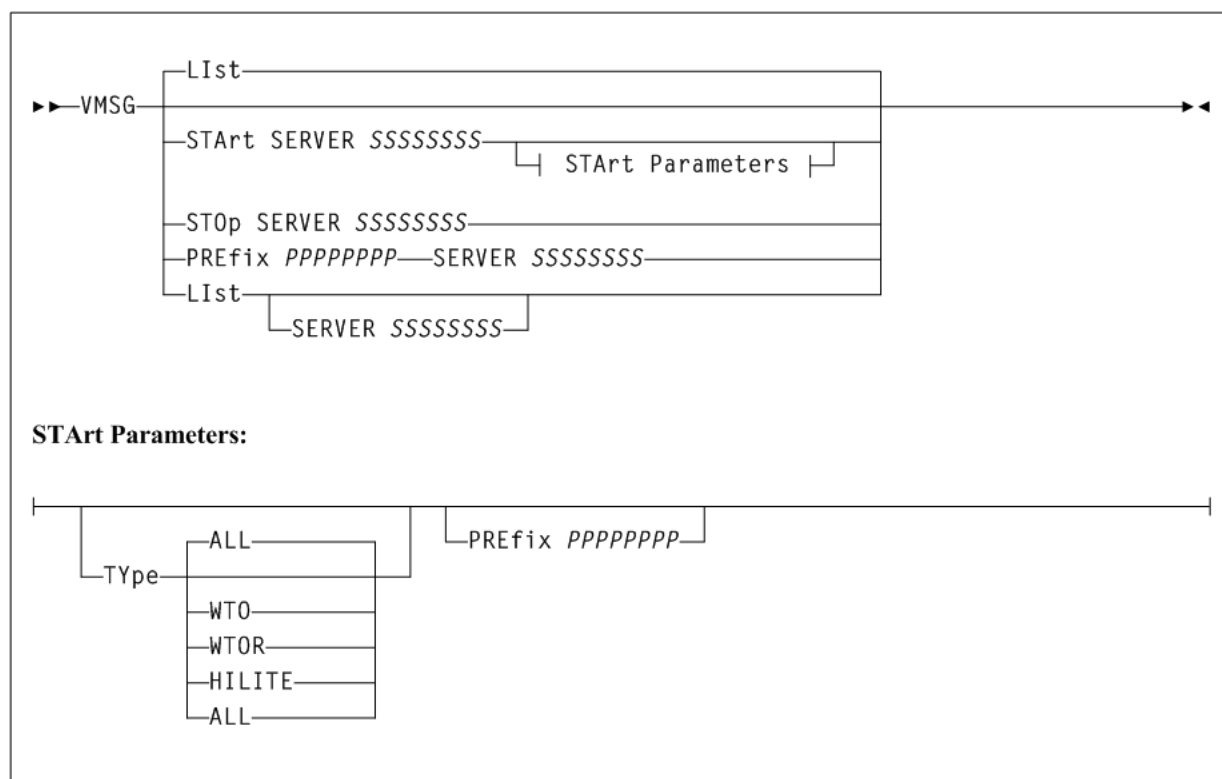
## Description

The SMC `VMSG` command enables you to start or stop the Virtual Storage Manager console (VSMc) message processor client. The `VMSG` message processor client enables the local SMC subsystem to receive and reply to messages issued by the remote VSM console server.

## Syntax

Figure 2–44 shows VMSG command syntax.

Figure 2–44 VMSG command syntax



## Parameters

As shown in Figure 2–44, the SMC VMSG command includes the following parameters:

### List

optionally, displays VMSG task status information.

#### **SERVER SSSSSSSS**

optionally, lists only the VMSG processor for the named server.

SSSSSSSS indicates the server name. If specified, then the named server must have been previously defined by an SMC SERVer command.

L ist is the default when the S TArt, S TOp, or P REfix keywords are not specified.

### STArt

optionally, starts a VMSG message processor client.

#### **SERVER SSSSSSSS**

specifies the server name for the VMSG processor.

SSSSSSSS indicates the server name. The server name is required and must have been previously defined by an SMC SERVer command.

#### **TYpe (type-list)**

optionally, defines a type-list containing one or more type(s) of messages that are received by the VMSG client.

Specify one or more of the following in the *type-list*, using a comma to separate values:

**ALL**

Receive all messages. This is the default. If **ALL** is specified, it cannot be specified with any other message types. This is the default.

**HILITE**

Receive highlighted WTO messages.

**WTO**

Receive non-highlighted WTO messages.

**WTOR**

Receive messages that require a reply.

**PREfix P P P P P P P P**

optionally, specifies the message prefix identifier that will identify messages from this VSM console server in the SMC subsystem log. If not specified, then the specified *server* name is used as the message prefix.

*PPPPPPPP* is the message prefix identifier. The prefix can be a maximum of 8 characters in length and can contain any combination of the following:

- A-Z
- 0-9
- @#\$,.(+ -= < | ! ; % > ? :

**STOp**

optionally, stops a VMSG message processor client.

**SERVER S S S S S S S S**

specifies the server name for the VMSG processor.

*SSSSSSSS* indicates the server name. The server name is required and must have been previously defined by an SMC *SERVER* command.

**PREfix P P P P P P P P**

optionally, specifies the message prefix identifier that will identify messages from this VSM console server in the SMC subsystem log. If not specified, then the specified server name is used as the message prefix.

*PPPPPPPP* indicates the message prefix identifier. The prefix can be a maximum of 8 characters in length and can contain any combination of the following:

- A-Z
- 0-9
- @#\$,.(+ -= < | ! ; % > ? :

**SERVER S S S S S S S S**

specifies the server name for the VMSG processor.

*SSSSSSSS* indicates the server name. The server name is required and must have been previously defined by an SMC *SERVER* command.

## XCLIENT

### Interfaces:

- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UUI Support: Yes, supports XML and CSV

### Subsystem Requirements:

Active SMC required, or may be input to the SMCUSIM utility

## Description

The SMC XCLIENT command enables you to define XAPI clients that use a different protocol version than the default server XAPI security protocol.

---



---

### Note:

- The XCLIENT command is only necessary to define clients that use the older "unsecure" protocol. If an XAPI request originates from a client that is not defined with a XCLIENT command, it will be challenged to produce XAPI login credentials.
- If you have back-level ELS or non ELS clients communicating with your ELS 7.3 SMC HTTP server, then they must be defined as protocol version 0 clients with the XCLIENT command.
- The entire XCLIENT command is not echoed in the SMC0000 message.

XCLIENT (suppressed) is echoed for all XCLIENT commands.

- The entire XCLIENT command is not logged when the SMC LOG command is active.

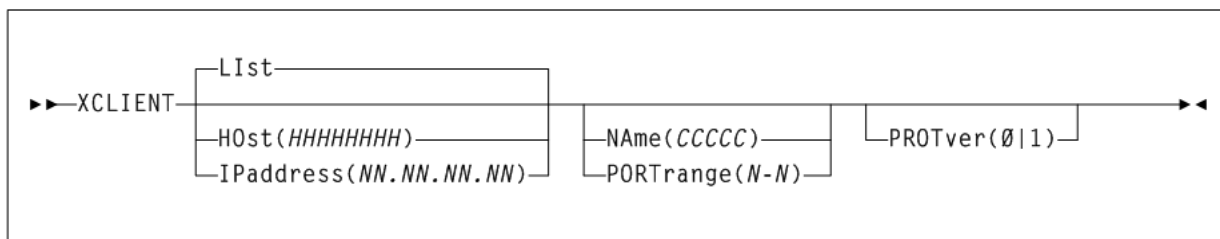
XCLIENT (suppressed) is logged for all XCLIENT commands.

- XCLIENT commands must be specified before any HTTP or SERVER commands.
- 
- 

## Syntax

Figure 2-45 shows XCLIENT command syntax.

**Figure 2-45 XCLIENT command syntax**



## Parameters

As shown in Figure 2-45, the XCLIENT command includes the following parameters:

**List**

optionally, displays all XCLIENT settings. **List** is the default when no parameters are specified on the XCLIENT command.

**Host(HHHHHHHH)**

optionally, specifies the IP resolver host name on which the client resides.

*HHHHHHHH* indicates the name of the client. The **Host** name must be a resolvable name in the TCP\_IP name table. The following rules apply:

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

**IPaddress(NN.NN.NN.NN)**

optionally, specifies the IP address for the client.

*NN.NN.NN.NN* indicates The TCP/IP address.

---

---

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

---

---

**Name(CCCCC)**

optionally, specifies the name of the client. If the client is SMC/MVS, the **Name** specified should be the name returned as *client\_subsystem\_name*. Otherwise, the **Name** specified should be the name returned in the *client\_name*. If **Name** is specified as "\*", then any request from the specified **Host** or **IPaddress** will be defined as using the specified protocol version.

*CCCC* indicates the name of the client.

**PORTrange(NNNNN-NNNN)**

optionally, specifies the range of client ports from which the XAPI request originates on the specified **Host** or IP address that are allowed to use the specified protocol version.

*NNNNN-NNNN* indicates the port range. Valid ports are 1 to 65535. The range may contain 10 to 1000 ports.

---

---

**Note:** Use **PORTrange** if the SMC/MVS TCPIP PORTRANGE command has been specified to restrict SMC/MVS client ports to the specified range. **Name** and **PORTrange** are mutually exclusive.

---

---

**PROTver**

optionally, specifies the protocol version. The default is 0.

- 0 indicates the older "unsecure" protocol version.
- 1 indicates the XAPI security protocol version.

## XUDB

**Interfaces:**



- Console, utility, SMCCMDS data set, or SMCPARMS data set
- UII Support: Yes, supports XML and CSV

**Subsystem Requirements:**

Active SMC required, or may be input to the SMCUSIM utility

## Description

The SMC XUDB command enables you to add, update, delete, and list XAPI Security users.

The XAPI security user list is maintained by both the client and server. In cases where the client and server reside on the same host (that is, SMC/MVS), the XUDB user list is shared.

---



---

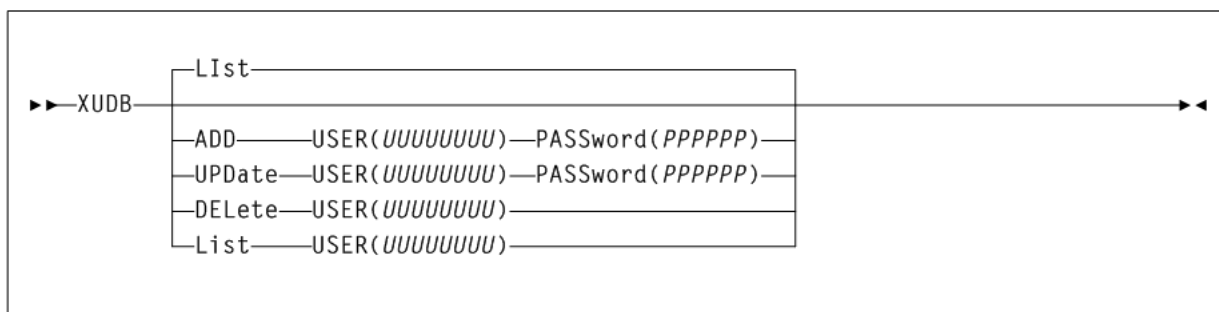
**Note:**

- The same USER must be defined on both the client and server.
  - A USER is unique by SERVER. If the same SMC HTTP server is servicing multiple TAPEPLEXes, any defined users will have access to all TAPEPLEXes.
  - The entire XUDB command is not echoed in the SMC0000 message. XUDB (suppressed) is echoed for all XUDB commands.
  - The entire XUDB command is not logged when the SMC LOG command is active. XUDB (suppressed) is logged for all XUDB commands.
  - The XUDB commands must be specified before any HTTP, SERVER or TCPip commands.
- 
- 

## Syntax

Figure 2-46 shows XUDB command syntax.

**Figure 2-46** XUDB command syntax



## Parameters

As shown in Figure 2-46, the XUDB command includes the following parameters:

**List**

displays the specified user on the XAPI security list. `LISt` is the default when no parameters are specified on the `XUDB` command. In this case, all users are listed.

**USER (UUUUUUUU)**

specifies the user name to be listed. `UUUUUUUU` indicates the user name.

**ADD**

specifies that the specified user name or password is to be added to the XAPI Security list.

**USER (UUUUUUUU)**

specifies the user name to be added. `UUUUUUUU` indicates the user name.

**PASSword (PPPPPP)**

specifies the user password. `PPPPPP` indicates the password.

**UPDate**

specifies that the specified user name or password is to be updated in the XAPI Security list.

**USER (UUUUUUUU)**

specifies the user name to be added. `UUUUUUUU` indicates the user name.

**PASSword (PPPPPP)**

specifies the user password. `PPPPPP` indicates the password.

**DElete**

specifies that the specified user/password is to be deleted from the XAPI Security list.

**USER (UUUUUUUU)**

specifies the user name to be added. `UUUUUUUU` indicates the user name.

## XUDB Security Data set

Oracle recommends that you maintain the XAPI Security user list in a sequential data set, to include `XUDB ADD` entries for the XAPI Security user list.

In your SMC startup parameters, include an `SMC READ` command for your XAPI security data set.

The `XUDB ADD/UPDATE/DELETE` command can be used to modify the XAPI Security user list on the operational SMC/MVS. Any changes must also be reflected in the XAPI SECURITY data set.

The XAPI Security data set will be secured by `SAF (RACF)` permissions.

---

## HSC Commands, Utilities, and Control Statements

---

This chapter contains reference information about HSC and VTCS commands and control statements.

---

**Note:**

- Control statements that are loaded by an operator command are described along with that command. Other control statements, including those specified in the `PARMLIB` data set, are described independently.
  - Certain HSC and VTCS commands are described in the *ELS Legacy Interfaces Reference*. These commands were introduced in a pre-ELS software release and their functionality has been replaced in ELS. These commands are supported by ELS, however, this support will end in a future release.
- 

### ACTivities

**Interfaces:**

- `SLUADMIN` utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

### Description

The `ACTivities` command generates a statistical report of library resource loading by volume groups (for example, scratch compared to nonscratch, pass-thru activity, mounts, dismounts, entered, and ejected). This report provides information to help you monitor library resources and usage.

The utility also requires the SMF data for the time period requested (see "`SET SMF`" on page 3-335). It is assumed that you elect to have cartridge movement SMF records recorded in the time period specified. It is also assumed that the SMF records are in sequence by:

- SMF ID
- date

- time, from earliest to latest.

Duplicate or out of sequence records cause the utility to terminate and to display a message indicating that the SMF records are not sorted. See "Additional Sort Control Cards" on page 3-4 for sort statements you can specify to avoid this situation.

---

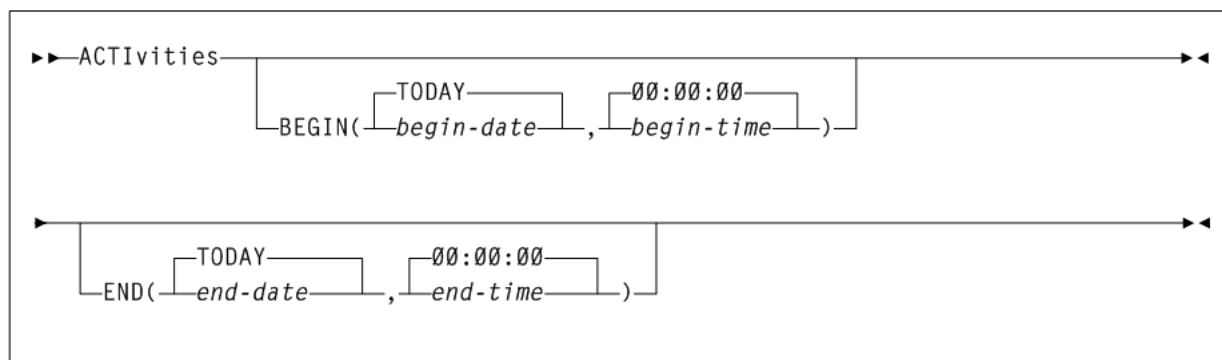
**Note:** Periodic performance statistics are not available for the SL8500 library. Volume movement statistics, including enter, eject, mount, dismount, and move, are produced for all library types.

---

## Syntax

Figure 3-1 shows ACTivities command syntax.

Figure 3-1 ACTivities command syntax



## Parameters

As shown in Figure 3-1, the ACTivities command includes the following parameters:

### **BEGIN(begin-date or TODAY),begin-time)**

optionally, specifies the beginning of the period for the activities analysis.

#### **begin-date or TODAY**

the starting date of the analysis.

- Specify a date in *mm/dd/yy* or *yyyymmdd* format.
  - When using this format, specify a *yy* value of 71 or greater to indicate a 20th century (19*xx*) date. Specify a *yy* value of 70 or less to indicate a 21st century (20*xx*) date. To avoid confusion, you are encouraged to use the *yyyymmdd* date format.
- Specify TODAY for the default date.

#### **begin-time**

the beginning time-of-day (24-hour value), expressed in *hh:mm:ss* format.

The allowable range for *begin-time* and *end-time* parameters is 00:00:00 to 24:00:00. The default value is 00:00:00.

In each of the following examples, activity reporting begins on October 27, 2008 at midnight:

```

BEGIN(10/27/08,00:00:00)
BEGIN(20081027,00:00:00)
BEGIN(TODAY,00:00:00)

```

```
BEGIN(,00:00:00)
```

### **END(end-date or TODAY),end-time)**

optionally, specifies the beginning of the period for the activities analysis.

#### **end-date or TODAY**

the ending date of the analysis.

- Specify a date in *mm/dd/yy* or *yyyymmdd* format.

When using this format, specify a *yy* value of 71 or greater to indicate a 20th century (19*xx*) date. Specify a *yy* value of 70 or less to indicate a 21st century (20*xx*) date. To avoid confusion, you are encouraged to use the *yyyymmdd* date format.

If *begin-date* is specified, *end-date* must be in the same format.

- Specify TODAY for the default date.

#### **end-time**

the ending time-of-day (24-hour value), expressed in *hh:mm:ss* format.

The allowable range for *begin-time* and *end-time* parameters is 00:00:00 to 24:00:00. The default value is 23:59:59.

For example:

```
END(10/27/08,18:00:00)
END(20081027,11:30:00)
END(TODAY,23:29:00)
END(,23:59:59)
```

## **Additional JCL Requirements**

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statements apply to the ACTivities JCL:

### **SLSSMF**

SMF data which the utility reports against. This may be a number of data sets concatenated from oldest to newest. This is an off-loaded copy of the active SMF data sets, created by the SMF dump utility IFASMFDP.

The SMF record type defined to HSC is the only SMF record used by ACTivities. To reduce the ACTivities run time, use the IFASMFDP utility to create a new SMF data set containing only the SMF record type defined to HSC. The following example can be used to create the HSC only SMF data set:

```
//jobcard
//SMFCOPY EXEC PGM=IFASMFDP
//MAN DD DISP=SHR,DSN=input.smf.data
//DUMPOUT DD DISP=(NEW,CATLG,DELETE),UNIT=your_esoteric,
// SPACE=(CYL,(500,500),RLSE),DSN=hsc.only.data
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INDD(MAN,OPTIONS(DUMP))
OUTDD(DUMPOUT,TYPE(your_HSC_SMF_record_number))
.
```

Note - A DISPLAY CDS command can be issued to find your\_HSC\_SMF\_record\_number.

### Additional Sort Control Cards

You must specify several additional control cards in the SYSIN portion of the SYELSORT JCL to make sure SMF records are sorted.

```
//SYSIN DD#
  INCLUDE COND=(6,1,BI,EQ,X'xx')
  SORT FIELDS=(15,4,CH,A,11,4,PD,A,7,4,BI,A)
  SUM FIELDS=NONE
```

---



---

**Note:** In the INCLUDE COND statement, the SMF type setting, shown as X'xx above, is supplied by the user. The default setting is FF (SMF type 255).

---



---

## ACTMVCgn

### Interfaces:

- SLUADMIN utility only
- UUI Support: Yes

### Subsystem Requirements:

Active HSC required only when specifying the MVCPOOL parameter

## Description

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 Oracle's ELS publication *ELS Disaster Recovery and Offsite Data Management Guide* for examples of use of this utility function.

---

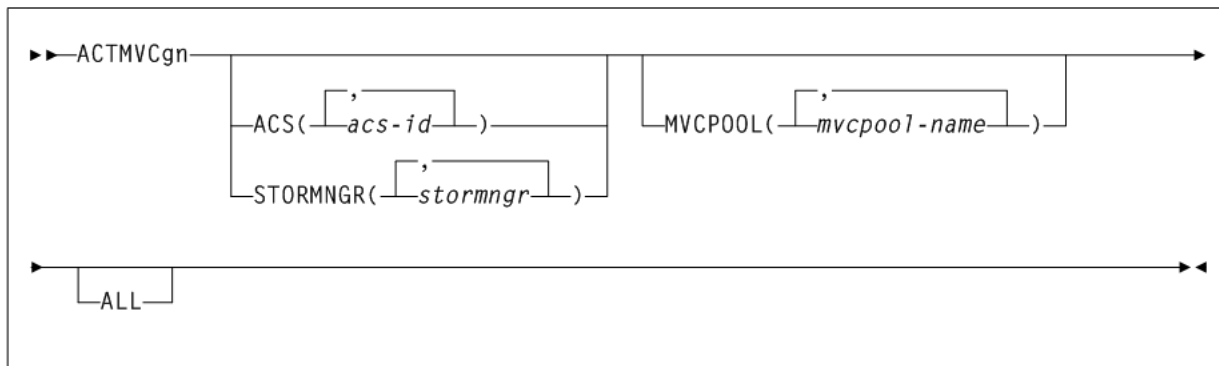


---

## Syntax

Figure 3–2 shows ACTMVCgn command syntax.

Figure 3–2 ACTMVCgn command syntax



## Parameters

As shown in [Figure 3–2](#), the ACTMVCgn command includes the following parameters:

### ACS(acs-id)

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

*acs-id* indicates 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.

### STORMNGR(stormngr-name)

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

*stormngr-name* indicates the STORMNGR name. This name can include a maximum of 8 characters. Blanks are not valid. To specify multiple STORMNGR names, separate each name with a comma.

### MVCPOOL(mvcpool-name)

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

*mvcpool-name* indicates the MVCPOOL name. This name can include a maximum of 13 characters. Blanks are not valid. To specify more than one MVCPOOL, 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.

---

### ALL

optionally, requests that READONLY (ON) and READONLY (OFF) control statements be generated for all non-empty MVCs. If ALL is not specified, READONLY control statements are not generated for full MVCs.

## Additional JCL Requirements

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statements apply to the ACTMVCGN JCL:

**SLUSMAUD**

Output in the form of SLUADMIN utility control statements to audit selected MVCs. This statement is optional; if it is present, then AUDIT utility control statements are generated for all non-empty MVCs, except for those in EXPORT status.

**SLUSMVON**

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

**SLUSMVOF**

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

## ARCHive

**Interfaces:**

- Utility only
- UI Support: Yes

**Subsystem Requirements:**

Active HSC not required

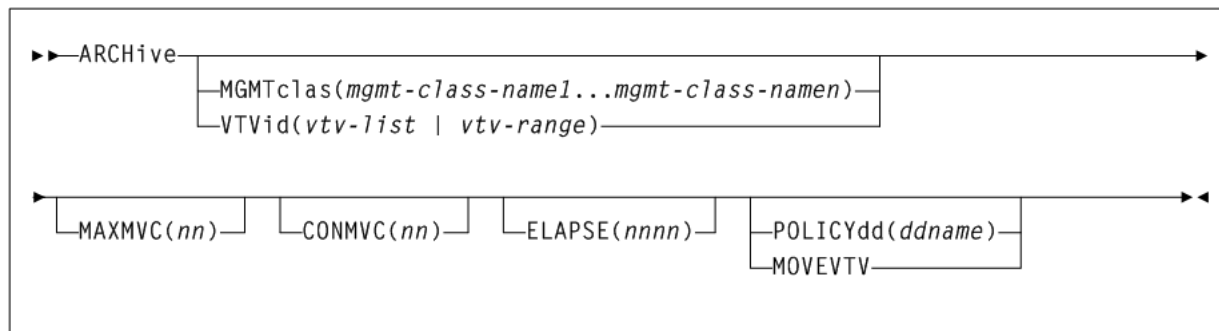
## Description

The ARCHive command archives VTVs, moving them to MVCs in a different Storage Class.

## Syntax

Figure 3–3 shows ARCHive command syntax.

**Figure 3–3 ARCHive command syntax**



## Parameters

As shown in Figure 3–3, the ARCHive command includes the following parameters:

**MGMTclas(mgmtclas-name1...mgmtclas-namen)**

optionally, specifies one or more Management Classes that define Archive Management Policies.

*mgmtclas-name1...mgmtclas-namen* indicates one or more Management Classes.



**VTVid(vtv-list or vtv-range)**

optionally, specifies a list or range of VTVs. This parameter is mutually exclusive with `MGMTclas`.

---

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

---

**MAXMVC(nn)**

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

*nn* indicates the maximum number of MVCs.

**CONMVC(nn)**

optionally, specifies the maximum number of MVCs (*nn*) 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)**

optionally, specifies the maximum time for the archive in minutes (*nnnn*). 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.

**POLICYdd(ddname)**

optionally, specifies the *ddname* of a file containing an alternate `MGMTclas` statement. This parameter is mutually exclusive with `MOVEVTV`.

**MOVEVTV**

optionally, 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 (no VTVs are moved). This parameter is mutually exclusive with `POLICYdd`.

## Examples

The following examples provide examples of the following ARCHive reports:

**ARCHive Report (MOVEVTV not specified)**

The following example shows an example of an ARCHive report (`MOVEVTV` not specified).

**Example 3-1 ARCHive report (MOVEVTV not specified)**

```

SLUADMIN (7.3.0)                STORAGETEK ENTERPRISE LIBRARY SOFTWARE UTILITY
SSYS=HSCI          PAGE 00002

TIME 15:50:09                    ARCHIVE VTV REPORT
DATE 2012-06-17

VTV      SIZE    %COMP    <-----CREATION----->    MGMT
```

VOLSER	(MB)		DATE	TIME	CLASS	MVC1	MVC2	MVC3
MVC4								
Y00001	1.62	39	2008JAN15	04:11:18	MJ4	02250	02251	02252
02253								
Y00002	1.62	39	2008JAN15	04:03:57	MJ4	02250	02251	02252
02253								
Y00003	1.62	39	2008JAN15	03:50:59	MJ3	02254	02255	
Y00004	1.62	39	2008JAN15	03:45:04	MJ2	02256	02257	02268
Y00005	0.01	0	2008JAN17	23:56:00	MJ1	02259		
Y00006	0.01	0	2008JAN17	23:41:37	MJ1	02259		
Y00007	1.62	39	2008JAN05	06:15:46	MJ4	02250	02251	02252
02253								

TOTAL VTVS=23                      TOTAL SIZE=29.32MB

TIME 15:50:09  
DATE 2012-06-17

ARCHIVE MVC REPORT

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

**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 re-migrated.

**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 Oracle 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 100 MB VTV compresses to 40 MB 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.

**Fields (MVC Report)**

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 re-migrated.

**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.

**ARCHive MOVEVTV Report (MOVEVTV specified)**

The following example shows an example of an ARCHive MOVEVTV report.

**Example 3-2 ARCHive MOVEVTV report**

```
SLUADMIN (7.3.0)      StorageTek Enterprise Library Software Utility      PAGE 0001
TIME 10:07:10                ARCHIVE MOVEVTV REPORT                DATE 2015-03-31
```

```
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
```

## AUDit

### Interfaces:

- Utility only
- Yes, when MVC or VTSS is specified

### Subsystem Requirements:

- Active HSC/VTCS (AUDit MVC, VTSS, or INVLDMIR)
- Active HSC at FULL service level (all others)

## Description

The AUDit command performs a volume inventory audit for the following:

- ACS or LSM
- One or more specified LSMs within an ACS
- One or more panels within an LSM
- One or more rows within a panel
- One or more columns (cells) within a row
- One or more VTSSs
- One or more MVCs

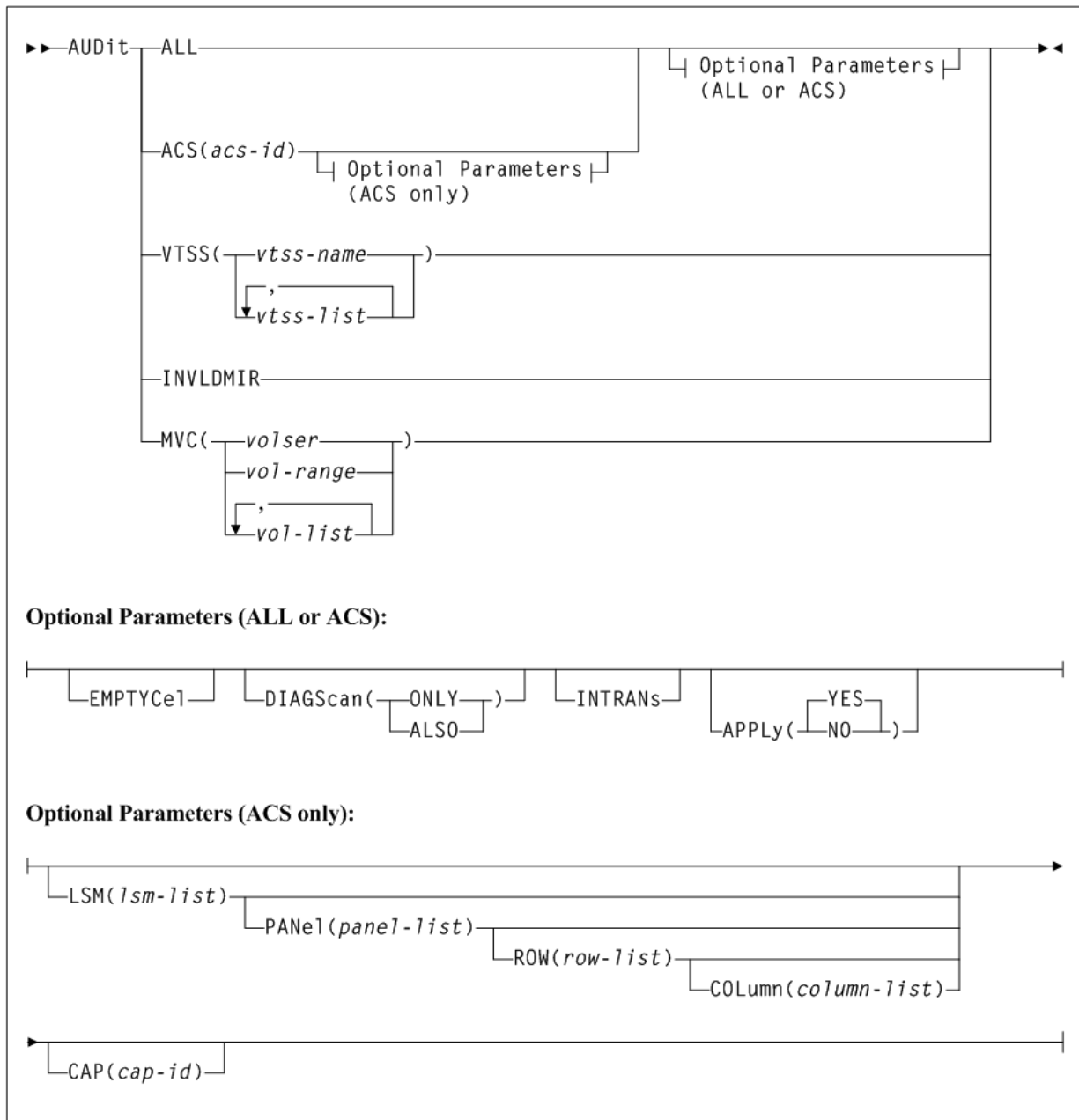
Optional parameters enable you to do the following:

- Update the library control data set to reflect cartridges observed.
- Produce a discrepancy list and do not update the control data set.

## Syntax

[Figure 3–4](#) shows AUDit command syntax.

Figure 3-4 AUDit command syntax



### Parameters (Real Volume Audit)

As shown in Figure 3-4, the **AUDit** command includes the following parameters:

#### **ALL**

optionally, specifies that the total library is to be audited.

#### **ACS(*acs-id*)**

optionally, specifies that only a particular ACS in the library is to be audited.

*acs-id* indicates a one or two digit hexadecimal ACS identifier.

---

---

**Note:** If the ACS parameter is specified with no LSM parameters, the audit is performed on all LSMs in the ACS at the same time (multitasking). To audit only certain LSMs within an ACS, you must code the LSM parameter with the LSM or list of LSMs.

---

---

### Optional Parameters (ALL or ACS)

The following parameters may be specified with either ALL or ACS:

#### EMPTYCe1

optionally, specifies that only empty cells are to be audited. This parameter cannot be used in combination with AUDIt parameter DIAGScan (ONLY).

The main benefit provided by auditing only empty cells is reducing the time required to run an audit.

---

---

**Note:** Oracle does not recommend running an empty cell audit on an SL8500 library because the time to run the audit is not substantially reduced.

---

---

Typically, users may specify this parameter when:

- they have entered an LSM and placed new cartridges in empty cells, or
- they need to correct cartridge location information for library volumes that have been manually moved to empty cells.

In either case, if APPLY (YES) is specified, the CDS is updated to reflect that these cells are no longer empty.

---

---

**Caution:** In either case, if APPLY (YES) is specified, the CDS is updated to reflect that these cells are no longer empty.

---

---

#### DIAGScan

optionally, specifies that diagnostic cells are to be scanned. The HSC scans these cells one at a time and displays the contents of each cell in the AUDIt utility report.

DIAGScan is mutually exclusive with ROW and COLUMN. It can be used in combination with all other AUDIt parameters.

#### ONLY

specifies that only diagnostic cells should be scanned. This parameter cannot be specified in combination with the EMPTYCe1 parameter.

#### ALSO

specifies that diagnostic cells should also be scanned along with the normal AUDIt utility operations.

---

---

**Note:** Cartridge movement between diagnostic cells and regular storage or CAP cells is not supported by the LMU, so diagnostic cell contents can only be reported.

---

---

**INTRANS**

optionally, specifies that in-transit cartridges in an LSM are to be processed. All in-transit cartridges, except those identified in the note below, are read and ejected as part of an AUDit utility operation. INTRANS is mutually exclusive with APPLY(NO).

- Only the first two cells (columns 0 and 1) in a 9310 playground are accessible to an in-transit audit. Cartridges in other cell positions in a 9310 playground (columns 2 through 5) cannot be ejected by an in-transit operation.
- 9740 diagnostic cells cannot be scanned by an audit.

**APPLY**

optionally, specifies whether the control data set is updated.

**YES**

specifies that the Audit operation ejects cartridges with duplicate volume serial numbers, unreadable or illegal external media labels, and/or new cartridges unreadable external labels, and performs corrective actions to the control data set to reflect the physical contents of storage cells. YES is the default.

**NO**

specifies that a discrepancy listing is produced, and the control data set is not updated.

**Optional Parameters (ACS Only)**

The following parameters may only be specified with ACS:

**LSM(lsm-list)**

optionally, specifies that only certain LSMs within the specified ACS are to be audited.

*lsm-list* is a single LSM number or a list of LSM numbers. An LSM number is a hexadecimal value (00-17). An LSM range is not allowed. If a list is specified, the elements must be separated by blanks or commas, and the entire list enclosed in parentheses.

**PANel(panel-list)**

optionally, specifies that only certain panels within an LSM are to be audited. If this parameter is specified, the LSM parameter may contain only one LSM number.

*panel-list* is a single panel or a list of panels. Panel ranges are not allowed. If a list is specified, the elements must be separated by blanks or commas, and the entire list enclosed in parentheses. A *panel-list* element is a one or two digit decimal number, specific to the LSM type. See "Panels" on page B-2 for a list of valid values.

The SL3000 library has static panels numbers starting with the left-most back panel (panel 0) and ending with the right-most front panel (panel 23) on a fully-configured library. The panel numbering starts with the BDM, the only required module (panels 12 and 13), and works outward to the left and to the right.

Panels 0 (CAP panel) and 1 (drive panel) are allowable entries for an SL8500 audit, however, in this case the HSC displays a condition code 0 because it does not find any storage cells on these panels.

**ROW(row-list)**

optionally, specifies that only certain rows within the specified LSM panel are to be audited. If this parameter is specified, the LSM and PANel parameters may contain only one element.

*row-list* is a single row or a list of rows. Ranges are not allowed. If a list is specified, the elements must be separated by blanks or commas, and the entire list enclosed in parentheses. A *row-list* element is a one or two digit decimal number, and is specific to the LSM type. See "Rows" on page B-3 for a list of valid values.

#### **COLumn(column-list)**

optionally, specifies that only certain columns (cells) within an LSM panel row are to be audited. If this parameter is specified, the LSM, PANel, and ROW parameters may contain only one element.

*column-list* is a single column, or a list of columns. Ranges are not allowed. If a list is specified, the elements must be separated by blanks or commas, and the entire list enclosed in parentheses. A *column-list* element is a one or two digit decimal number and is specific to the LSM type. See "Columns" on page B-4 for a list of valid values.

- A list of rows cannot be specified if a list of columns is specified.
- A list of columns cannot be specified if a list of rows is specified.
- All SL3000 columns are numbered 0-5 (left to right).

#### **CAP(cap-id)**

optionally, specifies a particular Cartridge Access Port to be used for any required cartridge ejections during the Audit operation.

*cap-id* identifies one or more CAPs. The format is *AA:LL:CC*, where *AA:LL* indicates the LSM id, and *CC* indicates the CAP. See "CAP Numbers" on page B-1 for a list of valid values.

If a CAP is not specified, a CAP in the identified ACS is selected based on the CAPPref operator command. See "CAPPref" on page 3-21 for more information.

Multi-ACS audits cannot specify the CAP parameter. A CAP is chosen for each ACS based upon CAP preference values.

## **Parameters (Virtual Volume Audit)**

As shown in Figure 3-4, the AUDit command includes the following parameters for a virtual volume audit:

#### **VTSS(vtss-name or vtss-list)**

optionally, specifies an audit of one or more VTSSs.

*vtss-name* or *vtss-list* indicates the names of one or more VTSSs.

#### **INVLDMIR**

optionally, specifies an audit of MVCs with invalid MIRs.

#### **MVC(volser, vol-range, or vol-list)**

optionally, specifies an audit of one or more MVCs.

*volser*, *vol-range* or *vol-list* indicates the volser, volser range, or volser list of MVC(s) to be processed.

## **Parameters (Virtual Volume Audit)**

As shown in Figure 3-4, the AUDit command includes the following parameters for a virtual volume audit:

#### **VTSS(vtss-name or vtss-list)**

optionally, specifies an audit of one or more VTSSs.



*vtss-name* or *vtss-list* indicate the names of one or more VTSSs to be processed.

### **INVLDMIR**

optionally, specifies an audit of MVCs with invalid MIRs.

### **MVC(volser, vol-range, or vol-list)**

optionally, specifies an audit of one or more MVCs.

*volser*, *vol-range* or *vol-list* indicates the volser, volser range, or volser list of MVC(s) to be processed.

## **Audit Report**

An audit report lists the VTVs and MVCs that are different from those listed in the CDS, as shown in [Example 3-3](#). In this example, 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.

### **Example 3-3 Audit report**

```
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
```

---

---

**Note:** An audit also generates the following:

- 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:

VVVVVVTV possibly corrupt (Block:BBBBBB)

*Explanation:* During the audit, an I/O error occurred for VTV VVVVVV at block BBBBBB on the MVC being audited.

VVVVVVTV not found [ , no MVC copies left ]

*Explanation:* 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 the VTV.

VVVVVVTV not found on CDS (Block:BBBBBB)

*Explanation:* 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 the VTV.

VVVVVVTV not current (Block:BBBBBB)

*Explanation:* 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 the VTV.

VVVVVVTV not current (Block:BBBBBB)

*Explanation:* 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 the VTV.

VVVVVVTV not current (Block:BBBBBB)

*Explanation:* 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 the VTV.

VVVVVVTV Added as secondary copy (Block:BBBBBB)

*Explanation:* 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.

VVVVVVvalid, but environment conflict

*Explanation:* A new VTV has been discovered in the VTSS, but it is not preferred because it was created on a different environment from the one on which the AUDIT

command is running. For example, a Production VTV is discovered when running on a DR Test system, or a DR Test VTV is discovered when running on a Production system.

VVVVVvalid, even though environment conflict

*Explanation:* A new VTV has been discovered in the VTSS, and it is not going to be deleted because even though it was created on a different environment from the one on which the AUDIT command is running. For example, a Production VTV is discovered when running on a DR Test system, or a DR Test VTV is discovered when running on a Production system, the VTV that matches the running environment is either corrupt or in a recovery situation.

VVVVVdiscovered but not adopted due to recovery situation

*Explanation:* A new VTV has been discovered in the VTSS, but it is not preferred because one or other of the VTVs being compared is in a recovery situation and any decision made at this point would be premature before the outcome of the recovering VTV is determined.

VVVVVCorrupt VTV version deleted from VTSS nnnnnnnn

*Explanation:* While comparing VTV copies a corrupt version is encountered in one of the VTSSs and is deleted.

VVVVVDuplicate copy ignored (Block:BBBBBB)

*Explanation:* The audit found a duplicate copy of VTV VVVVVV at block BBBBBB and ignored this copy.

VVVVVLink to old version on MVC MMMMMM removed

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

VVVVVOld VTV version deleted from VTSS SSSSSSSS

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

VVVVVOld version of VTV retained [ VTSS SSSSSSSS ]

*Explanation:* 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.

VVVVVVersion older than MVC copies [ VTSS SSSSSSSS ]

*Explanation:* 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 ]

*Explanation:* 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 ]

*Explanation:* 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 ]

*Explanation:* 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 \*\*\*

*Explanation:* 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

*Explanation:* 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

*Explanation:* 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

*Explanation:* 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

*Explanation:* 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 Oracle StorageTek Software Support.

## BACKup

### Interfaces:

- SLUADMIN utility only
- UUI Support: No

### Subsystem Requirements:

Active HSC not required

---



---

**Note:** Backup to tape is not supported.

---



---

## Description

The BACKup command produces a backup of the library control data set (CDS).

The utility automatically selects the primary CDS copy based on the specified CDS DD statements or the CDS order used by the active HSC on the system.

If you want to back up a single, specific copy of the CDS, you can specify a single DD statement, SLSCNTL, to point to the file you want to back up. In general, however, you should only back up the primary CDS; the HSC RESTORE utility automatically creates the correct number of CDS copies when it executes.

---



---

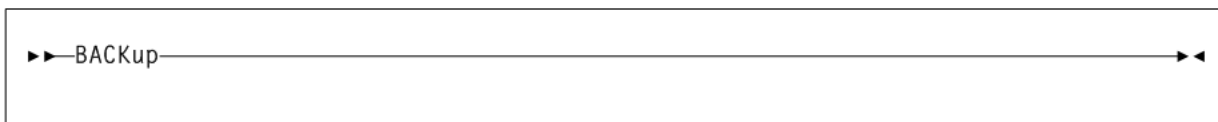
### Note:

- Refer to Oracle's ELS publication *ELS Legacy Interfaces Reference* for information about using BACKup when copies of the CDS are in different locations.
  - See "PITCOPY" on page 3-292 for information about the PITCOPY command as an alternative to backup.
- 
- 

## Syntax

Figure 3-5 shows BACKup command syntax.

**Figure 3-5** BACKup command syntax



## Parameters

None. Refer to Oracle's ELS publication *ELS Legacy Interfaces Reference* for information about using BACKup when copies of the CDS are in different locations.

## Additional JCL Requirements

In addition to the required JCL definition statements described in "Invoking SLUADMIN" on page 5-1, the following definition statements apply to the BACKup JCL:

**SLSBKUP**

specifies the created backup data set. The LRECL and BLKSIZE of this data set default to 4096.

## CANcel

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

## Description

The `CANcel` command stops active and queued processes that use an RTD.

## Syntax

Figure 3–6 shows `CANcel` command syntax.

**Figure 3–6** *CANcel command syntax*



## Parameters

As shown in Figure 3–6, the `CANcel` command includes the following parameters:

**ID(process-id)**

specifies a process to cancel.

*process-id* indicates the process ID.

**Type**

optionally, specifies the type of process to cancel.

**ALL**

Cancel all processes.

**MIGrate**

Cancel all migration processes.

**RECA11**

Cancel all recall processes.

**RECLaim**

Cancel all reclaim processes.

## CAPPref

### Interfaces:

- Console or PARMLIB
- UUI Support: No

### Subsystem Requirements:

Active HSC at FULL service level

## Description

The CAPPref command assigns a preference value to one or more designated CAPs, and places a CAP into automatic or manual mode.

Several HSC commands and utilities require the use of a CAP and give you the option to either specify the CAP to use, or allow the HSC to automatically select a CAP. When you allow the HSC to make the selection, the CAP is chosen based on availability and the CAP preference value.

Assigning CAP preference values establishes an ordered list of CAPs from which the HSC selects an available CAP with the highest nonzero preference value. A CAP's preference value is zero until it is changed by a CAPPref command. Zero preference CAPs are never automatically selected by the HSC; however, they can be explicitly requested by the user.

---



---

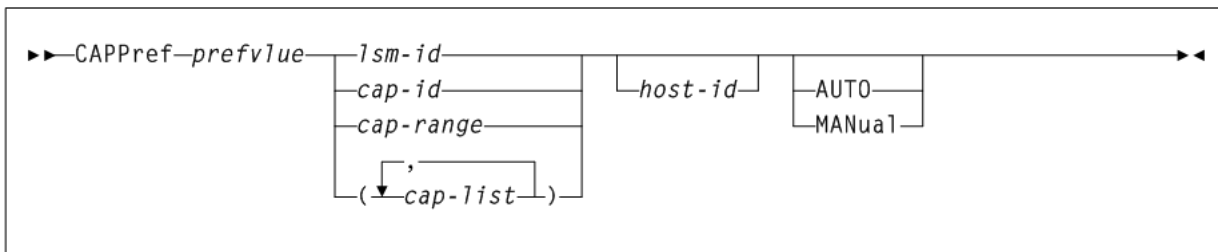
### Note:

- A CAP preference value is only in effect on the host that executes the command.
  - CAP mode (automatic or manual) is in effect on all hosts system-wide for each CAP; it cannot be set discretely for each host.
- 
- 

## Syntax

Figure 3-7 shows CAPPref command syntax.

**Figure 3-7 CAPPref command syntax**



## Parameters

As shown in Figure 3-7, the CAPPref command includes the following parameters:

### preflue

specifies a preference value which is assigned to all listed CAPs. Preference values are decimal in the range from 0-9.

A CAP with a preference value of 9 has the highest priority and is always selected first when it is available. If two or more available CAPs in an ACS have the highest preference value, the one with the lowest CAPid is selected.

The preference value for a PCAP must be zero. A PCAP is only used when explicitly requested by the user; the HSC does not automatically select a PCAP.

To use the CAPPref command to place a PCAP into `AUTO` or `MANual` mode, specify a `prefvlue` of 0.

**lsm-id**

identifies an LSM. The format of an *lsm-id* is *AA:LL*, where *AA* indicates the ACSid (decimal 00-99) and *LL* indicates the LSM number (decimal 00-99).

---

---

**Note:** If you enter this parameter when there is more than one CAP in the specified LSM, an error message is generated.

---

---

**cap-id, cap-range, or cap-list**

identifies one or more CAPs. The format of a *cap-id* is *AA:LL:CC*, where *AA:LL* indicates the LSMid, and *CC* indicates the CAP. See "CAPid" on page B-1 for a list of valid values.

---

---

**Note:**

If SL3000 library CAPs are not present, the HSC reports them as "not installed." This keeps CAP addresses from changing as new CAPs are added.

SL3000 and SL8500 libraries do not contain a PCAP.

---

---

A *cap-range* identifies an inclusive range of multi-cell CAPs; PCAPs are excluded from the range. The beginning and ending values in a range must be valid CAPids and neither CAPid can specify a PCAP. The rules for specifying ranges apply.

PCAPs must be explicitly specified in the CAPPref command.

Example 1:

```
00:00:00-00:03:00
```

In this example, all CAPs in LSMs 00:00, 00:01, 00:02, and 00:03, except PCAPs, are included in this range.

Example 2:

```
00:00:00-00:03:01
```

In this example, CAPid 00:00:00 is either a standard CAP or a magazine-style CAP. CAPid 00:03:01 is a magazine-style CAP. All CAPs in LSMs 00:01 and 00:02, except PCAPs, are included in this range.

Each *cap-list* element can be either a single CAPid or a CAPid range. The elements in a list must be separated by a comma or a blank, and the entire list must be enclosed in parentheses.

**host-id**

optionally, specifies that the CAPPref command is to be performed only if the *host-id* parameter matches the identifier of the issuing host (the SMF system identifier for both JES2 and JES3).



- If `CAPPref` is issued from `PARMLIB` and a `host-id` is specified, the command is executed only by the host with the matching ID.
- If `CAPPref` is issued from `PARMLIB` and a `host-id` is not specified, the command is executed by each host that accesses `PARMLIB`.

Refer to Oracle's ELS publication *Managing HSC and VTCS* for more information about `PARMLIB`.

### **AUTO**

optionally, indicates that the specified CAPs are to be placed in automatic mode (referred to as auto-mode). Auto-mode CAPs remain unlocked when not in use. `AUTO` is the initial setting for a PCAP.

When a CAP is in auto-mode, you can initiate an enter operation without issuing an `ENTER` command. You do this by opening the CAP door, placing one or more cartridges inside, and closing the CAP. The CAP is locked while the enter is being processed; then it is unlocked again.

Auto-mode CAPs can be used for eject processing by either:

- specifying the `CAPid` on an `EJECT` command or `EJECT` utility, or
- assigning a preference value and allowing the HSC to automatically select the CAP

A CAP in auto-mode is locked whenever it is selected for an eject operation, to prevent an enter operation on this CAP until the eject completes.

---



---

**Note:** HSC termination will be slower with CAPs in auto-mode; especially in a multiple-host environment.

---



---

### **MANual**

optionally, indicates that the specified CAP is to be locked when not in use. `MANual` is the initial setting for all multi-cartridge CAPs.

`AUTO` or `MANual` settings are sent host-to-host using LMU broadcast and the status is preserved in the control data set. For this reason, it is recommended that you not make frequent changes to the CAP mode.

A `prefvlu` of 0 must be specified when using the `CAPPref` command to set a PCAP to `AUTO` or `MANual` mode.

## **CDs**

### **Interfaces:**

- Console or `PARMLIB`
- UUI Support: No

### **Subsystem Requirements:**

Active HSC at `BASE` or `FULL` service level

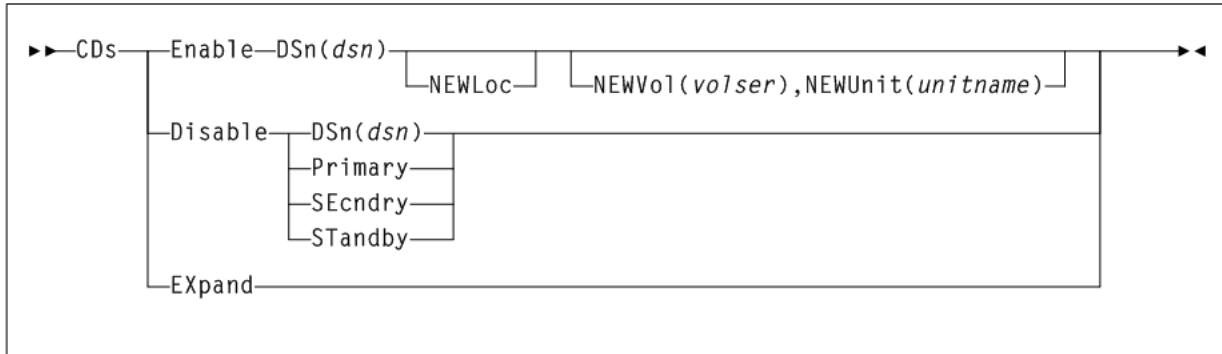
## **Description**

The `CDs` command enables, disables, relocates, or expands a control data set on an active HSC (or, in a multiple-host environment, on all active hosts) without requiring you to stop or substantially disrupt normal tape processes.

## Syntax

Figure 3–8 shows CDs command syntax.

**Figure 3–8** CDs command syntax



## Parameters

As shown in Figure 3–8, the CDs command includes the following parameters:

### Enable

allocates and activates any preexisting, renamed, or new CDS specified by the *DSn* parameter. The CDs *Enable* command can allocate and activate a CDS residing in a new location, either with a preexisting or a new data set name, when the *NEWLoc* parameter is specified. MVS catalog services are used to resolve the volume and unit definitions, if the *NEWVol* or *NEWUnit* parameters are not specified.

The specified control data set must be named in the CDSDEF control statement in PARMLIB.

### DSn(dsn)

specifies the control data set name to allocate for all hosts. *dsn* indicates the data set name. This parameter is required.

### NEWLoc

optionally, indicates that a relocation activity has occurred for the CDS named by the *DSn* parameter. MVS uses the catalog facility to determine the volume location of the relocated CDS.

### NEWVol(volser)

optionally, specifies the volume for a relocated CDS copy. In MVS if the new CDS copy is not cataloged, this parameter is required. If *NEWVol* is specified, but *NEWUnit* is not, *NEWUnit* defaults to *SYSALLDA*.

*volser* indicates the volume.

*NEWVol* is required if the user is executing the command in the HSC VM environment.

### NEWUnit(unit-name)

optionally, specifies the unit name for a relocated CDS copy. This parameter defaults to *SYSALLDA* if it is omitted and *NEWVol* is specified.

*unitname* indicates the unit name.

*NEWUnit* is required if the user is executing the command in the HSC VM environment.

**Disable**

unallocates (makes inactive) the specified CDS. The `CDS Disable` command does not disable the last active copy of the control data set.

**DSN(dsn)**

optionally, specifies the control data set name to unallocate for all hosts. *dsn* indicates the data set name.

**Primary**

optionally, indicates that the current primary control data set is to be disabled.

**SEcndry**

optionally, indicates that the current secondary control data set is to be disabled.

**STandby**

optionally, indicates that the current standby control data set is to be disabled.

**EXpand**

expands all enabled CDSs to the maximum number of 4096 blocks that can fit in the physical space allocated for the CDS. The maximum number of 4096 blocks is determined by the smallest CDS copy.

## CDSCREat

**Interfaces:**

- SLUADMIN utility only
- UUI Support: Yes

**Subsystem Requirements:**

None.

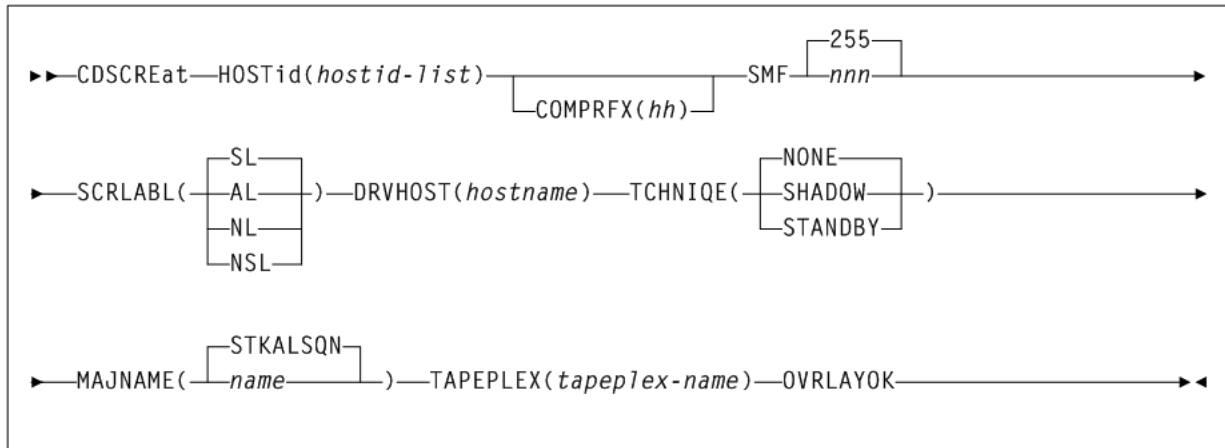
## Description

The `CDSCREat` utility uses a small set of parameters to define the CDS for a tapeless configuration. A tapeless configuration may include VLE hardware but may not include any defined ACSs or real tape drives.

## Syntax

Figure 3–9 shows `CDSCREat` command syntax.

Figure 3–9 CDSCREAt command syntax



## Parameters

As shown in [Figure 3–9](#), the CDSCREAt command includes the following parameters:

### HOSTid(hostid-list)

specifies the list of SMF host IDs on which this system can be executed. This parameter is required.

*hostid-list* indicates one or more host ids. You can specify a maximum of 16 host IDs.

### COMPREFX(hh)

optionally, specifies the two-character hexadecimal representation of the command prefix to be used to enter commands to the subsystem. If this parameter is omitted or is set to a null value (40), then you must use the MVS MODIFY command to direct requests to the subsystem, for example, F HSC0,D CDS. See [Table 3–14](#) under the SET COMPREFX command for valid values.

*hh* indicates the two-character hexadecimal value.

### SMF(nnn)

optionally, specifies the SMF ID of statistics data that will be written to the SMF data set.

*nnn* indicates the SMF ID. The value must be between 128 and 255.

If this parameter is omitted, the value is set to 255.

### SCRLABL

optionally, specifies the label type used as the scratch default. Specify one of the following:

- SL indicates standard-labeled tape. This is the default if the parameter is omitted.
- AL indicates ANSI-labeled tape.
- NL indicates nonlabeled tape.
- NSL indicates nonstandard labeled tape.

**DRVHOST(hostname)**

optionally, specifies the host ID that will be used as the "master" value for drive addresses. If drive addresses are different for different hosts, a DRVHOST should be set to allow SMC to map client drive addresses to fixed server drive addresses.

*host-name* indicates the "master" host id.

**TCHNIQE**

optionally, specifies the recovery technique (number of CDS copies) that will be created and used by the system.

- NONE specifies one CDS copy. This is the default if the parameter is omitted.
- SHADOW specifies two CDS copies.
- STANDBY specifies three CDS copies.

If this parameter is omitted, the default value is NONE. The CDSCREAT utility validates that the specified (or implied) technique matches the number of CDS DD statements specified.

**MAJNAME(name)**

optionally, specifies an ENQ/DEQ/RESERVE QNAME. *name* indicates the a 1-8 character QNAME.

If this parameter is omitted, the default value is STKALSQN.

**TAPEPLEX(tapeplex-name)**

optionally, specifies a TapePlex name for the HSC/VTCS system. *tapeplex-name* is a 1-8 character TapePlex name. This must match the SMC TAPEPLEX name.

If this parameter is omitted, it is set automatically when SMC initially communicates with HSC/VTCS.

**OVLAYOK**

optionally, specifies that the input CDS file(s) may be existing CDS data sets, and that they may be overlaid by the create process. If this parameter is omitted, and any CDS file is an existing CDS file, the CDSCREAT process fails.

## Additional JCL Requirements

Execution of this utility requires specification of a DD statement for each CDS copy indicated by the TCHNIQE parameter (or exactly one DD statement if the TCHNIQE parameter is omitted).

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statements apply to the CDSCREAT JCL:

**SLSCTL**

specifies the primary CDS.

**SLSCTL2**

optionally, specifies the secondary CDS.

If a secondary CDS exists, include this statement so that if a switch occurs and the secondary CDS becomes active, the CDSs can be reordered to maintain database integrity.

**SLSSTBY**

optionally, specifies the standby CDS.

## CDSData

### Interfaces:

- Utility only
- UUI Support: Yes

### Subsystem Requirements:

Active HSC not required

## Description

The CDSData command produces XML output data for library configuration and volume and pool definitions from VOLPARM control statements.

---



---

**Note:** No text output is produced by this command, so the SLSXML DD statement must be included in the SLUADMIN execution to create output from this command. See ["Output Data Set Definition \(DD\) Statements"](#) on page 4-7 for more information.

---

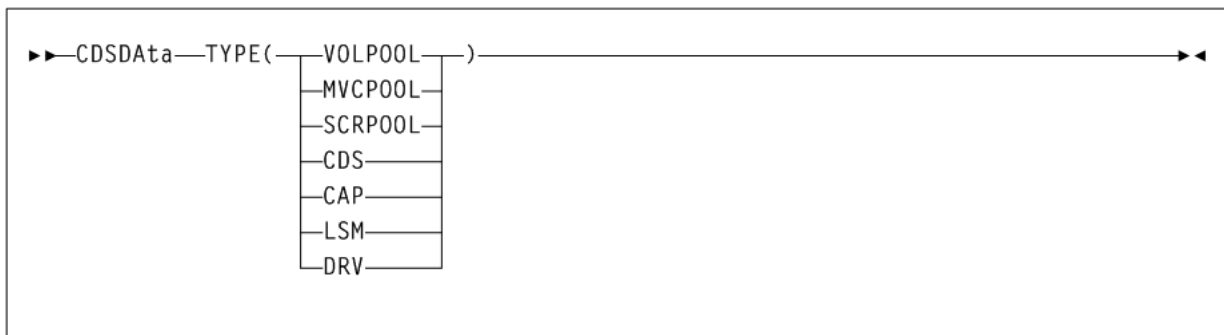


---

## Syntax

Figure 3–10 shows CDSData command syntax.

**Figure 3–10 CDSData command syntax**



## Parameters

As shown in [Figure 3–10](#), the CDSData command includes the following parameters:

### TYPE

specifies the type of CDS data for which output is produced. The TYPE Keyword is positional and therefore optional. One of the listed types must be specified.

#### **VOLPOOL**

produces XML data for VOLPARM VOLUME pool definitions.

#### **MVCPOOL**

produces XML data for VOLPARM MVC pool definitions.

#### **SCRPOOL**

produces XML data for VOLPARM SCRATCH pool definitions.

**CDS**

produces XML CDS data.

**CAP**

produces XML CAP data.

**LSM**

produces XML LSM data.

**DRV**

produces XML drive data.

**TYPE**

specifies the type of CDS data for which output is produced. The `TYPE` Keyword is positional and therefore optional. One of the listed types must be specified.

- `VOLPOOL` produces XML data for `VOLPARM VOLUME` pool definitions.
- `MVCPool` produces XML data for `VOLPARM MVC` pool definitions.
- `SCRPOOL` produces XML data for `VOLPARM SCRATCH` pool definitions.
- `CDS` produces XML CDS data.
- `CAP` produces XML CAP data.
- `LSM` produces XML LSM data.
- `DRV` produces XML drive data.

## CDSDEF

**Interfaces:**

- `PARMLIB` only
- `UI Support`: No

**Subsystem Requirements:**

None.

## Description

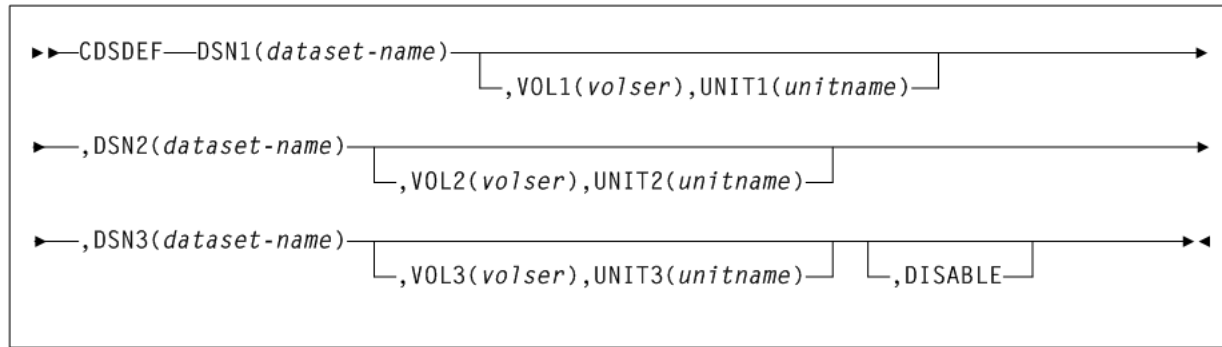
The `CDSDEF` command specifies control data set names, locations, and number of control data sets that you want running. This information is used by HSC during initialization.

`CDSDEF` is a required command.

## Syntax

Figure 3–11 shows `CDSDEF` command syntax.

Figure 3–11 CDSDEF command syntax



## Parameters

As shown in Figure 3–11, the CDSDEF command includes the following parameters:

### DSN1(*dataset-name*)

specifies the primary CDS. *dataset-name* indicates the name of an HSC control data set.

At least one DSN parameter must be specified, as many as three can be specified. If two data sets are specified, then the volume inventory information on both copies is kept current, and these two data sets are referred to as the primary and the secondary CDS.

If all three are specified, then two copies are kept current and the third data set is a standby CDS which is enabled as the default condition. This standby CDS can be disabled with the `DISABLE` parameter. In the case where a standby copy of the CDS is being used, you do not need to specify which two of the three CDSs are the primary and secondary copies. The HSC determines, based on last use, which two are the correct copies to use during the initialization of the Control Data Set Services.

### ,VOL1(*volser*)

optionally, specifies a volume serial number of the DASD volume where the specified CDS resides. If the data set is not cataloged, this must be specified.

If omitted, volume and unit information is determined from the MVS catalog, and the `VOL1` parameter is ignored.

### ,UNIT1(*unit-name*)

optionally, specifies the unit parameter in the `SVC99` dynamic allocation parameter list. If omitted, `SYSALLDA` is used.

*unit-name* can be an IBM-defined generic (for example, 3390), a system-built esoteric (for example, `SYSALLDA`), a user-defined esoteric, or a specific device number. It is recommended that a general value of `SYSALLDA` be specified (or defaulted) to allow maximum flexibility and to reduce the need for `CDSDEF` updates to keep them consistent with the active CDS.

### DSN2(*dataset-name*)

optionally, specifies a secondary CDS. *dataset-name* indicates the name of an HSC control data set.

### ,VOL2(*volser*)

optionally, specifies a volume serial number of the DASD volume where the secondary CDS resides. If the data set is not cataloged, this must be specified.



**,UNIT2(unit-name)**

optionally, specifies the unit parameter in the SVC99 dynamic allocation parameter list. If omitted, *SYSALLDA* is used.

*unit-name* can be an IBM-defined generic (for example, 3390), a system-built esoteric (for example, *SYSALLDA*), a user-defined esoteric, or a specific device number. It is recommended that a general value of *SYSALLDA* be specified (or defaulted) to allow maximum flexibility and to reduce the need for *CDSDEF* updates to keep them consistent with the active CDS.

**DSN3(dataset-name)**

optionally, specifies a standby CDS. *dataset-name* indicates the name of an HSC control data set.

**,VOL3(volser)**

optionally, specifies a volume serial number of the DASD volume where the standby CDS resides. If the data set is not cataloged, this must be specified.

**,UNIT3(unit-name)**

optionally, specifies the unit parameter in the SVC99 dynamic allocation parameter list. If omitted, *SYSALLDA* is used.

*unit-name* can be an IBM-defined generic (for example, 3390), a system-built esoteric (for example, *SYSALLDA*), a user-defined esoteric, or a specific device number. It is recommended that a general value of *SYSALLDA* be specified (or defaulted) to allow maximum flexibility and to reduce the need for *CDSDEF* updates to keep them consistent with the active CDS.

**,DISABLE**

optionally, disables the standby CDS. If this parameter is not specified, the default is to enable the standby CDS at HSC initialization.

If *DISABLE* is specified, initialization verifies the standby CDS by allocating, opening, and reading the data set. The standby CDS is then closed and deallocated. This process enables manual control over the use of the standby CDS during a control data set switch situation.

If the standby CDS is enabled, the HSC automatically uses it for switching. If it is disabled, it is not used. There is no enable parameter for *CDSDEF*, as the enable condition is the default.

## CLean

**Interfaces:**

- Console or *PARMLIB*
- UUI Support: No

**Subsystem Requirements:**

Active HSC at *FULL* service level

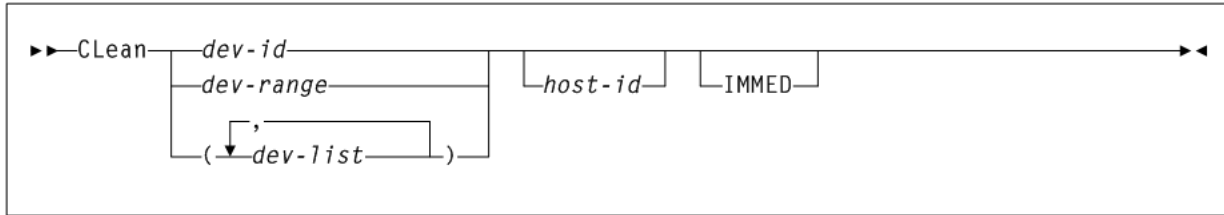
## Description

The *CLean* command schedules a cleaning cartridge to be mounted on a library-controlled transport. The specified transport is flagged by the control data set and a cleaning cartridge is mounted prior to the next volume mount.

## Syntax

Figure 3–12 shows CLean command syntax.

Figure 3–12 CLean command syntax



## Parameters

As shown in Figure 3–12, the CLean command includes the following parameters:

### dev-id, dev-range, or dev-list

specifies a device address, range of device addresses, or list of addresses for which cleaning is to be scheduled. Each *dev-list* element can be either a single device address or a range. The elements in a list must be separated by a comma or a blank, and the entire list must be enclosed in parentheses.

### host-id

optionally, indicates that cleaning is to be performed for the device address of the specified host (the SMF system identifier for both JES2 and JES3).

### IMMED

optionally, specifies that an immediate clean of the device is to be performed if the drive is idle. This occurs regardless of the MNTD *Autocln* setting. If the device is not idle then the device is scheduled to be cleaned after the dismount of the volume that is currently mounted.

## COMMPath

### Interfaces:

- Console or PARMLIB only
- UI Support: No

### Subsystem Requirements:

Active HSC at BASE or FULL service level

## Description

The COMMPath command is used in a multiple-host environment to do the following:

- Set up a tiered communication service between HSC hosts which can be modified at any time without bringing down the HSC.
- Define the hierarchy of methods available to each host for communication with other HSC hosts in the TapePlex.
- Establish the current method of communication for an HSC host (or switch from one communication method to another).

In a single host environment, the COMMPATH command has no effect.

It is recommended that the COMMPATH command which defines the host-to-host communications hierarchy be defined in PARMLIB at startup. After startup, issue the COMMPATH command from the console to switch or delete communication paths.

The hierarchy of available HSC host-to-host communication methods listed in order, according to burden on HSC performance, are:

1. TCP (least burden)
2. VTAM
3. LMU
4. CDS (greatest burden)

TCP and VTAM host-to-host communications are strongly recommended. LMU host-to-host communications method places a greater burden on HSC performance because it competes with library hardware communications. CDS host-to-host communications places the greatest burden on HSC performance because it requires access to the shared CDS.

The COMMPATH command updates an in-memory table of definitions for the HSC host that executes the command. This table must contain a list of paths for this host and separate lists for each additional HSC host defined. Therefore, on each HSC host, you must issue the COMMPATH command once to define the path itself, once to define the path(s) for a second host, and so on. For example, in a system with hosts A, B, and C:

```
COMMPATH commands in host A startup PARMLIB:
  COMMPATH HOST(A) VTAMPATH(APPLID1) LMU(00) METHOD(VTAM)
  COMMPATH HOST(B) VTAMPATH(ALLLID2) LMU(00) METHOD(VTAM)
  COMMPATH HOST(C) VTAMPATH(APPLID3) LMU(00) METHOD(VTAM)

COMMPATH commands in host B startup PARMLIB:
  COMMPATH HOST(A) VTAMPATH(APPLID1) LMU(00) METHOD(VTAM)
  COMMPATH HOST(B) VTAMPATH(ALLLID2) LMU(00) METHOD(VTAM)
  COMMPATH HOST(C) VTAMPATH(APPLID3) LMU(00) METHOD(VTAM)

COMMPATH commands in host C startup PARMLIB:
  COMMPATH HOST(A) VTAMPATH(APPLID1) LMU(00) METHOD(VTAM)
  COMMPATH HOST(B) VTAMPATH(ALLLID2) LMU(00) METHOD(VTAM)
  COMMPATH HOST(C) VTAMPATH(APPLID3) LMU(00) METHOD(VTAM)
```

It is recommended that path methods mirror each other. However, in the above example, host C could set its method to LMU while host A and B continue to communicate using VTAM.

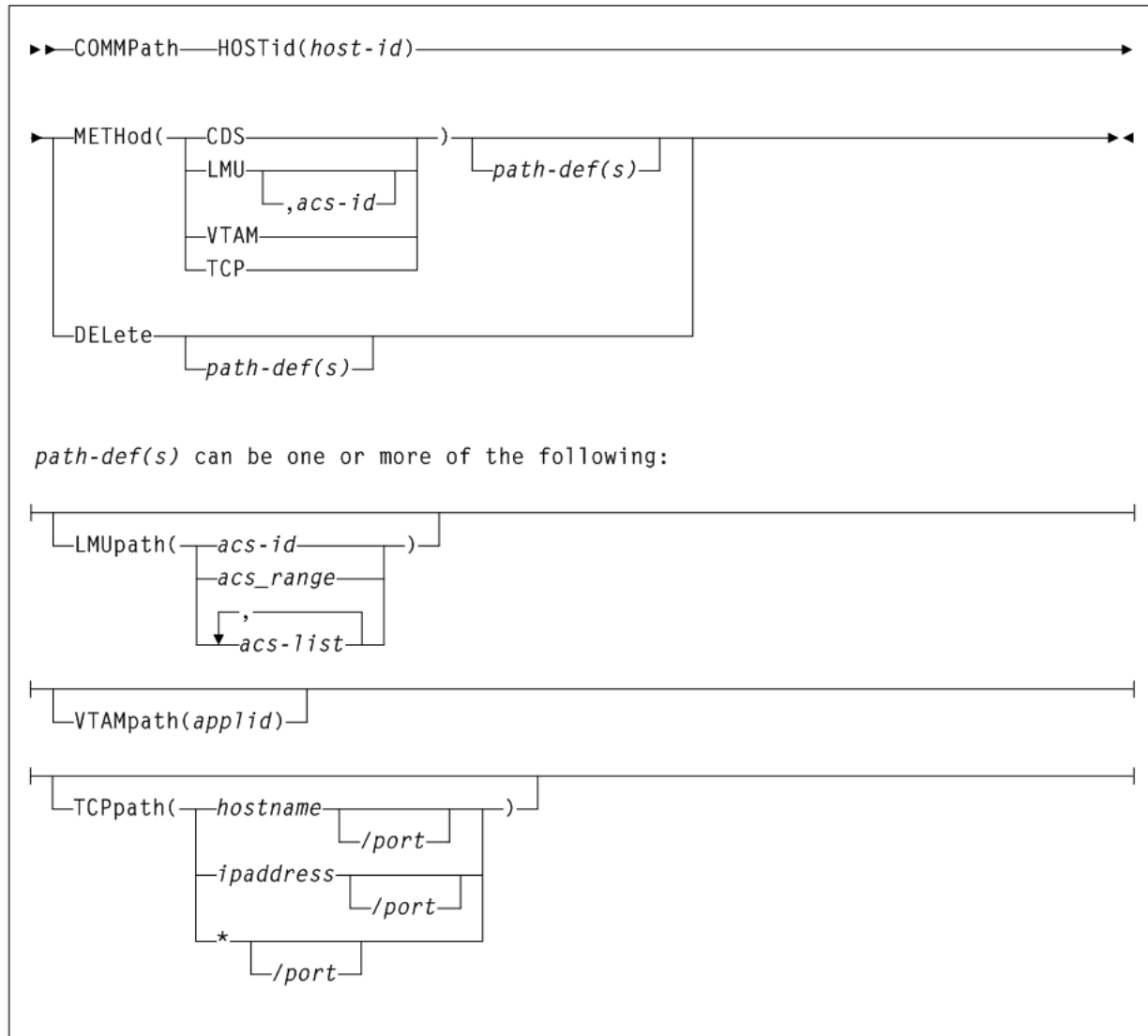
If an error occurs during a host-to-host communication, the HSC performs an automatic downward switch to the next lower defined communication method. After this switch is performed, an upward switch can only be performed using the COMMPATH command.

If a COMMPATH command is not defined for each host in a multiple-host environment, then the missing host's communication method defaults to CDS, which may impose a performance burden on all HSC hosts sharing the CDS. However, if the missing host is infrequently active (for example, a test system) then the additional performance burden would be minimal.

## Syntax

Figure 3–13 shows COMMPATH command syntax.

**Figure 3–13 COMMPath command syntax**



## Parameters

As shown in [Figure 3–13](#), the COMMPATH command includes the following parameters:

### HOSTid(host-id)

specifies the host for which the command sets or modifies the parameters. This parameter is required for all COMMPATH commands.

*host-id* indicates the host identifier (the SMF system identifier for both JES2 and JES3).

### METHOD

optionally, specifies the current method of communications to be used for the specified host. This parameter enables you to switch to a higher, lower, or equivalent (for LMUPATH only) level method of communications.

### CDS

indicates that communication is to be through the control data set. CDS is the initial setting for HSC host-to-host communications.

**LMU**

indicates that communication is to be through an LMU.

*acs-id* indicates the LMU<sub>path</sub> definition that is to be the active (or current) LMU<sub>path</sub>. If you do not supply an *acs-id*, the first LMU<sub>path</sub> specified in the list of LMU<sub>path</sub> definitions is made active.

If LMU is specified, the LMU<sub>path</sub> parameter must have been specified in a previous command, or it must be included in the same command.

If the LMU method is specified in PARMLIB, the switch from CDS to LMU is delayed until the HSC initializes to the full service level.

If the HSC service level drops to base, LMU communications are switched to CDS. When the full service level is restored, an upward switch to the LMU method requires you to issue another COMMPath command.

**VTAM**

indicates that communication is to be through a VTAM network.

If VTAM is specified, the VTAM<sub>path</sub> parameter must have been specified in a previous command, or it must be included in the same command.

A downward switch from the VTAM method is performed when the VTAM APPLID of the HSC is inactivated, or VTAM is shut down.

**TCP**

indicates that communications is to be through a TCP/IP network.

A downward switch from the TCP method is performed when a TCP/IP error occurs, or TCP/IP is shut down.

**LMU<sub>path</sub>(*acs-id*, *acs-range*, or *acs-list*)**

specifies one or more LMU paths that can be used for host-to-host communications. An LMU is eligible to be used for host-to-host communications if it is connected to other HSCs sharing a library.

*acs-id* or *acs-range* or *acs-list* specifies one or more specific LMUs. Each *acs-list* element can be either a single ACSid or an ACSid range. The elements in a list must be separated by a comma or a blank, and the entire list must be enclosed in parentheses.

If more than one eligible LMU is specified, the HSC begins searching for an available communications path beginning with the first *acs-id* in the list. The search continues until a path can be established or the end of the list is reached.

The LMU<sub>path</sub> definitions are cumulative; that is, the definitions specified in a command are appended to the current list of definitions. You can insert an LMU path into the current list by specifying the LMU path that you wish to insert, followed by the LMU paths that are to come after it. If a COMMPath command specifies an LMU path that is already defined in the list, the path is removed from its current position and appended to the end of the list as specified in the command.

---



---

**Note:** For a temporary LMU outage, it may be desirable to delete the LMU's path from the list. When the LMU becomes available again, it may be re-added.

---



---

**VTAMpath(applid)**

optionally, defines the VTAM application to be associated with an HSC host. The specified application name is used in the process of establishing a VTAM connection with other HSCs.

*applid* indicates the name defined by the VTAM APPL statement for the HSC application.

**TCPpath(hostname, ipaddress, or /port)**

optionally, defines the host name and port number combination, or IP address (IPv4 or IPv6) and port number combination associated with an HSC host. The specified hostname is resolved into an IPv4 or IPv6 address. The combined IP address and port number combination is used to start a host-to-host TCPPATH server on the associated HSC host.

*hostname* must be a resolvable host name.

*ipaddress* must be valid IPv4 or IPv6 format address. If you specify "\*", the IP address of the current host (the issuer of the command) is used.

*port* must be a number between 1 and 65535 and be separated from the hostname, IP address, or "\*" by a "/" character with no intervening blanks. If no "/" is found, then the default port number of 58080 is used.

Specify the port number so that it does not conflict with TCP or UDP well known port numbers, or with other applications on this host. When a TCPPATH server is started on this host, it will listen on the specified port number for host-to-host communications from other HSC hosts.

The following are examples of TCPpath definitions:

- TCPPATH(10.135.73.101/50090)
- TCPPATH(PRODMVS/50090)
- TCPPATH(10.135.72.102)
- TCPPATH(\* /50020)
- TCPPATH(\*)

**DELeTe**

optionally, specifies to delete the definitions specified by either or all of the LMUpath, VTAMpath, or TCPpath parameters associated with the specified HSC host.

- If LMU is not the current method, you can delete all LMUpath definitions by specifying the LMUpath keyword without any values.
- If the current method is LMU, the LMUpath cannot be deleted.
- If the current method is VTAM, the VTAMpath definition cannot be deleted.
- If the current method is TCP, the TCPpath definition cannot be deleted.

**Note:**

- Specifying `COMMPATH HOSTID(id) METHOD(TCP)` on the current host starts the `TCPPATH` server for that host.
- Specifying `COMMPATH HOST(id) TCPPATH DELETE` on the current host stops the `TCPPATH` server for that host, as well as deletes the `TCPPATH` definition.
- One host cannot start or stop the `TCPPATH` server on another host.

## CONFIg

**Interfaces:**

- Utility only
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC not required, and must be down on all hosts when running `CONFIG RESET`.

## Description

The `CONFIg` command defines or modifies the VSM configuration stored in the HSC CDS. You create a single file that contains the `CONFIg` command and its input statements.

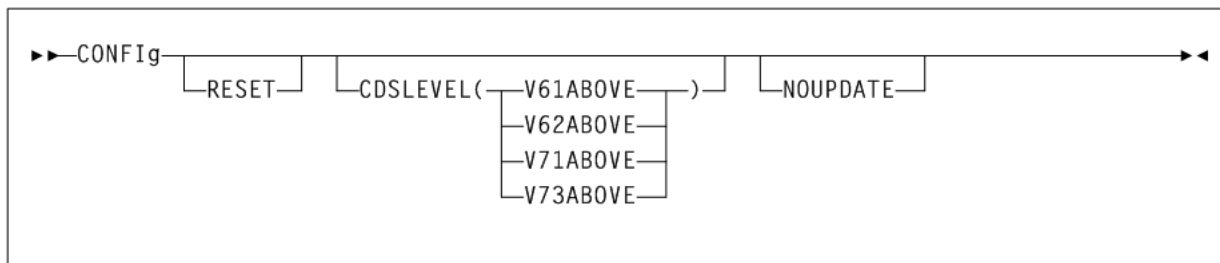
**Note:**

- The `CONFIg GLOBAL` statement specifies VTCS global values. This statement is required. See "[CONFIg GLOBAL Statement](#)" on page 3-42.
- Dynamic reclaim requires an H level CDS.
- Support for 32 GB VTVs requires an I level CDS.

## Syntax

Figure 3-14 shows `CONFIg` command syntax.

**Figure 3-14** *CONFIg command syntax*



## Parameters

As shown in [Figure 3-14](#), the `CONFIg` command includes the following parameters:

**RESET**

optionally, resets the VSM configuration in the CDS.

Specify `RESET` when:

- Changing the CDS to a lower level.
- Adding `LOCKSTR=structure-name` (through `CONFIg GLOBAL`) to a CDS that did not previously use `LOCKSTR=structure-name`.
- Removing `LOCKSTR=structure-name` (through `CONFIg GLOBAL`) from a CDS.
- Removing 32 RTD support (i.e. move from a CDS that supported 32 RTDs to one that doesn't).

Do not specify `RESET` when:

- Changing the CDS from F to G, G to H, or H to I.
- Changing lock structure names (for example, going from `LOCKSTR=VTCSL1` to `LOCKSTR=VTCSL2`).
- Changing 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`.
- Changing 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 take effect when you restart HSC.

Although some changes can be done dynamically by running `CONFIg` and not restarting HSC/VTCS, there may be additional parameter changes required to use any additional resources.

If you run a `CONFIg` with `RESET`, then the status flags are reset and message `SL$6746E` disappears. If a VTSS was offline when you ran the `CONFIg` (with `RESET`), you must remember to audit the VTSS.

---

---

**CDSLEVEL**

optionally, specifies one of the following CDS levels:

**V61ABOVE**

creates an "F" level CDS, which provides the following enhancements:

- Full VSM4 Support
- 4 MVC copies
- 800Mb VTVs
- Near Continuous Operations (NCO)
- Bi-directional clustering
- Improved CDS I/O performance - reduces I/O required to manage virtual scratch subpools

**V62ABOVE**

creates a "G" level CDS, which provides the following enhancements:

- 400Mb/800Mb/2Gb/4gb VTVs



- Standard/Large VTV Pages
- 65000 VTVs per MVC

**V71ABOVE**

creates an "H" level CDS, which provides the following enhancements:

- Dynamic reclaim
- Autonomous device support
- Additional VLE RTDs

**V73ABOVE**

creates an "I" level CDS, which provides 32 GB VTV support.

---

**Note:** 32 GB support requires VSM 6 microcode level 6.2 and VLE microcode level 1.5.1 (if VLE is included in the configuration)

---

The update to a 'I' level CDS may be run non-disruptively with 7.3 hosts active.

Downgrading from an 'I' level CDS to an 'H' or 'G' level CDS is permitted. However, VTVs that have been written in a 32 GB format will be reported and must be deleted. To do this, change the management class to one with the delete-on-scratch attribute and then scratch the VTV.

**NOUPDATE**

optionally, specifies that the configuration is checked and any appropriate messages output, but no updates are made to the CDS.

**CONFIg CLINK Statement**

The CONFIg CLINK statement defines the channel interface between two VTSSs in a Cluster and enables you to define CLINKs on IP interfaces to support replication of VTVs over IP CLINKs.

The VTSSs can be in one of three modes:

- Primary-Secondary, in which there are two VTSSs and you write CLINK statements for only the Primary.
- Peer-to-Peer, in which case there are two VTSSs and you write CLINK statements for both VTSSs to enable bi-directional VTV replication.
- Any-to-Any, in which there are more than two VTSSs in the cluster and the individual relationships are defined through CLINK statements.

In addition, CLINKs can be defined from the VTSS to perform electronic export to another TapePlex. These links need not be to the same VTSS within the remote TapePlex.

Each VTSS can either be connected to a maximum of three different TapePlexes, or two TapePlexes and a partner VTSS in a cluster.

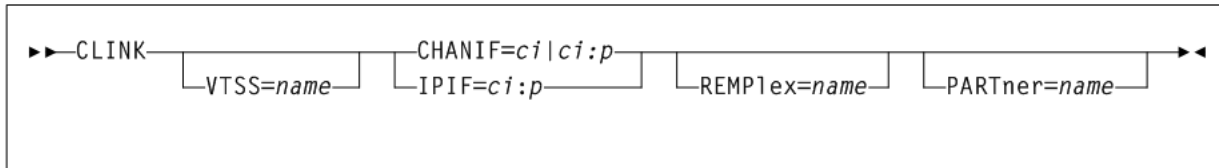
An electronic export connection can always be to multiple VTSSs. A cluster connection can be to multiple VTSSs only if the CONFIg CLUSTER statement specifies more than two VTSS members.

Refer to Oracle's ELS publication *ELS Disaster Recovery and Offsite Data Management Guide* for examples.

## Syntax

Figure 3–15 shows CONFIg CLINK statement syntax.

Figure 3–15 CONFIg CLINK statement syntax



## Parameters

As shown in Figure 3–15, the CONFIg CLINK statement includes the following parameters:

### VTSS=name

optionally, specifies the name of the VTCS that owns the CLINK. This is the VTSS from which the VTVs are replicated or exported.

If this parameter is omitted, by default, the CLINK is attached to the current VTSS being defined.

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
```

### CHANIF=ci or ci:p

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

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

For a VSM 5, this value must match the actual FICON interface values. For a VSM 6, this must be unique for each VTSS, and does not correspond to an actual value on the VSM 6 FICON ports.

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*).

Refer to Oracle’s ELS publication *Configuring HSC and VTCS* for more information.

### IPIF=ci:p

optionally, enables Ethernet connection of two Native IP (IFF3 card) ports, where the *ci:p* values are shown in parentheses before each Target IP Address on the IFF IP Configuration Status screen for each IFF ethernet port. Valid values for *c:ip* are:

- *c* indicates the VTSS Storage Cluster number (0 or 1).

- *i* indicates the interface number (A or I)
- *p* indicates the device number on the interface (0, 1, 2, or 3).

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

Refer to Oracle's ELS publication *Configuring HSC and VTCS* for more information.

---

**Note:** The `CLINK` statement must contain either the `CHANIF` or `IPIF` parameter, but not both.

---

#### **REMPlex=name**

optionally, specifies the name of the TapePlex at the receiving end of the `CLINK` that receives the copies of the VTVs. During validation of the link, this name is compared with the `THISPLEX` definition in the remote system configuration.

The `REMPlex` parameter indicates that the `CLINK` is to be used for electronic export. Absence of this parameter indicates that the `CLINK` is to be used for a cluster connection.

#### **PARTner=name**

optionally, specifies the name of the VTSS at the receiving end of the `CLINK` that receives copies of the VTVs.

This parameter is required for `CLINKs` used for electronic export, or where the cluster consists of more than two VTSSs.

During validation of the link, this name is compared with the VTSS name at the other end of the `CLINK`.

## CONFIg CLUSTER Statement

The `CONFIg CLUSTER` statement defines the VTSSs in a Cluster.

### Syntax

Figure 3–16 shows `CONFIg CLUSTER` statement syntax.

**Figure 3–16** *CONFIg CLUSTER statement syntax*

```
▶▶—CLUSTER—NAME=nnnnnnnn—VTSSs(vtss-list)————▶▶
```

### Parameters

As shown in Figure 3–16, the `CONFIg CLUSTER` statement includes the following parameters:

#### **NAME=nnnnnnnn**

specifies the 1 to 8 character identifier of the Cluster.

This parameter is required. There is no default value.

**VTSSs(vtss-list)**

Specifies the names of the VTSSs in a cluster. At least two VTSSs must be specified. If more than two VTSSs are specified, then down level hosts will not function.

The relationship between each of the VTSSs is defined by the CONFIg CLINK statement. See "[CONFIg CLINK Statement](#)" on page 3-39 for more information.

This parameter is required. There is no default value.

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
```

The DEComp utility outputs CLUSTER statements with blanks instead of commas as separators, which is also valid input to CONFIg. For example, DEComp outputs the following, which is valid input to CONFIg:

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

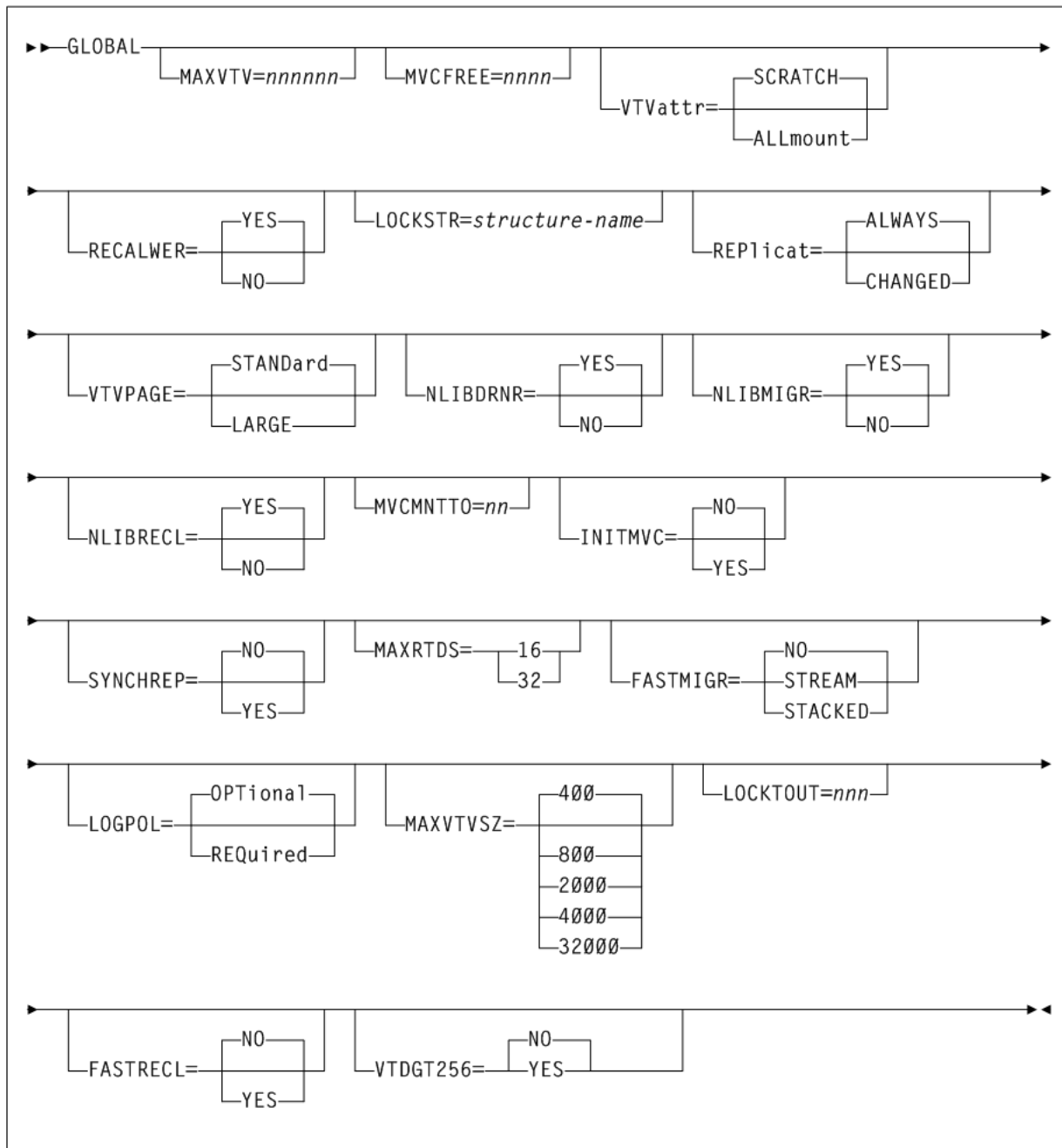
## CONFIg GLOBAL Statement

The CONFIg GLOBAL statement specifies VTCS global values. This statement is required.

**Syntax**

[Figure 3-17](#) shows CONFIg GLOBAL statement syntax.

Figure 3–17 CONFlg GLOBAL statement syntax



### Parameters

As shown in Figure 3–17, the CONFlg GLOBAL statement includes the following parameters:

#### MAXVTV=nnn

optionally, 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" on page 3-37.

**MVCFREE=nnn**

optionally, 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**

optionally, 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 job name), which can also cause a VTV's Management Class to change.

---

---

**RECALWER**

optionally, 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**

optionally, specifies the Coupling Facility Structure that holds VTCS Lock Data. *structure-name* must be 16 characters or less and conform to IBM's standard for naming Coupling Facility Structures. Refer to Oracle's ELS publication *Installing ELS* for more information.

---



---

**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

optionally, 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 either of the following have occurred:

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

Once the expected number of MVC copies exist, replication does not occur.

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.

### VTVPAGE

optionally, specifies the page size used to store VTV data in the VTSS and on the MVCs. This setting only applies to 400 MB and 800 MB VTVs. If VTVPAGE is not specified on either the MGMTclas statement or the CONFIG GLOBAL statement, the default is STANDARD.

#### STANDARD

standard page size, which is compatible with all VSM3 or 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" on page 3-37. For 2 GB, 4 GB, and 32 GB VTVs (MAXVtvsz 2000, 4000, or 32000), a VTVPAGE setting of LARGE is always used.

VTVPAGE does not apply to VSM2s. VTVPAGE (LARGE) requires VSM4 or 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.

Consider the following:

- 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 or VTSS microcode does not support `LARGE`, then VTCS issues warning messages and `VTVPAGE` defaults to `STANDARD`.
- If you specify `STANDARD` for 2 GB or 4 GB VTVs, then 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.

**NLIBDRNR**

optionally, specifies how VTCS handles non-library resident MVCs for drain or reclaim processing.

**YES**

VTCS requests the mount of the non-library MVC. This is the default.

**NO**

VTCS suppresses the mount and purges the request.

For drain, non-library resident MVCs can be selected.

For reclaim, only library resident MVCs can be selected for processing, never non-library ones. However, between the time a library resident MVC is selected and the time it is actually processed, it may have become non-resident by being ejected.

**NLIBMIGR**

optionally, specifies whether non-library resident MVCs will be selected for Migration processing.

**YES**

Allow non-library resident MVCs to be selected (default).

**NO**

Do not allow non-library resident MVCs to be selected.

**NLIBRECL**

optionally, specifies whether non-library resident MVCs will be selected for `Recall` processing.

**YES**

Allow non-library resident MVCs to be selected (default).

**NO**

Do not allow non-library resident MVCs to be selected.

**MVCMNTTO=nn**

optionally, specifies the value in minutes when a mount of an MVC will timeout. Valid values are 5 to 30 minutes. The default is 15.

**INITMVC**

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

**NO**

Un-initialized MVCs should not be initialized. This is the default.



**YES**

Un-initialized MVCs should be initialized.

**SYNCHREP**

optionally, 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 "[MGMTclas Control Statement](#)" on page 3-222.

---



---

**MAXRTDS**

optionally, specifies the maximum number of RTDs supported.

**16**

up to 16 RTDs supported.

**32**

up to 32 RTDs supported.

**FASTMIGR**

optionally, specifies whether the stacked or streamed migrates feature is enabled for all VTSSs that support this feature.

**STREAM**

Specifies to use the streaming method for migrations. VTCS monitors responses from the RTD and uses them to decide as to when a VTV has become migrated. Full advantage is made of the buffer within the RTD to improve the throughput when performing migration.

This option also implies the use of the STACKED feature.

**STACKED**

Specifies to use the stacked method for migrations. VTCS maintains a small queue of requests to the VTSS. Advantage is made of the various buffers in the VTSS and RTD to improve the throughput when performing a migration. For backward compatibility, the value YES is the equivalent of STACKED.

**NO**

Disable stacked migrates (the default).

FASTMIGR=STREAM or STACKED has the following prerequisites:

- FASTMIGR=STACKED: VSM4/VSM5 microcode D02.05.00.00 or higher. If this level of microcode is not installed on all VTSSs in the configuration, Stacked Migration will be limited to the VTSSs that have it installed.
- FASTMIGR=STREAM: VSM4/VSM5 microcode D02.15.xx.00 or higher. If this level of microcode is not installed on all VTSSs in the configuration, Streamed Migration will be limited to the VTSSs that have it installed.
- ELS 7.0 or higher with PTFs.

- CDS level G or higher.
- FICON ports for FICON RTDs and CLINKs.

For the Stacked Migration feature to be enabled, all hosts must be running the prerequisites, otherwise:

- If a host is active and does not support or tolerate stacked migrates, then the `CONFIg` utility returns an error.
- If a host is started and does not support or tolerate this feature, then the host shuts down.

### **LOGPOL**

optionally, specifies whether VTCS CDS logging is optional or required.

#### **OPTIONal**

Logging is optional. This is the default. This mode is required for configurations that include 7.1, 7.2, 7.3 and lower level hosts.

#### **REQuired**

Logging is enabled for all events on all hosts that share the CDS. This requires all hosts to be at level 7.0 or higher. The following events are logged:

- new version of VTV
- imported VTV
- first use or re-use of an MVC
- imported MVC
- add VTV to MVC
- VTV on an imported MVC
- unlink VTV from MVC
- reclaim VTV from MVC
- reset MVC EOT backwards
- electronic export of VTV

### **MAXVTVSZ**

optionally, specifies a default maximum compressed VTV size (MB) that may be used during the creation of VTVs. Valid values for this parameter depend on both the CDS level and the microcode levels of the applicable VTSSs.

#### **400**

400 MB. This is the default.

#### **800**

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

#### **2000**

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

#### **4000**

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

#### **32000**

32 GB. The CDS must be at I level.

**Considerations:**

- 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)` or `MAXVTVSZ(4000)` requires VSM4 or VSM5 microcode `D02.02.00.00` or VSM3 microcode `N01.00.77.00`. No installed option is required.
- `MAXVTVSZ(32000)` requires VSM6 minimum microcode level 6.2 and VLE minimum microcode level 1.5.1, if VLE is in the configuration.

The `CONFIg GLOBAL` and `MGMTCLAS MAXVTVSZ` parameters interact as follows:

- If `MAXVTVSZ` is specified on `MGMTCLAS`, this value overrides the `CONFIg GLOBAL MAXVTVSZ` value.
- If `MAXVTVSZ` is not specified on `MGMTCLAS`, the `CONFIg GLOBAL MAXVTVSZ` value, if specified, is used. Otherwise, `MAXVTVSZ` defaults to 400 MB.
- If `MAXVTVSZ` is not specified on `MGMTCLAS` or on `CONFIg GLOBAL`, `MAXVTVSZ` defaults to 400 MB.

**LOCKTOUT=nnn**

optionally, specifies the minimum number of minutes that a resource is locked before message `SLS6946E` is issued.

Valid values are 0, or any value between 5 and 240. If 0 is specified, message `SLS6946E` will not be issued when a required resource is locked. If this parameter is not specified, the current default of 10 minutes is retained.

---



---

**Note:** `LOCKTOUT` is only supported at 'F' level CDS (`V61ABOVE`) and above.

---



---

**FASTRECL**

Optionally specifies whether VTCS performs Early Time to First Byte (ETTFB), also known as concurrent tape recall/mount, for all VTSSs that support the feature. This parameter applies to recalls from RTDs and recalls from VLE.

**NO**

Disable the ETTFB feature. This is the default.

**YES**

Enable the ETTFB feature. If you globally enable this feature, you can disable it for individual VTSSs through the `CONFIg VTSS NOERLYMT` parameter. See "[CONFIg VTSS Statement](#)" on page 3-59 for more information.

**VDGT256**

Optionally, specifies whether greater than 256 VTDs are to be used.

---



---

**Note:** `VDGT256=YES` is only valid with VSM model 7 and above. VSM 7 VTSS supports 512 VTDs with current maintenance.

---



---

**NO**

Specifies that only the first 256 VTDs will be used. The first 256 VTDs are on control unit 00 - 0F of the VTSS with VDID address of x0000 - x0F0F. NO is the default.

**YES**

Specifies that all configured VTDs in the VTSS can be used. This is only valid if the VTSS supports greater than 256 devices. The feature is globally enabled and applies to all VSMS that support greater than 256 devices. The feature can be disabled for an individual VTSS through the CONFIG VTSS NOGT256 parameter. See "CONFIg VTSS Statement" on page 3-59 for more information.

## CONFIg HOST Statement

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

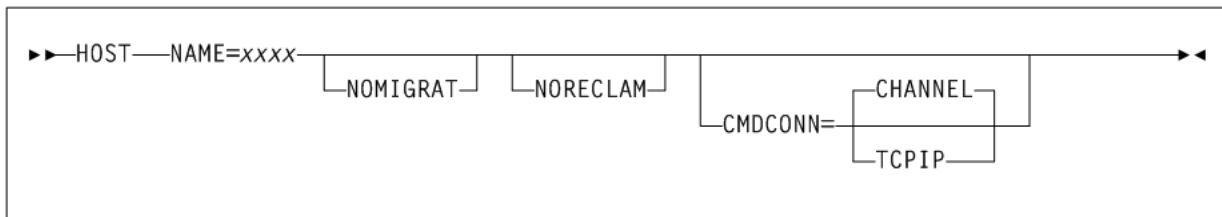
If specified, the HOST statement must follow the VTSS statement for the VTSS attached to that host.

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 3-18 shows CONFIg HOST statement syntax.

**Figure 3-18 CONFIg HOST statement syntax**



### Parameters

As shown in Figure 3-18, the CONFIg HOST statement includes the following parameters:

**NAME=xxxx**

specifies the LIBGENed hostname. xxxx indicates the host name.

**NOMIGRAT**

optionally, 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.

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**

optionally, specifies that this host cannot initiate automatic or demand reclaim processing using the VTSS(s) that the host accesses. The host can still perform MVC drains using `MVCDRAIN`. This parameter is optional.

**CMDCONN**

optionally, specifies one of the following VTSS access methods:

**CHANNEL**

indicates ESCON or FICON channel access. This is the default.

**TCPIP**

indicates TCP/IP access. TCPIP requires a valid CONFIg VTSS `IPCONN=(name-list)` value.

**CONFIg MVCVOL Statement**

The CONFIg `MVCVOL` statement defines a range of MVCs available to VTCS.

Oracle recommends that you use `SET VOLPARM` to define VTV and MVC ranges in the VTCS CONFIg.

- See "[SET VOLPARM](#)" on page 3-341 for more information about the `SET VOLPARM` statement.
- Refer to Oracle's ELS publication *ELS Legacy Interfaces Reference* for information about the use of the `VTVOL` and `MVCVOL` statements in CONFIg to define volumes.

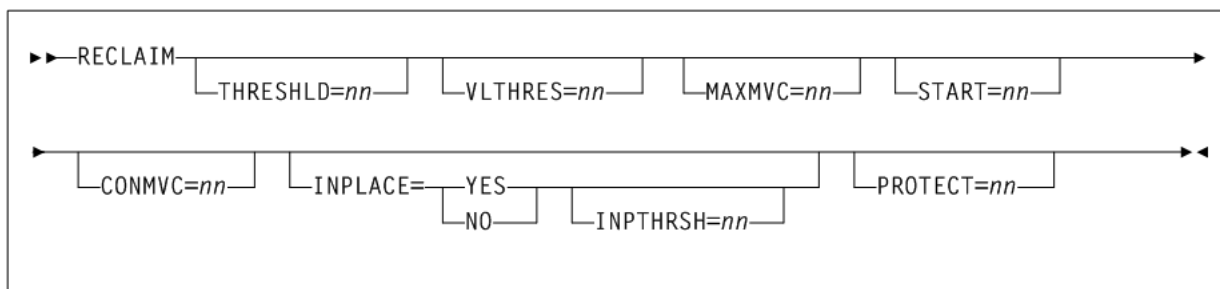
**CONFIg RECLAIM Statement**

The CONFIg `RECLAIM` statement controls demand and automatic MVC and VMVC space reclamation.

**Syntax**

Figure 3–19 shows CONFIg `RECLAIM` statement syntax.

**Figure 3–19 CONFIg RECLAIM statement syntax**

**Parameters**

As shown in Figure 3–19, the CONFIg `RECLAIM` statement includes the following parameters:

**THRESHLD=nn**

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

**VLTHRES=nn**

optionally, specifies the percentage of fragmented space (*nn*) that makes a Virtual MVC (VMVC) eligible for demand or automatic reclamation. Valid values are 4 to 98. The default is 35.

---



---

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

---



---

**MAXMVC=nn**

optionally, specifies the maximum number of MVCs that will be processed by a single space reclamation task.

*nn* indicates the maximum number of MVCs. Valid values are 1 to 98. The default is 40.

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

**START=nn**

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

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

where:

- *Reclaim Candidates* indicates the number of Reclaim Candidates determined by the CONFIG RECLAIM THRESHLD parameter.
- *Reclaim Candidates + Free MVCs* equals the number of Reclaim Candidates plus the number of free MVCs. Valid values are 1 to 98. The default is 35.

**CONMVC=nn**

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

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

**INPLACE**

optionally, enables or disables dynamic reclaim support within VTCS. There is no default, though the absence of INPLACE indicates that dynamic reclaim support is not enabled.

**YES**

Enable dynamic reclaim by formatting and processing all eligible MVCs as partitioned on a global (all storage class) basis.

**NO**

Disable dynamic reclaim. This option will likely be accompanied by STORCLAS level overrides.

The CDS must be at H level before INPLACE is accepted, enabling dynamic reclaim support. If INPLACE is specified and the CDS is not at H level, the configuration fails and error messages are displayed.

Only T10000B media is supported for dynamic reclaim. Sport volumes are not supported in partitioned format.

**INPTHRSH=nn**

optionally, specifies the percentage of fragmented space that makes an MVC in partitioned format eligible for dynamic reclaim processing.

*nn* is a percentage between 3 and 97. This value must be less than the `THRESHLD` value. The default is half the `THRESHLD` value (rounded up).

`INPTHRSH` cannot be specified without `INPLACE`.

If `INPTHRSH` is specified, `THRESHLD` must also be specified. Both `INPTHRSH` and `THRESHLD` play roles when dynamic reclaim processes MVCs in partitioned format. Refer to Oracle's ELS publication *Configuring HSC and VTCS* for more information.

**PROTECT=nn**

optionally, specifies the time, in hours, to prevent (or protect) an MVC from being reused after it is drained or reclaimed. Valid values are 1 to 99.

## CONFIg RTDpath Statement

The VTCS `CONFIg RTDpath` statement defines the path to one of the following:

- RTDs operating under a local TapePlex. Local RTD connections are defined using the `DEVNO` and `CHANIF` parameters and no `STORMGNR` parameter. Refer to Oracle's ELS publication *Configuring HSC and VTCS* for more information.
- A remote library. Remote library connections are defined via the `DEVNO`, `CHANIF` and `STORMGNR` parameters. Refer to Oracle's ELS publication *Configuring HSC and VTCS* for more information.
- A VLE connected to a VTSS, defined using only the `STORMNGR` and `IPIF` parameters. Refer to Oracle's ELS publication *Configuring the Host Software for VLE* for more information.

Except in tapeless environments, `CONFIg RTDpath` 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 or VSM6, 32 RTDs.
- For a VSM5 or VSM6, 4 VLE appliances.

---

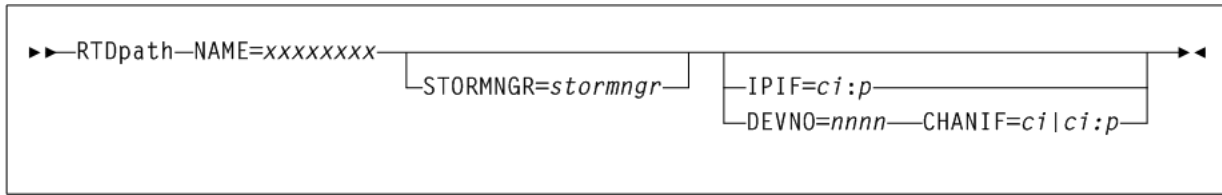
**Note:** You must specify the `RESET` parameter to change RTD definitions if VTCS is running with a CDS level lower than `V61ABOVE`. See "CONFIg" on page 3-37 for more information. For an initial RTD definition, if the RTD name displayed at the VTSS LOP is anything other than all blanks, you must also specify `RESET`.

---

### Syntax

Figure 3–20 shows `CONFIg RTDpath` statement syntax.

Figure 3–20 CONFIg RTDpath statement syntax



### Parameters

As shown in Figure 3–20, the CONFIg RTDpath statement includes the following parameters:

#### NAME=xxxxxxx

specifies the 1 to 8 character identifier of the RTD or VLE appliance.

- During VTCS CONFIg processing, RTDs with the same name may be defined for multiple VTSSs. RTD names must be unique across all VTSSs.
- For RTDs, you set or change the RTD identifier only using the RTD NAME parameter. To do so, the RTD identifier must be all blanks as displayed at the VTSS LOP or DOP.
- For VLEs or remote libraries, use any meaningful 1 to 8 character identifier.

This parameter is required; there is no default value.

#### STORMNGR=stormngr

specifies one of the following:

- VLE subsystem name
- remote library

*stormngr* indicates the Storage Manager name:

- For VLEs, this value must match the VLE Subsystem Name.
- For remote libraries, this value must match a TapePlex name defined on the SMC TAPEPLEX NAME parameter.

For more information, refer to Oracle’s ELS publication *Configuring the Host Software for VLE*.

#### IPIF=ci:p

The IPIF value for each VTSS to VLE port connection in *ci:p* format where:

- *c* indicates the VTSS Storage Cluster number (0 or 1).
- *i* indicates the interface number (A or I)
- *p* indicates the device number on the interface (0, 1, 2, or 3).

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

#### DEVNO=nnnn

specifies the unit address of the RTD or remote library. 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* indicates the VTSS Storage Cluster number (0 or 1).
- *i* indicates the interface number (A, C, E, G, I, K, M, or O)
- *p* indicates the device number on the interface (0, 1, 2, or 3).

For a VSM5, this CHANIF value must match the actual FICON interface values. For a VSM 6, this must be unique for each VTSS and does not correspond to an actual value on the VSM 6 FICON ports. For remote library connections, this value must be unique for each library and must follow the *ci:p* rules for RTDs.

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*).

Refer to Oracle’s ELS publication *Configuring HSC and VTCS* for more information.

---



---

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

---



---

## CONFIg STORMNGR Statement

The CONFIg STORMNGR statement defines a VLE to VTCS.

---



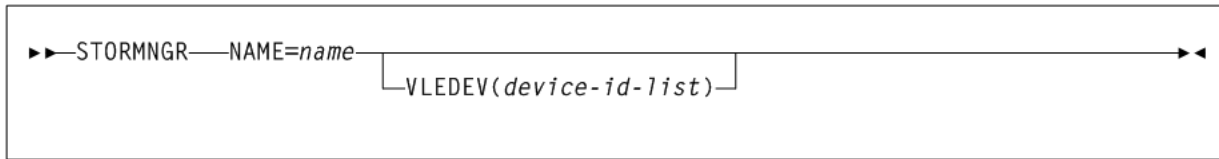
---

### Note:

- Do not specify this statement when down-level hosts are active. Once CONFIg STORMNGR is specified, down-level hosts no longer function.
  - This statement requires CDS level V62ABOVE or higher.
  - This statement is an alternative to the CONFIg TAPEPLEX statement. That is, you can either specify a list of VLEs on CONFIg TAPEPLEX or each VLE individually via CONFIg STORMNGR. If you code CONFIg STORMNGR statements, they must immediately follow the CONFIg TAPEPLEX statement to which the CONFIg STORMNGR statements apply.
  - For VTCS 7.3, you can define additional emulated VLE RTDs with the VLEDEV parameter. For more information, refer to Oracle’s VLE publication *Configuring the Host Software for VLE*.
- 
- 

### Syntax

Figure 3–21 shows CONFIg STORMNGR statement syntax.

**Figure 3–21 CONFIg STORMNGR statement syntax****Parameters**

As shown in [Figure 3–21](#), the CONFIg STORMNGR statement includes the following parameters:

**STORMNGR**

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

**NAME=name**

the Subsystem name of the VLE, which is set by the VLE installation scripts.

**VLEDEV(device-id-list)**

optionally, specifies the logical device IDs of a VLE. This parameter requires CONFIg CDSLEVEL=V71ABOVE (CDS Level H) and applies to VLE 1.2.

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

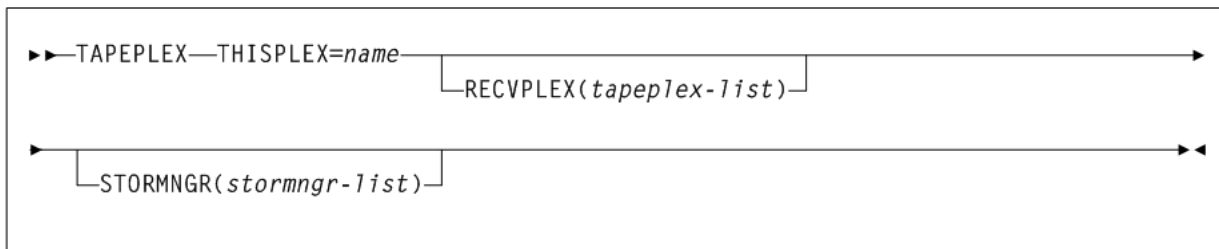
**CONFIg TAPEPLEX Statement**

The CONFIg TAPEPLEX statement defines values to VTCS for Cross-TapePlex Replication (CTR) or for replication to a VLE.

- 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.
- Refer to Oracle's ELS publications *Managing HSC and VTCS* and *ELS Disaster Recovery and Offsite Data Management Guide* for more information about Cross-TapePlex Replication.

**Syntax**

[Figure 3–22](#) shows CONFIg TAPEPLEX statement syntax.

**Figure 3–22 CONFIg TAPEPLEX statement syntax**

## Parameters

As shown in [Figure 3–22](#), the `CONFlg TAPEPLEX` statement includes the following parameters:

### **THISPLEX=name**

specifies the TapePlex name for this configuration. This name is associated with any VTV copies that are exported to other TapePlexes via Cross-TapePlex Replication.

*name* indicates the TapePlex name. This name must match the name specified on an `SMC LIBRARY` or `TAPEPLEX` command defined in the local SMC. 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.

### **RECVPLEX(tapeplex-list)**

optionally, specifies the list of TapePlexes from which VTV copies can be received (from Cross-TapePlex Replication). Attempts by hosts to send or scratch VTV copies from TapePlexes that are not included on the list are rejected.

*tapeplex-list* indicates the list of TapePlex names. 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:** Specify `RECVPLEX` on the receiving TapePlex. You can also specify `RECVPLEX` on the sending TapePlex to allow this TapePlex to recall a VTV from the receiving TapePlex.

---



---

### **STORMNGR(stormngr-list)**

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

*stormngr-list* indicates the list of VLE subsystem names. For more information, refer to Oracle's ELS publication *Configuring the Host Software for VLE*.

## CONFlg VTD Statement

The `CONFlg 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 and later models provide 256 VTDs per VTSS.

---



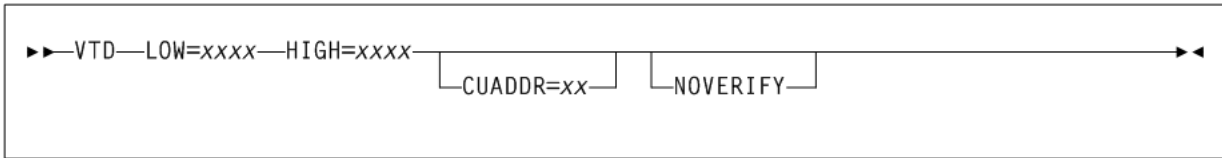
---

You can specify the VTD unit addresses to either apply to all hosts or to define which VTDs are available to specific hosts. See "[Specifying VTD Unit Addresses](#)" on page 3-58 for more information.

## Syntax

Figure 3–23 shows CONFIg VTD statement syntax.

**Figure 3–23 CONFIg VTD statement syntax**



## Parameters

As shown in Figure 3–23, the CONFIg VTD statement includes the following parameters:

### LOW=xxxx

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

### HIGH=xxxx

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

### CUADDR=xx

optionally, specifies a control unit identifier (xx) that matches its value in the IOCP. Valid values are 0 through 15 for VSM4 and 0 through 3 for VSM2 or VSM3. 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

optionally, suppresses VTD verification.

Normally, VTCS attempts verification of all predicted VTD identifiers associated with Virtual Tape Drives. In order to do this, each MVS I/O address must be available to allow the associated Host to issue the ECAM-T request `Virtual_Device_Identify`. ECAM is the method VTCS uses to communicate with a VTSS when bringing a VTSS online and scheduling replication and migrations.

In specific cases where, for example, VTCS is providing support for a remote client host, 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 fails and results in error messages posted to the operator. `NOVERIFY` suppresses verification and prevents these failures.

When `NOVERIFY` is specified for a device or range of devices, the following occurs:

- The device is reported to SMC so that VTVs can be mounted from MVS.
- VTCS does not use that device for ECAM I/O. At least one device must be left without `NOVERIFY` so that it can be used for ECAM I/O.

## Specifying VTD Unit Addresses

To specify VTD addresses, do 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.

The VTVs created and MVCs initially written to from a VTSS are considered to be resources of that VTSS. Therefore, only hosts with access to a VTSS have access to its VTVs and MVCs. In this type of “restricted” access configuration, 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 Oracle 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. You must include UNITATTR MODEL(IGNORE) commands for all overgened 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.

---

## CONFlg VTSS Statement

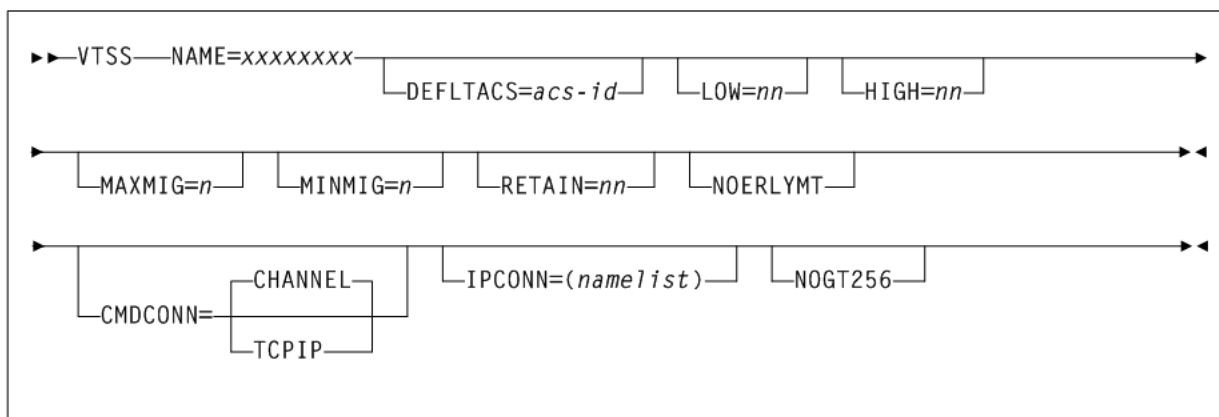
The CONFlg VTSS statement defines a VTSS and sets its operating values. This statement is required.

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

### Syntax

Figure 3–129 shows CONFlg VTSS statement syntax.

**Figure 3–24** CONFlg VTSS statement syntax



## Parameters

As shown in [Figure 3–24](#), the CONFIg VTSS statement includes the following parameters:

---



---

**Note:** If you physically remove a VTSS from your configuration, reconfigure the VTSS with a VTSS statement only (no parameters).

---



---

### **NAME=xxxxxxx**

specifies the VTSS name (xxxxxxx). This parameter is required; there is no default value. 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. 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 does not allow you to 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.

---



---

**Caution:** 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 one of the following:

- 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 Oracle StorageTek hardware representative must use the VTSS LOP or VOP to set the VTSS name to 99999999. This enables you to set the VTSS name to the value you want via the `NAME` parameter.

### **DEFLTACS=acs-id**

optionally, VTCS supports multi-VTSS confirmations, and supports connecting multiple ACSs to each VTSS. In configurations where a VTSS is connected to multiple ACSs, use the `DEFLTACS` parameter to specify the default ACS (*acs-id*) from which MVCs will be selected for migration, consolidation, and reclaim processing. You can migrate up to 4 VTV copies to separate ACSs (per `MGMTclas` statement) from the `MGMTclas MIGpol` parameter (and ignores the `DEFLTACS` parameter, as described below).

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

VTCS ignores the `DEFLTACS` value 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**

optionally, specifies the low automatic migration threshold (LAMT) for this VTSS. *nn* indicates the threshold value.

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

**HIGH=nn**

optionally, specifies the high automatic migration threshold (HAMT) for this VTSS. *nn* indicates the threshold value.

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

**MAXMIG=n**

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

*n* indicates the number of tasks. Valid values are 1 to the number of RTDs attached to the VTSS. The default is half the number of RTDs attached to the VTSS.

**MINMIG=n**

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

*n* indicates the number of tasks. Valid values are 1 to the MAXMIG setting. The default is 1 task.

**RETAIN=nn**

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

*nn* indicates the number of minutes. Valid values are 1 to 60. The default is 15.

An MVC on an idle RTD is not dismounted until the RETAIN time has elapsed, unless the RTD in question is required, and no other 'like RTDs' (same ACS, LSM, and device type) are free.

An exception to this behavior occurs when the last operation on the RTD was a job-initiated recall or an 'MVC is full' condition has occurred. In this case, the MVC is dismounted from the idle MVC when VTCS has determined that, although the RTD is not currently needed, there are no other 'like' RTDs free.

**NOERLYMT**

optionally, disables Early Time to First Byte (ETTFB), also known as the concurrent tape recall/mount feature, for this VTSS. This parameter only applies if CONFIg GLOBAL FASTRECL=YES. This parameter applies to recalls from RTDs and recalls from VLE.

**CMDCONN**

optionally, specifies the VTSS access method:

**CHANNEL**

indicates ESCON or FICON channel access. This is the default.

**TCPIP**

indicates TCP/IP access. This requires a valid CONFIg VTSS IPCONN=(*name-list*) value.

**IPCONN=(name-list)**

optionally, specifies a comma-separated list of one to four IP addresses, either explicitly as IP addresses or implicitly by names that resolve to IP addresses, used to define TCP/IP connections.

*name-list* indicates the list of IP addresses or names. Values entered in the list cannot be verified until a VTCS system is initialized to use them. They must meet the following requirements:

- Any IPv4 values must be specified in format *nnn.nnn.nnn.nnn*, where each byte is one to four hexadecimal characters. There must be four bytes, numeric only except for the decimal points.
- Any IPv6 values must be specified in format *x.x.x.x.x.x.x.x*, where each byte is 0-255. There must be four bytes: 0-9, A-F (or a-f). IPv6 abbreviations are not allowed. The full eight part address must be specified.
- If names are specified, they can be a maximum of 60 characters. A name can be any valid DNSname containing only the characters A-Z, a-z, 0-9, '-', or '.'.

**NOGT256**

optionally, disables greater than 256 VTD support for this VTSS. This parameter only applies if `CONFIG GLOBAL VTDGT256=YES` is specified.

---

---

**Note:** `VTDGT256=YES` is only valid with VSM model 7 and above. VSM 7 VTSS supports 512 VTDs with current maintenance.

---

---

## CONFIg VTVVOL Statement

The `CONFIg VTVVOL` statement defines a range of VTVs.

Oracle recommends using `SET VOLPARM` to define VTV and MVC ranges in the VTCS `CONFIg`.

- See "[SET VOLPARM](#)" on page 3-341 for more information about the `SET VOLPARM` statement.
- Refer to Oracle's ELS publication *ELS Legacy Interfaces Reference* for information about the use of the `VTVVOL` and `MVCVOL` statements in `CONFIg` to define volumes.

## CONSolid

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

## Description

The `CONSolid` command consolidates VTVs on MVCs.

---

---

**Note:** VTVs cannot be consolidated onto vMVCs in a VLE.

---

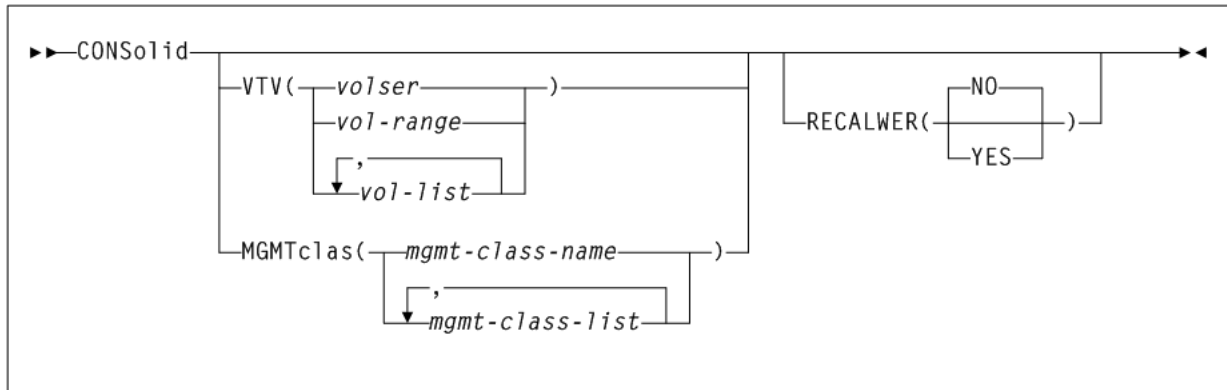
---



## Syntax

Figure 3–25 shows CONSolid command syntax.

Figure 3–25 CONSolid command syntax



## Parameters

As shown in Figure 3–25, the CONSolid command includes the following parameters:

### **VTV(volser, vol-range, or vol-list)**

optionally, specifies one or more VTVs to consolidate.

*volser*, *vol-range*, or *vol-list* indicates the *volser*, *volser* range, or *volser* list of VTVs. You can specify a maximum of 2,000 VTVs.

### **MGMTclas(mgmtclas-name or mgmtclas-list)**

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

*mgmt-class-name* or *mgmt-class-list* indicates 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 3-222. You can consolidate a maximum of 2,000 VTVs by specifying a Management Class.

### **RECALWER**

optionally, specifies whether VTCS recalls VTVs with read data checks.

#### **NO**

Do not recall VTVs with read data checks. This is the default.

#### **YES**

Recall VTVs with read data checks.

## CONSolid Report Messages

The consolidation report displays the following messages:

```
MIGRATE ONLY FROM VTSS vtssname
```

*Explanation:* The VTV is resident on VTSS *vtssname*.

```
REMIGRATE FROM MVC mvcname VIA VTSS vtssname
```

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

VTV *vtvname* NOT SELECTED; VTV IS SCRATCH

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

VTV *vtvname* NOT SELECTED; VTV ALREADY CONSOLIDATED

*Explanation:* The specified VTV is already consolidated.

VTV *vtvname* NOT SELECTED; VTV RECORD NOT FOUND

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

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

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

REDRIVING REQUEST BECAUSE OF ERROR

*Explanation:* VTCS is retrying an unsuccessful consolidation request.

CONSOLID CMD PROBLEM DECODING VCI REQUEST FROM HSC

*Explanation:* The consolidation failed.

VTV *vtvnumber* NOT SELECTED: LIMITED ACCESS TO VTSS

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

MIGRATE NO MVCS AVAILABLE

*Explanation:* Sufficient free MVCs are not available to complete the request.

## DBSERVer

### Interfaces:

- Console or Utility only
- UUI Support: No

### Subsystem Requirements:

Active HSC required. VTCS must not be active.

## Description

The `DBSERVer` command starts or stops a VSM console oVTCS CDS database server. The oVTCS CDS database server services CDS database I/O requests from an oVTCS client.

---

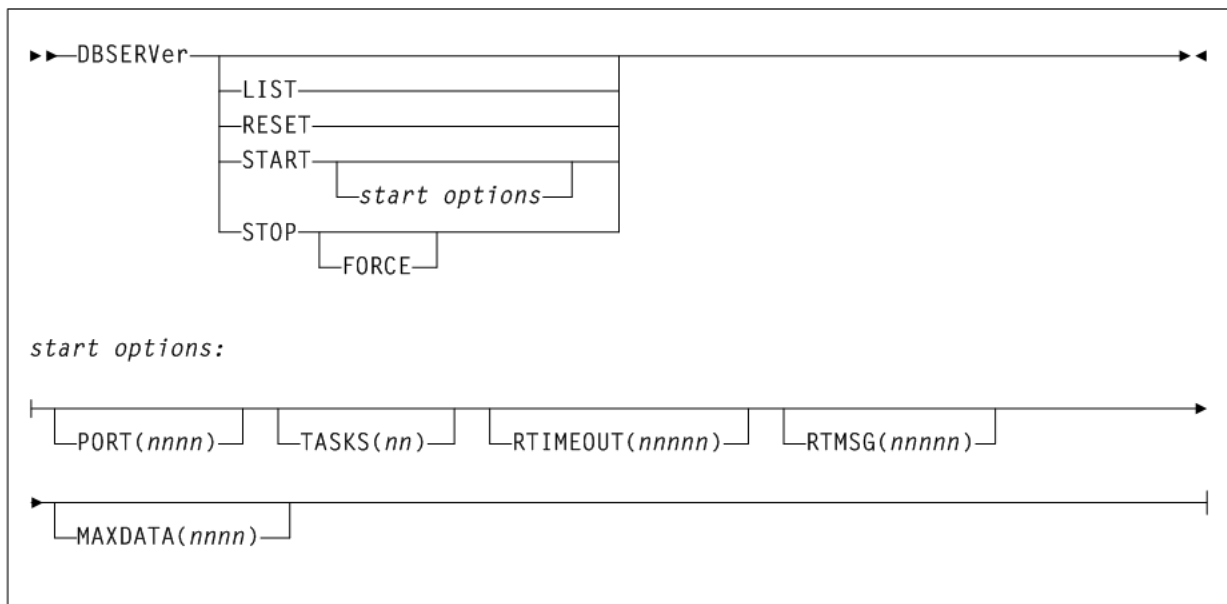
**Note:** You can only start an oVTCS CDS database server in an HSC subsystem with no executing VTCS component.

---

## Syntax

Figure 3–26 shows `DBSERVer` command syntax.

Figure 3–26 *DBSERVer* command syntax



## Parameters

As shown in Figure 3–26, the `DBSERVer` command includes the following parameters:

### LIST

optionally, list the settings and status of the oVTCS CDS database server.

### RESET

optionally, force a reset of the oVTCS CDS database server which closes the accepted socket, clear any pending work, and listen again for the oVTCS client to re-connect.

### START

optionally, start the oVTCS CDS database server.

#### PORT(nnnn)

optionally, specifies the socket listener port.

*nnnn* indicates the port. If you do not specify this parameter, the default is 8081.

#### TASKS(nn)

optionally, specifies the maximum number of tasks for asynchronous CDS reads.

*nn* indicates the maximum number of tasks. Allowable values are 1-25. If you do not specify this parameter, the default is 5.

**RTIMEOUT(nnnnn)**

optionally, specifies the reserve timeout.

*nnnnn* indicates the timeout in minutes. Allowable values are 1-99999. If you do not specify this parameter, the default is 99999 (69 days).

**RTMSG(nnnnn)**

optionally, specifies the time interval to display the SLS0789I "CDS reserved by client" message.

*nnnnn* indicates the time interval in minutes. Allowable values are 1-99999. If you do not specify this parameter, the default is 60 (1 hour).

**MAXDATA(nnnn)**

optionally, specifies the maximum CDS block length for tracing.

*nnnn* indicates the block length. Allowable values are 0-4096. If you do not specify this parameter, the default is 64.

**STOP**

optionally, stop the oVTCS CDS database server.

**FORCE**

optionally, force termination even when the oVTCS CDS database server has not terminated successfully.

## DEComp

**Interfaces:**

- Utility only
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC not required

## Description

The DEComp command lists the VSM configuration information in the HSC CDS.

## Syntax

Figure 3–132 shows DEComp command syntax.

**Figure 3–27 DEComp command syntax**



## Parameters

As shown in [Figure 3-27](#), the DEComp command includes the following parameters:

### FLATdd(ddname)

optionally, specifies the output destination dd name if a flat file is required. *ddname* indicates the dd name of the flat file included in the JCL.

### LISTDELR

optionally, list volser ranges of “logically deleted” MVCs and VTVs.

---



---

**Note:** LISTDELR is only valid for a CDS at level G or above.

---



---

## Output

The following examples show sample output for the DEComp command.

### FLATDD Parameter Output

[Example 3-4](#) shows an example of DEComp output with the FLATDD parameter.

#### **Example 3-4 DEComp FLATDD output**

```
CONFIG CDSLEVEL(V62ABOVE)
GLOBALMAXVTV=32000 MVCFREE=40 VTVattr=ALLmount RECALWER=YES
REPLICAT=ALWAYS LOCKSTR=STK_VTCS_LOCKS
RECLAIMTHRESHLD=70 MAXMVC=40 START=35 CONMVC=1 VLTHRES=1 +
INPLACE=Yes INPTHRSH=15
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
TDNAME=VTS28805 DEVNO=8805 CHANIF=0I
RTDNAME=VTS28806 DEVNO=8806 CHANIF=1A
RTDNAME=VTS28807 DEVNO=8807 CHANIF=1I
VTDLOW=9900 HIGH=993F
```

### SLSPRINT Output

[Example 3-5](#) shows an example of DEComp output to SLSPRINT.

#### **Example 3-5 DEComp SLSPRINT output**

```
DECOM
SLS1315I SSRDMP.P775644.TESTCDS WAS SELECTED AS THE PRIMARY CONTROL DATA SET
TIME 09:07:06 VTCS DECOMPILE
CONFIG CDSLEVEL(V62ABOVE)
GLOBAL MAXVTV=32000 MVCFREE=40 VTVattr=ALLmount RECALWER=YES
```

## DELETSCR

---

```
REPLICAT=ALWAYS LOCKSTR=STK_VTCS_LOCKS
RECLAIM THRESHLD=70 MAXMVC=40 START=35 COMMVC=1 VLTHRES=1 +
INPLACE=Yes INPTHRSH=15
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
```

## DELETSCR

### Interfaces:

- Utility only
- UI Support: Yes

### Subsystem Requirements:

Active HSC/VTCS

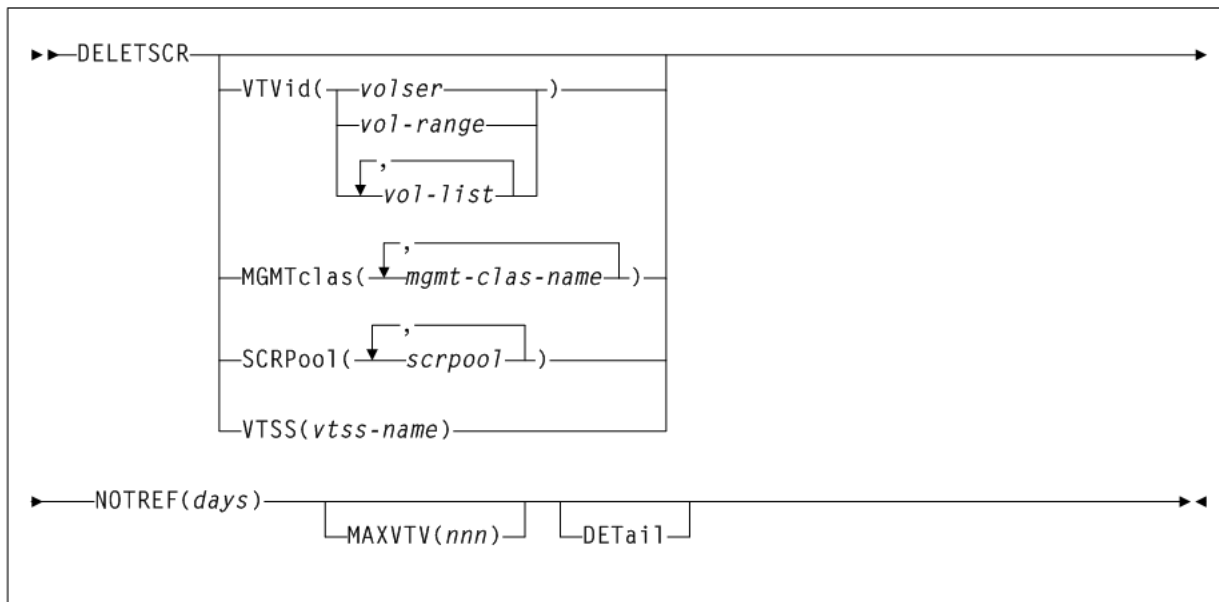
## Description

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

## Syntax

Figure 3–28 shows DELETSCR command syntax.

Figure 3–28 DELETSCR command syntax



## Parameters

As shown in Figure 3–28, the DELETSCR command includes the following parameters:

### VTVID(volser, vol-range, or vol-list)

optionally, specifies one or more VTVs to be deleted.

*volser*, *vol-range*, or *vol-list* indicates the volsers of one or more VTVs.

### MGMTCLAS(mgmtclas-name)

optionally, specifies the Management Class(es) of the VTVs.

*mgmt-class-name* indicates the Management Class name that you specified on the MGMTCLAS control statement.

### SCRPOOL(scrpool)

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

*scrpool* is an existing HSC scratch pool.

### VTSS(vtss-name)

optionally, causes scratch VTVs within the specified VTSS to be deleted.

*vtss-name* indicates the VTSS name.

---

**Note:** VTVID, MGMTCLAS, SCRPOOL, and VTSS are mutually exclusive. If you do not specify one of these parameters, VTCS processes all scratch VTVs.

---

### NOTREF(days)

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

*days* indicates the number of days (1-999).

**MAXVTV(nnn)**

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

*nnn* indicates 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**

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

## DELETSCR Report

Example 3–6 is an example of a DELETSCR report for the following batch invocation:

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

**Example 3–6 DELETSCR report**

```
SLUADMIN (7.3.0)      StorageTek Enterprise Library Software Utility      PAGE 0001
TIME 06:32:03        SCRATCH VTV DELETE                                          DATE 2015-03-31
```

```
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
```

A DELETSCR report shows the following:

- 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.

## DIRBLD

### Interfaces:

- Utility only
- UUI Support: No

### Subsystem Requirements:

Active HSC not required

## Description

The DIRBLD command rebuilds the database directory for all defined CDS copies.

The control data set contains a directory of pointers to various important locations and if damaged, can cause operational problems. Normally, HSC corrects such errors automatically. The DIRBLD utility repairs a corrupted CDS database directory. This utility can be run while the HSC is stopped or running to repair damage to the directory and maintain continuous operation. It must be run using all CDS copies as input.

---



---

**Caution:** Contact Oracle StorageTek Support before running the DIRBLD utility to insure that the CDS is not further damaged by inappropriate use or inappropriate operational conditions.

---

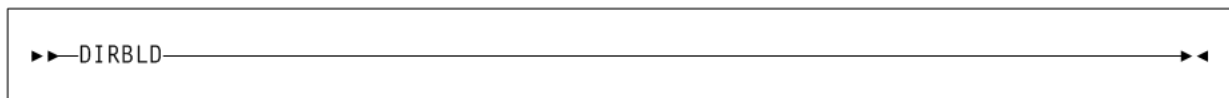


---

## Syntax

Figure 3–29 shows DIRBLD command syntax.

**Figure 3–29** DIRBLD command syntax



## Parameters

None.

## DISMount

### Interfaces:

- Console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC at FULL service level

## Description

The DISMount command dismounts a cartridge or VTV.

Although the volume may be rewound and unloaded by the operating system, it may not be dismounted from the real drive or VTD. This command is provided for instances where hosts are not communicating with a specific ACS (disconnected mode).

Hosts that are still communicating (connected mode) may be able to semi-automate tape handling for hosts that cannot communicate with a specific ACS. When HSC or VTCS does not dismount a volume, you must ensure that the volume is unloaded before you issue the DISMount command.

---

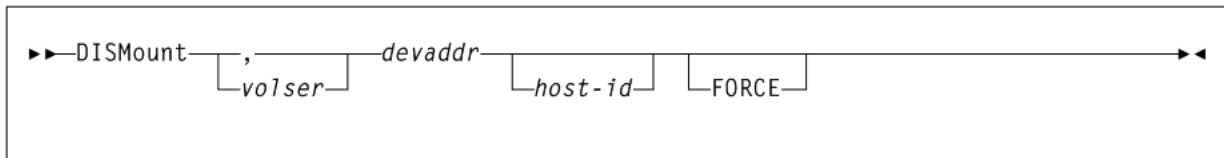
**Note:** The HSC DISMount command does not support VTDs that are not defined in the VTCS CONFIG. The SMC DISMount command must be used in place of the HSC DISMount command for any VTDs that are not specified in the VTCS CONFIG.

---

## Syntax

Figure 3–30 shows DISMount command syntax.

**Figure 3–30 DISMount command syntax**



## Parameters

As shown in Figure 3–30, the DISMount command includes the following parameters:

### volser

optionally, specifies which volume to dismount. The *volser* operand is optional. If it is not specified, the robot dismounts whatever volume is mounted on the device.

If *volser* is not specified, a comma must be entered immediately before the device address to indicate the missing operand. For example:

```
DISMOUNT ,B00
```

### devaddr

specifies the device address of the transport from which the volume is to be dismounted.

### host-id

optionally, indicates that the DISMount command is to be performed for the device address of the specified host (the SMF system identifier for both JES2 and JES3).

### FORCE

optionally, specifies that the drive will be unloaded before the volume is dismounted. This parameter is not valid for virtual drives.

## Display

The Display command displays status and current settings for various ELS components.

Issue this command with any of the options listed in the following table. Each option is described individually, and in more detail, on the pages to follow.

**Table 3–1 Display Options**

Option	Description
Acs	status for one or more ACSs.
ACTive	active VTCS processes
ALl	CDS and ACS status
ALLOC	HSC allocation settings
Cap	CAP settings and status
CDS	status of all copies of the CDS, some operational settings (e.g., the cleaning cartridge prefix, SMF record type, etc.), and the status of all HSC-defined hosts
CLink	cluster link status
CLUster	cluster status
CMD	detailed information for an ELS command
COMMPath	current settings for HSC host-to-host communications
CONFIG	CONFIG parameter settings
DRives	current and queued mount activity for each transport address
DRIVE_INFO	status information for real and/or virtual drives
EXceptns	hardware status
LINKSto	replication link and CLINK status information
LMUPDEF	information about the definition data set, containing LMUPATH statements, that is currently loaded
LOCKs	VTCS lock status
Lsm	LSM status
Message	detailed information for an ELS message
MGMTDEF	information about active MGMTclas statements
MIGrate	migration status
MNTD	current settings for HSC mount options set by the MNTD command
MONitor	list of monitoring consoles. Refer to Oracle's ELS publication <i>ELS Legacy Interfaces Reference</i> for more information about this option.
MVC	information about a specific MVC
MVCDEF	information about active MVCPool statements. Refer to Oracle's ELS publication <i>ELS Legacy Interfaces Reference</i> for more information about this option.
MVCPool	information about a specific MVC pool
OPTion	current settings for general HSC options set by the OPTion command

**Table 3–1 (Cont.) Display Options**

<b>Option</b>	<b>Description</b>
PATH	status of paths from VTSSs to either RTDs or Virtual Libraries.
Queue	status of queued VTCS processes
REPLicat	VTV replication status
Requests	all pending LMU requests
RTD	usage information for the one or more RTDs
SCRatch	scratch counts by subpool name, ACS id, LSM id, media type, recording technique, or owning host name
SCRPDEF	information about the definition data set, containing SCRPOOL statements, that is currently loaded. Refer to Oracle's ELS publication <i>ELS Legacy Interfaces Reference</i> for more information about this option.
SEN	SEN LISTEN requests and their status.
SERVER	Server (HSC) status, including service level and features
SRVlev	current service level of the HSC on the system from which you issue the command
Status	status of pending requests currently active on the host issuing the command
STORCLas	information about a Storage Class
STORMNgr	status of an external storage manager and the paths defined to it from the VTSSs.
TASKs	task status
THRshld	scratch count and scratch threshold information by subpool name, ACS id, LSM id, media name, and recording technique name
VOLDEF	information about the definition data set, containing VOLATTR statements, that is currently loaded. Refer to Oracle's ELS publication <i>ELS Legacy Interfaces Reference</i> for more information about this option.
Volser	volume information
VOLUME_Info	status information for real and/or virtual volumes
VSCRatch	diagnostic information for virtual scratch counts
VTD	VTD status
VTSS	VTSS status
VTV	VTV status

## Display ACS

### Interfaces:

- Console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC at FULL service level

## Description

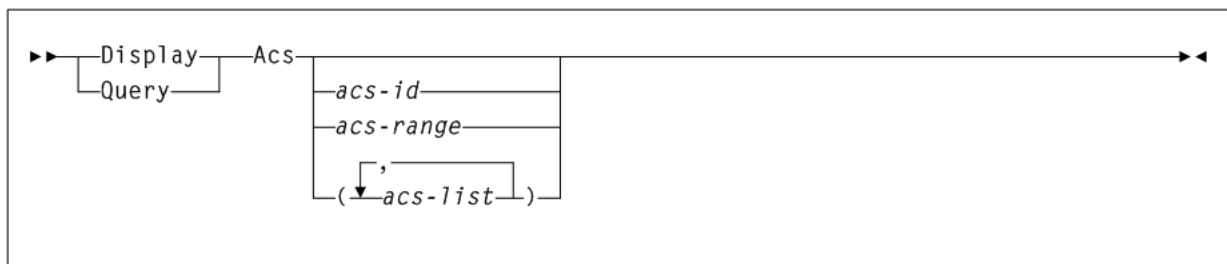
The `Display Acs` command displays the status of one or more ACSs. Information may include the following:

- partition ID
- LMU and HSC compatibility levels
- redundant electronics connections (summary of all Library Controllers)
- available scratch volumes and free cells

## Syntax

Figure 3–31 shows `Display ACS` command syntax.

**Figure 3–31** *Display Acs command syntax*



## Parameters

As shown in Figure 3–31, the `Display Acs` command includes the following parameters:

### **acs-id, acs-range, or acs-list**

one or more ACSs for which the system is to display status. Each *acs-list* element can be either a single ACSid or an ACSid range. The elements in a list must be separated by commas or blanks, and the entire list must be enclosed in parentheses.

If you do not supply an *acs-id*, the status of all defined ACSs in the library is displayed.

## Output

The following is an example of the output produced by the `Display Acs` command:

### **Example 3–7** *Display ACS output*

```

SLS0000I D ACS
SLS1969I User exit 15 module SLSUX15 INACTIVE
SLS1000I ACS 00 status: Connected
Active Queue elements          1
Partition ID=005
Compatibility levels: HSC=23, LMU=23
Redundant Electronics is Configured
Scratch Volumes available..... 0
Free Cells available.....      20
SLS1000I ACS 01 status: Disconnect

ACS slots available for COMPLEX 7
  
```

## Display ACTIVE

### Interfaces:

- Console or utility
- UII Support: Yes

### Subsystem Requirements:

Active HSC/VTCS

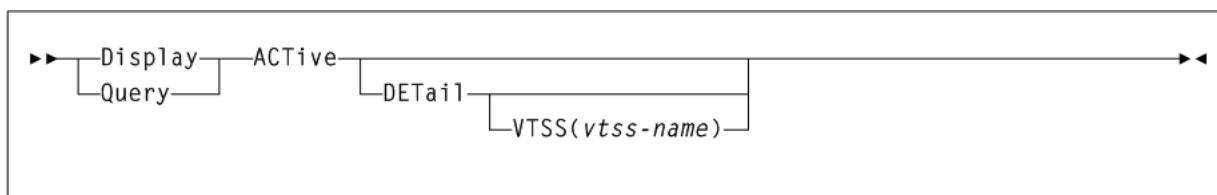
### Description

The `Display ACTIVE` command displays active VTCS processes.

### Syntax

Figure 3–32 shows `Display ACTIVE` command syntax.

**Figure 3–32** *Display ACTIVE command syntax*



### Parameters

As shown in Figure 3–32, the `Display ACTIVE` command includes the following parameters:

#### DETail

optionally, display detailed status.

#### VTSS

optionally, display processes for the specified VTSS.

*vtss-name* indicates the VTSS name.

### Output

The following is an example of the output produced by the `Display Active` command (with no detail):

#### **Example 3–8** *Display ACTIVE output (no detail)*

```
MIGRATES=3 RECALLS=2 RECLAIMS=0
```

The following is an example of the output produced by the `Display Active DETail` command:

#### **Example 3–9** *Display Active DETail output*

Function	ID	VTV	MVC	RTD	VTSS	Task	Reason
Reclaim@	62180	-	-	-	-	RCM	Child Finish
.ReclmMVC	62251	-	V10191	-	-	RCM	Chile Finish
. VtvMover	62254	-	V10191	-	-	RCM	Child Finish
. Migrate	02003	-	-	-	TVSM22N1	Drv	MVC for command
. Migrate	02090	-	-	-	TVSM22N1	Drv	MVC for command

. Migrate	02148	-	-	-	TVSM22N1	Drv	MVC for command
. Migrate	02163	-	-	-	TVSM22N1	Drv	MVC for command
. Migrate	02184	-	-	-	TVSM22N1	Drv	MVC for command
. Migrate	02229	-	-	-	TVSM20N1	Drv	MVC for command
. Migrate	02242	-	-	-	TVSM20N1	Drv	MVC for command
Display@	02277	-	-	-	-	DSP	

---

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

---

### Fields

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. A period (.) before the **Function** indicates that the **Function** is a child.

**Function** 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.

#### **DELETSCR**

Delete scratch utility.

#### **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.

**Getconfig**

Get configuration information.

**Getmgpol**

Obtain current management and storage class definitions.

**HSCChnge**

Notification of parameter files being changed.

**Import#**

Importing of VTV or MVC by a utility.

**Int\_cons**

PGMI initiated consolidate request.

**Inventry**

`INVENTORY` utility.

**MEDVERfy**

Media Verify parent task

**Mig\_set@**

Set migration threshold command

**Mig\_thr@**

Migrate to threshold command.

**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 `Move MVC`, which is a child request of a `MoveVTV#` request, is responsible for managing the entire VTV movement process for a single MVC. Depending upon circumstances, this may be subsequently seen as a VTV transfer or recall request. The **VTV** column is used to indicate the status of the processing against the MVC.



**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.

**MVC\_chek**

Query MVC. Check status of MVC.

**MVC\_eot**

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

**MVC\_inv**

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

**MVC\_upd**

Reset or update MVC status.

**MvcMaint**

MVCMAINT utility 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.

**QRY/SET@**

Query or Set command.

**Query@**

Query or Display command.

**Recall**

General request to perform the recall of VTVs from an MVC. This may include a Cross TapePlex Autorecall (CTA) request from the mounting system.

**Recall** 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**

A Scratch a VTV request from HSC.

**Sel\_scr**

PGMI select scratch.

**Set@**

Set command.

**Trace@**

Trace command

**Transfer**

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

**Unload**

Unload MVC from RTD.

**Unscratch**

An Unscratch a VTV request from HSC.

**Vary@**

A Vary command. This command applies to resources such as CLINKs, VTSSs and ACSs.

**Vary\_dev**

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

**VTSS\_inv**

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

**VTSS\_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\_chek**

Query VTV.

**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. This Function can show an invalid VTV ID of VTS0B, VTS0C or VTS0D and maybe other similar ones. It also shows a VTSS ID and a task of SS. Invalid VTV IDs may occur because VTCS sometimes uses the fields that hold VTV/MVC volsers for other internal purposes, and the display routines don't filter out those cases.

**VTV\_upd**

Resynchronize VTV status in the VTSS with the CDS.

**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

**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`). **Task** displays one of the following values:

**CIK**

CLINK processor task.

**CMD**

Command processor task.

**Csh**

Clink scheduler task.

**DRV**

RTD scheduler task.

**DSP**

Main dispatcher task.

**INV**

Inventory manager or audit task.

**MSc**

Migration scheduler task.

**MVC**

MVC lock task.

**RCM**

Drain/space reclaim manager task.

**RTD**

RTD task.

**Scr**

Scratch manager task.

**SS**

VTSS task.

**TSK**

Waiting for processing lock on other host.

**unk**

Unknown task.

**VTD**

Waiting for VTD.

**VTV**

Waiting for VTV lock.

**REASON**

Indicates 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.

**Created**

The request is being created.

**DBU DROP**

The request is currently held because the DBU is high.

**DEVICE LOCK**

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

**MVC dismount**

The request is waiting for an MVC to dismount.

**MVC for Class**

The SCHLIMIT storage class limit has been reached in MVC selection.

**MVC for Command**

The SCHLIMIT command concurrency limit has been reached in MVC selection.

**MVC for Funct**

The SCHLIMIT function limit has been reached in MVC selection.

**MVC LOCK**

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

**MVC MOUNT**

The request is waiting for a MVC to be mounted.

**MVC SELECTION**

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

**MVC/SCHLIMIT=0**

MVC selection determined that SCHLIMIT=0 has been specified.

**QUEUED**

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

**RTD ALLOCATION**

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

**RTD LOCK**

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

**RTD ONLINE**

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

**STEAL A RTD**

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

**TASK LOCK**

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

**VTD LOCK**

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

**VTVLOCK**

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

**Wait child**

The request has been purged, but is waiting for child requests to finish.

**WAIT RESOURCE**

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

**Wait lock**

The request is waiting for a lock to free.

## Display ALI

**Interfaces:**

- Console or PARMLIB only
- UUI Support: No

**Subsystem Requirements:**

Active HSC at FULL service level

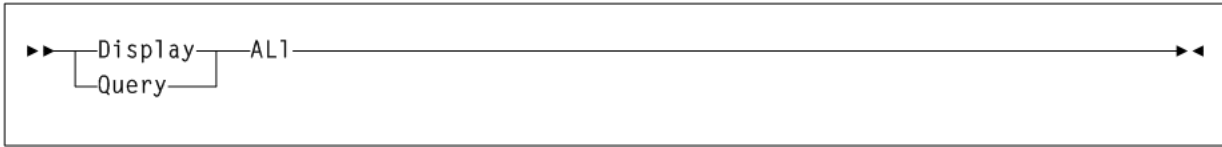
**Description**

The `Display ALI` command displays CDS and ACS status.

**Syntax**

Figure 3–33 shows `Display ALI` command syntax.

**Figure 3–33 Display ALI command syntax**



**Parameters**

None.

**Display Cap**

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at FULL service level

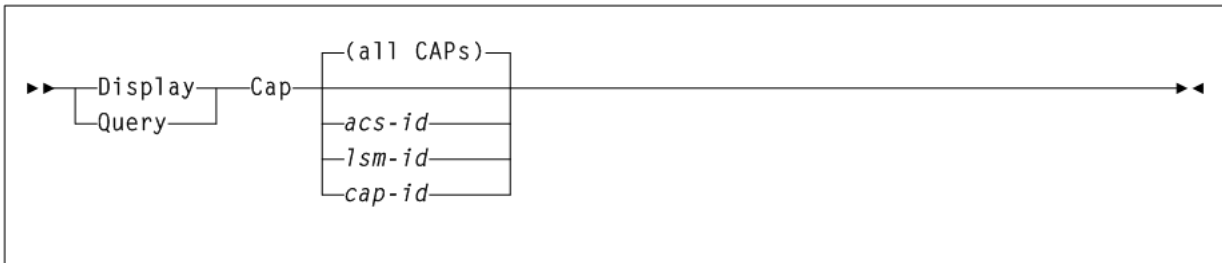
**Description**

The `Display Cap` command displays CAP settings and status.

**Syntax**

Figure 3–34 shows `Display Cap` command syntax.

**Figure 3–34 Display Cap command syntax**



**Parameters**

As shown in Figure 3–34, the `Display Cap` command includes the following parameters:

---



---

**Note:** If the CAP status indicates `RECOVERY`, refer to Oracle’s ELS publication *Managing HSC and VTCS* for information about how to clear this condition.

---



---

**all CAPs**

displays all CAP activity for all ACSs online to this host.

**acs-id**

optionally, displays all CAP activity for a specified ACS.

**lsm-id**

optionally, displays all CAP activity for a specified LSM.

**cap-id**

optionally, displays all CAP activity for a specified CAP.

**Note:**

- The *acs-id*, *lsm-id*, and *cap-id* parameters are positional operands. If no positional operand is specified, the default operation is to display the status of all CAP activity for all ACSs.
- The *acs-id*, *lsm-id*, and *cap-id* may be expressed as a list.

**Output**

The following is an example of the output produced by the `Display Cap` command for Bulk CAPs with 36 slot capacity:

**Example 3-10 Display Cap output**

```
SLS0000I D CAP
SLS2008I CAP Status: 874
CAP ID      Size      Partid  Hostid  Priority  Mode      Status
-----
00:00:00   36-CELL   None    None    00       Idle      Manual
                                           Online
00:00:01   36-CELL   None    None    00       Idle      Manual
                                           Online
00:01:00   36-CELL   None    None    00       Idle      Manual
                                           Online
00:01:01   36-CELL   None    None    00       Idle      Manual
                                           Online
00:02:00   36-CELL   None    None    00       Idle      Manual
                                           Online
00:02:01   36-CELL   None    None    00       Idle      Manual
                                           Online
00:03:00   36-CELL   None    None    00       Idle      Manual
                                           Online
00:03:01   36-CELL   None    None    00       Idle      Manual
                                           Online
```

The `Display Cap` command displays the following information:

- CAPid
- CAP size:
  - PCAP (priority CAP)
  - 21-cell or 14-cell 9740 (standard CAP)
  - 40-cell (enhanced CAP)
  - 20-cell (9360 CAP)
  - 30-cell (9360 optional CAP)
  - 26-cell (SL3000 CAP)
  - 234-cell (SL3000 AEM CAP)
  - 39-cell (SL8500 CAP)

- 36-cell (SL8500 Bulk CAP)
- Partition ID
- Host ID of the host owning the CAP
- Priority: CAP preference value
- CAP mode:
  - cleaning
  - draining
  - ejecting
  - entering
  - idle
- CAP status:
  - active
  - automatic mode
  - manual mode
  - offline
  - online
  - recovery needed

## Display CDS

### Interfaces:

- Console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC at `BASE` or `FULL` service level

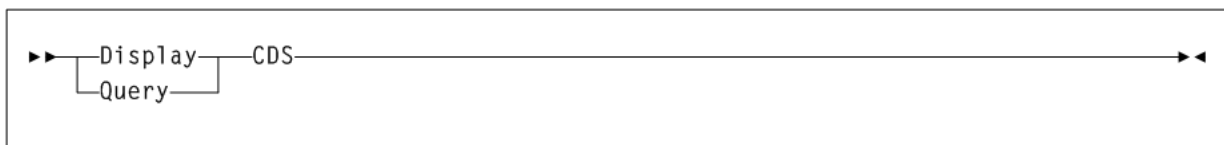
### Description

`Display CDS` displays the status of all copies of the CDS, primary and secondary log files, operational settings (for example, the cleaning cartridge prefix, SMF record type, etc.), and the status of all HSC-defined hosts.

### Syntax

Figure 3–35 shows `Display CDS` command syntax.

**Figure 3–35** *Display CDS command syntax*



### Parameters

None.



## Output

The following is an example of the output produced by the `Display CDS` command:

### **Example 3–11 Display CDS output**

```
SLS0000I D CDS
SLS1969I User exit 15 module SLSUX15 INACTIVE
SLS2716I Database Information 951

PRIMARY   = SEA710.SPRG1.DBASEPRM.LGEN8501
VOLSER    = HSC20J  ACTIVE
SECONDARY = SEA710.SPRG1.DBASESEC.LGEN8501
VOLSER    = HSC20G  ACTIVE

CDS LEVEL = 060100          VER = 16012446

CDS CREATE      = 20141203 13:49:56
LAST CDS BACKUP = NONE
LAST CDS RESTORE = NONE

CDS BLOCK COUNT = 7,200
CDS FREE BLOCKS = 4,135
ENQNAME         = STKSBADD      SMFTYPE = 245
CLEAN PREFIX    = CLN          LABTYPE = (00) SL

NO LOG FILE DSN(S) DEFINED

VAULT RECORDS NOT FOUND IN THIS CDS

CDKLOCK RECORDS NOT FOUND IN THIS CDS

LAST NCO ON ECCY
  START = 12/03/14 13:50:55
  END   = 12/03/14 13:53:19

HOSTID---LEVEL---DESCRIPTION-----
ECCL    -.-.- -INACTIVE-
ECCY    7.1.0  ACTIVE PRIMARY SECONDARY
EC20    -.-.- -INACTIVE-
EC21    -.-.- -INACTIVE-
```

## Display CLink

### **Interfaces:**

- Console or utility
- UUI Support: Yes

### **Subsystem Requirements:**

Active HSC/VTCS

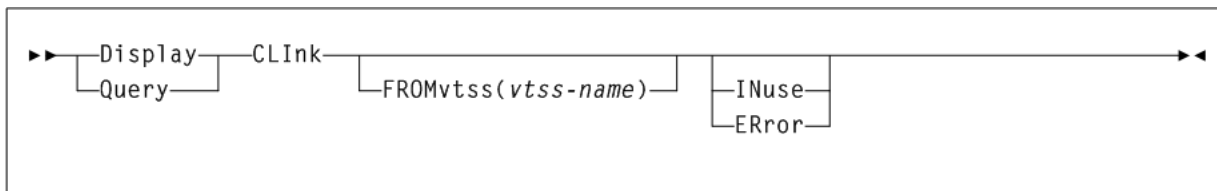
### **Description**

The `Display CLink` command displays Cluster link (CLINK) status. Optionally, you can display CLINK status at the VTSS level, or display only those CLINKs that are in use, or in an error state.

## Syntax

Figure 3–36 shows Display CLInk command syntax.

**Figure 3–36 Display CLink command syntax**



## Parameters

As shown in Figure 3–36, the Display CLink command includes the following parameters:

### FROMvtss(vtss-name)

optionally, specifies the name of the VTSS that you want to display CLINKs from.

### INuse

optionally, displays only CLINKs that are in use.

### ERror

optionally, displays only CLINKs that are in an error state.

---

**Note:** INuse and ERror are mutually exclusive.

---

## Output

The following is an example of the output produced by the Display CLink command with no parameters:

### Example 3–12 Display CLink output (no parameters)

```

VTSS   Clink  Name      Status  Usage   Host      Partner  VTD
HBVTSS16 07 1A:0 PLEXNAME Online  Free    REMOTEAA 7
          06 1E:0 PLEXNAME Online  Free    REMOTEBA 6
HBVTSS18 07 1A:0 -Cluster Online  Free    HBVTSS19 AB07
          06 1E:0 -Cluster Online  Free    HBVTSS19 AB06
*SLS5013I Command completed (0)
  
```

The following is an example of the output produced by the Display CLink command with FROM(DVTSS18) specified, to display output for CLINKs owned by VTSS DVTSS18.

### Example 3–13 Display CLink output (FROMvtss specified)

```

VTSS   Clink  Name      Status  Usage   Host      Partner  VTD
DVTSS18 08 C0I  SECNDARY On-Sync  Free    DVTSS17 #008
          09 C1I  SECNDARY On-Sync  Free    DVTSS17 #009
  
```

## Fields

Display CLink command output includes the following fields:

### VTSS

the Primary or Sending VTSS name.

**CLINK**

a composite of the device Id that has been assigned to the CLINK within the VTSS and the back end channel interface to which it is connected.

**NAME**

one of the following values:

**name**

the name of the TapePlex to which the CLINK is connected. This also indicates that the CLINK will be used for electronic exporting of VTVs. The VTSS name in the **Partner** column indicates the VTSS that will receive VTV copies over this CLINK.

**-Cluster**

indicates that the CLINK is used for replication of VTVs within the TapePlex and is part of a cluster. The VTSS name in the **Partner** column indicates the other VTSS that operates in the cluster.

**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.

**Online**

The link is available for replication.

**P\_offline**

The link is pending offline.

**P\_online**

The link is pending online.

**On-Sync**

Available for synchronous replication.

**On-Async**

Available for asynchronous replication.

**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.

**UI err**

This is a CLINK defined for electronic export and it has been unable to contact the remote VTCS. There should be messages in the HSC `JOBLOG` that indicate the reason for the problem. This could include problems with the definitions, the local SMC, or the remote HTTP server on the remote VTCS.

**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.

**PARTNER**

the secondary or receiving VTSS.

**VTD**

the address of the VTD on the partner VTSS that forms the other end point to the connection. For a cluster link, the MVS address of the VTD is reported. For an electronic export link, this is not possible as there is no access to the other TapePlex configuration. In this case, only the ordinal number of the VTV is reported.

**Display CLUster****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

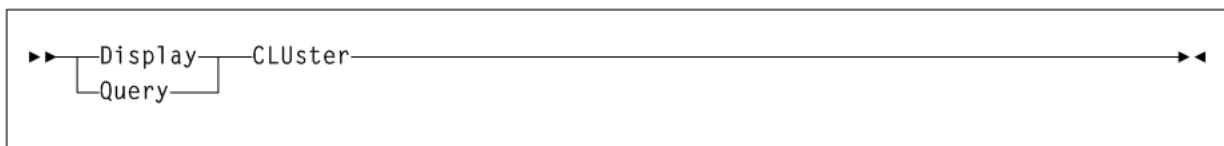
**Description**

The `Display CLUster` command displays Cluster status.

**Syntax**

Figure 3–37 shows `Display CLUster` command syntax.

**Figure 3–37** *Display CLUster command syntax*

**Parameters**

None.

**Output**

The following is an example of the output produced by the `Display CLUster` command:

**Example 3-14 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

**Fields**

Display CLUster command output includes the following fields:

**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 to 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-2ndary**

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.

**CLINKs offline**

All defined CLINKs are offline. No workload is possible on this Cluster.

**Only 2ndary**

The Secondary is online to VTCS and the Primary has no CLINKs online. Workload can run on the Secondary. VTVs requiring replication, however, are allocated to the Secondary only if no other Full Function Clusters are available.

**Only primary**

The Primary is online to VTCS and the Secondary has no CLINKs online. 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.

## Display CMD

Interfaces:

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at `BASE` or `FULL` service level

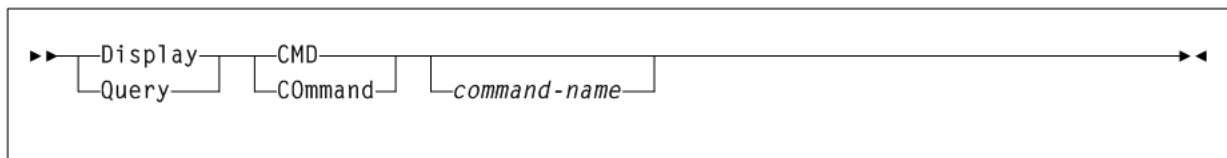
**Description**

The `Display CMD` command displays syntax and usage information for a VTCS or HSC command.

**Syntax**

Figure 3–38 shows `Display CMD` command syntax.

**Figure 3–38** *Display CMD command syntax*

**Parameters**

As shown in Figure 3–38, the `Display CMD` command includes the following parameters:

**CMD or COmmand**

the command name. If a `command-name` is not specified, a list of all HSC/VTCS commands is produced.

**command-name**

the command name. If a `command-name` is not specified, a list of all HSC/VTCS commands is produced.

For certain commands that accept multiple options, including `Display`, a two-part command may be entered. For example:

```
D CMD D VTV
```

This command displays help information only for the `Display VTV` command.

**Display COMMPath****Interfaces:**

- Console or `PARMLIB` only
- UUI Support: No

**Subsystem Requirements:**

Active HSC at `BASE` or `FULL` service level

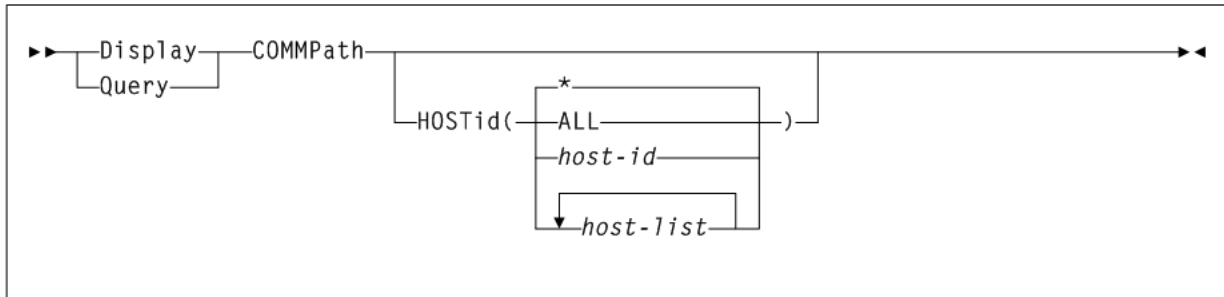
**Description**

The `Display COMMPath` command displays current settings for HSC host-to-host communications.

## Syntax

Figure 3–39 shows Display COMMPath command syntax.

**Figure 3–39 Display COMMPath command syntax**



## Parameters

As shown in Figure 3–39, the Display COMMPath command includes the following parameters:

### HOSTID

optionally, indicates that you want to obtain information about specific hosts.

#### ALL

displays the settings for all defined hosts. Depending on the number of hosts in your environment, the display can be extremely long.

#### host-id or host-list

the host or hosts about which you want information. The HSC issues console messages that display the current settings for each specified host.

The elements in a *host-list* must be separated by commas or blanks, and the entire list must be enclosed in parentheses. Ranges are not valid.

#### \*

displays the settings for the host on which you enter the command. This is the default if HOSTid is specified without a value.

## Display CONFIG

### Interfaces:

- Console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC at FULL service level

### Description

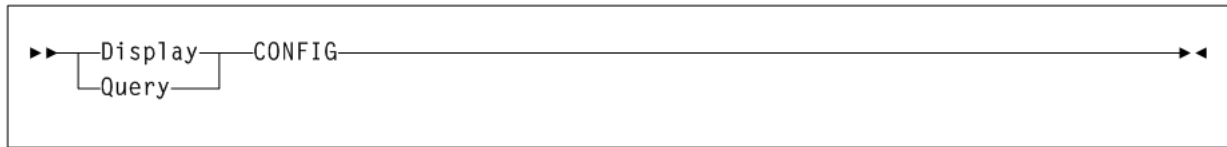
The Display CONFIG command displays CONFIG parameter settings.

## Syntax

Figure 3–40 shows Display CONFIG command syntax.



Figure 3-40 Display CONFIG command syntax



## Parameters

None.

## Output

The following is an example of the output produced by the Display CONFIG command:

### Example 3-15 Display CONFIG output

```
SLS6603I Configuration information
MaxVTV  MVCFree  VTVAttr  RECALWER  REPLICAT  VTPPAGE  SYNCHREP
4000    2          Scratch  Yes       Always    Standard No
```

```
MAXRTDs  FASTMIGR  INITMVC  MAXVTVSZ  FASTRECL  LOCKTOUT
32       Yes     Yes     4000     Yes       10
```

```
NLIBDRNR  NLIBMIGR  NLIBRECL  MVCMTTO  LOGPOL  VTDGT256
No        No      No       15      Optional Yes
```

This tapeplex: LOCALPLX

```
CDS level support:  V5/5.1  V6  V6.1  V6.2  V7  V7.1
                   *      *      *
```

```
Reclaim : Max MVC  Start  Commvc  Threshold: Tape VL
          4         10    1          30    70
```

```
Reclaim: MaxMVC Start Commvc Inplace  Threshold: Tape  VL  Inpthrsh
          4     10    1     No          45    30    15
```

```
          Auto Migr Thr  Migr Tasks  Default  VSM  2GB/  Page  RTDs
VTSSNAME Low High   Min  Max    ACS  Model 4GB  Size  Yes
DVTSS16  65  70    2   3    FF   5   Yes Large
```

```
Devno  RTD Type  Inplace  ACS    Retain  VTSSNAME  RTD NAME  CHANIF
2A00   36TRACK No      00     10     DVTSS16  SS162A00 00 0A:1
2A01   36TRACK No      TMVSB  10     DVTSS16  SS162A01 01 0A:0
2A02   36TRACK No      00     10     DVTSS16  SS162A02 02 0E:0
```

## Fields

Display CONFIG command output includes the following fields:

---



---

**Note:** See "CONFIG GLOBAL Statement" on page 3-42 for more information about CONFIG GLOBAL statement parameters.

---



---

## MaxVTV

the CONFIG GLOBAL MAXVTV setting.

**MVCFree**

the CONFIG GLOBAL MVCFREE setting.

**VTVAttr**

the CONFIG GLOBAL VTVAttr setting (SCRATCH or ALLmount).

**RECALWER**

the CONFIG GLOBAL RECALWER setting (YES or NO).

**REPlicat**

the CONFIG GLOBAL REPlicat setting (ALWAYS or CHANGED).

**VTVPAGE**

the CONFIG GLOBAL VTVPAGE setting The VTV page size (STANDARD or LARGE).

**SYNCHREP**

the CONFIG GLOBAL SYNCHREP setting (YES or NO).

**MAXRTDS**

the CONFIG GLOBAL MAXRTDS setting (16 or 32).

**FASTMIGR**

the CONFIG GLOBAL FASTMIGR setting (YES or NO).

**INITMVC**

the CONFIG GLOBAL INITMVC setting (YES or NO).

**MAXVTVSZ**

the CONFIG GLOBAL MAXVTVSZ setting (400, 800, 2000, 4000, or 32000).

**FASTRECL**

the CONFIG GLOBAL FASTRECL setting (YES or NO).

**LOCKTOUT**

the CONFIG GLOBAL LOCKTOUT setting (0 or 5-240).

**NLIBDRNR**

the CONFIG GLOBAL NLIBDRNR setting (YES or NO).

**NLIBMIGR**

the CONFIG GLOBAL NLIBMIGR setting (YES or NO).

**NLIBRECL**

the CONFIG GLOBAL NLIBRECL setting (YES or NO).

**MVCMNTTO**

the CONFIG GLOBAL MVCMNTTO setting (5-30 minutes).

**LOGPOL**

the CONFIG GLOBAL LOGPOL setting (OPTIONal or REQuired).

**VDGT256**

the CONFIG GLOBAL VTDGT256 setting (YES or NO).

**CDSLEVEL SUPPORT**

the VTCS level(s) that can access the active CDS.

**THRESHLD**

the RECLAIM THRESHLD setting.

**MAX MVC**

the RECLAIM MAXMVC setting.

**START**

the RECLAIM START setting.

**CONMVC**

the RECLAIM CONMVC setting.

**VTSSNAME**

the VTSS identifiers (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 or 4 GB VTV sizes (Y or N).

**PAGE SIZE**

VTV page size (STANDARD or LARGE).

**RTDs**

indicates whether the VTSS has RTDs.

**DEVNO**

RTD MVS device numbers for the VTSS (RTD DEVNO settings), or device numbers for a VLE, as generated by VTCS.

- Single-node VLE device numbers start with a "V".
- Multi-node VLE device numbers are prefixed with an "S".

**RTD TYPE**

the RTD type. This can include the VLE device type.

**Inplace**

the Storclas INPLACE setting.

**ACS**

- For local tape drives, this column displays the ACS or LSM to which the drive is attached.
- For remote RTDs, this column displays '*name: number*' where *name* indicates the name of the TapePlex and *number* indicates the number of the ACS in decimal.
- For VLEs, this column displays the VLE name.

**RETAIN**

the VTSS `RETAIN` setting.

**VTSSNAME**

the VTSS identifiers (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 DRives**

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at `FULL` service level

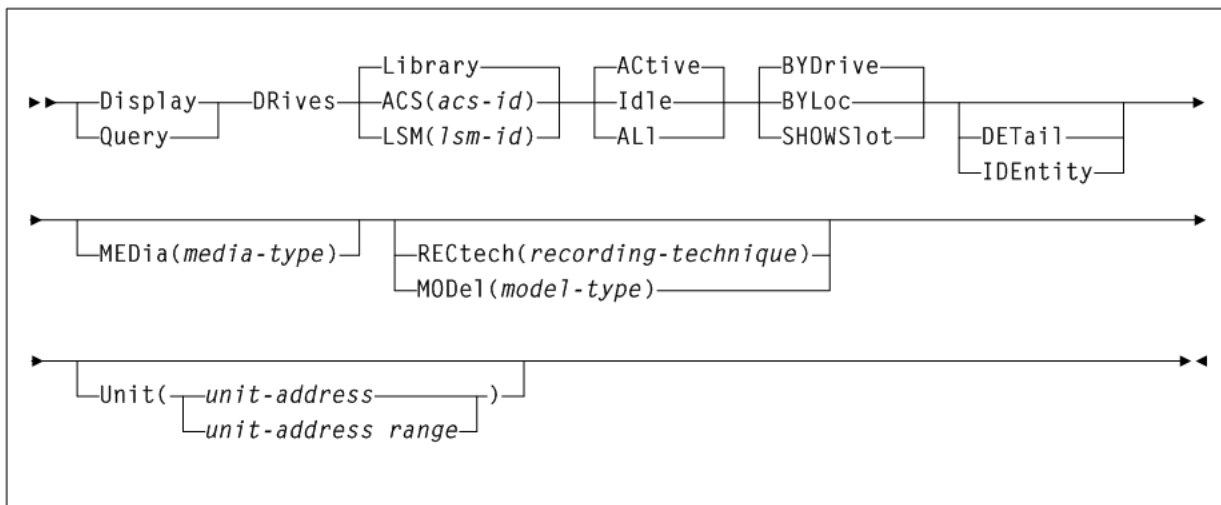
**Description**

The `Display DRives` command displays current and queued mount activity for each transport address.

**Syntax**

Figure 3–41 shows `Display DRives` command syntax.

**Figure 3–41 Display DRives command syntax**



## Parameters

As shown in [Figure 3-41](#), the `Display DDrives` command includes the following parameters:

### Library

only library-controlled drives are processed. This is the default.

### ACS(*acs-id*)

only drives in a specified ACS are processed.

*acs-id* indicates the ACS for which the system is to display drive information.

### LSM(*lsm-id*)

only drives in a specified LSM are processed.

*lsm-id* indicates the LSM for which the system is to display drive information.

### ACtive

only active drives are processed. This is the default.

### Idle

only idle drives are processed (includes drives that may be offline).

### ALI

all drives are processed regardless of status.

### BYDrive

displays the drives by host device address. This is the default.

### BYLoc

displays the drives by library location.

### SHOWSlot

optionally, displays the drives by host device address, and displays the drive bay location (slot) for drives in SL3000 and SL8500 libraries.

### DETail

optionally, displays the drive media types and recording techniques.

### IDEntity

optionally, displays the serial number and world wide name of a transport associated with a drive.

The tape library must be at a sufficient compatibility level to provide this information.

- The serial number requires LMU level 13.
- The WWN requires LMU level 21.
- `DISPLAY ACS` shows the LMU compatibility level. In addition, the transport must supply the LMU with the serial number.

---

---

**Note:** `IDEntity` and `DETail` are mutually exclusive.

---

---

### MEDia(media-type)

optionally, limits the display of drives only to those that support the specified type of media. The default is for all types of media.

*media-type* indicates the media type.

See [Appendix A, "MEDia, RECtech, and MODel Values"](#) for a list of valid *media-type* values.

---

---

**Note:** If *MEDia* is not specified, the drive is selected without regard to media type.

---

---

**RECtech(recording-technique)**

optionally, limits the display of drives only to those that support the specified recording technique. The default is for all recording techniques. *RECtech* refers to the method used to record data tracks on the tape surface.

If *RECtech* is not specified, the drive is selected without regard to recording technique.

*recording-technique* indicates the recording technique.

See [Appendix A, "MEDia, RECtech, and MODel Values"](#) for valid *recording-technique* values.

---

---

**Note:** *RECtech* and *MODel* are mutually exclusive.

---

---

**MODel(model-type)**

optionally, limits the display of drives by the model type of a transport. You can enter a list of models, separated by commas. A list specifies a generic pool.

*model-type* indicates the model type.

See [Appendix A, "MEDia, RECtech, and MODel Values"](#) for a list of valid *model-type* values.

---

---

**Note:** *MODel* and *RECtech* are mutually exclusive.

---

---

**UNIT(unit-address)**

optionally, limits the display of drives to a unit address or a range of unit addresses.

*unit-address* or *unit-address-range* indicates a single unit address or a range of addresses.

## Display DRIVE\_INFO

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

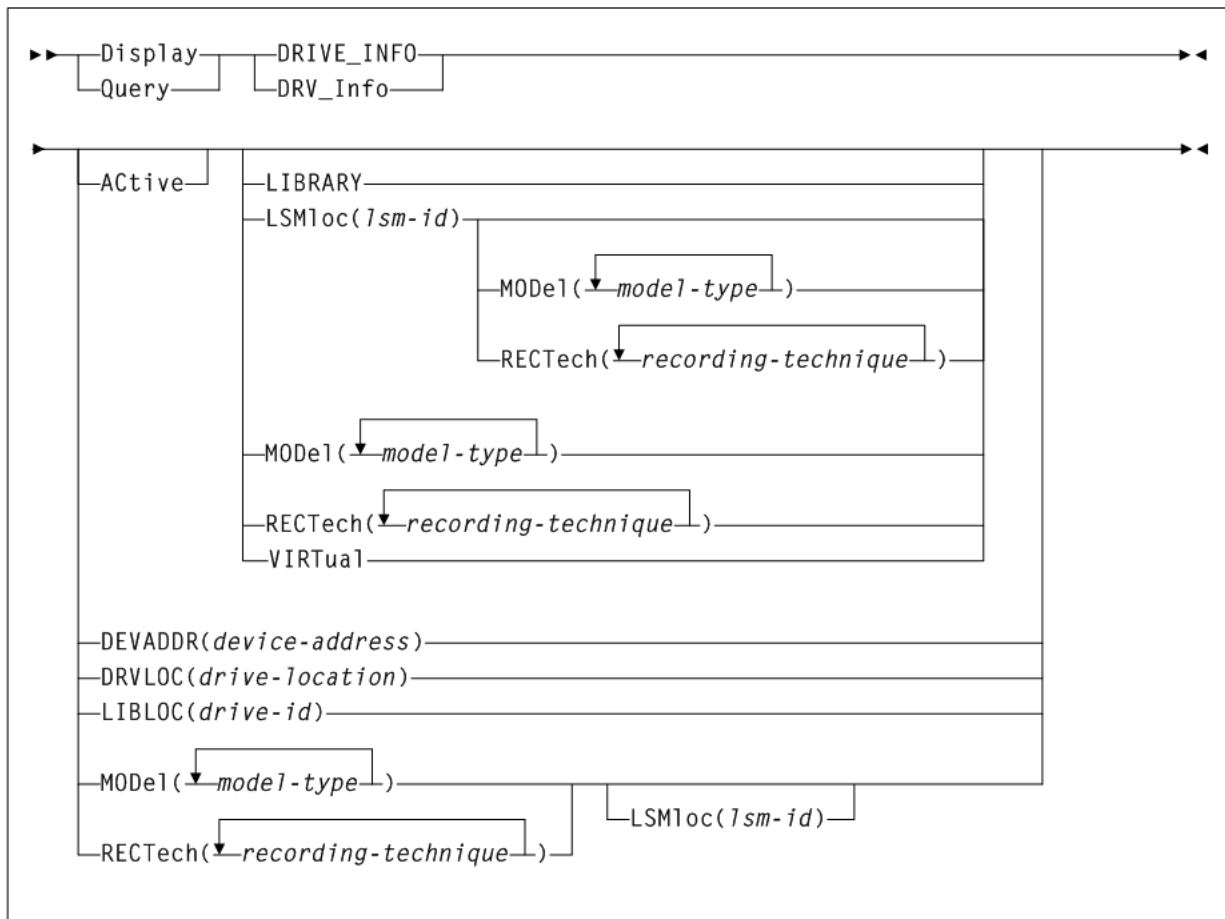
**Description**

The `Display DRIVE_INFO` displays information for real and/or virtual drives.

**Syntax**

[Figure 3–42](#) shows `Display DRIVE_INFO` command syntax.

Figure 3-42 Display DRIVE\_INFO command syntax



### Parameters

As shown in Figure 3-42, the Display DRIVE\_INFO command includes the following parameters:

#### Active

optionally, specifies that only drives with a volume mounted, or in the process of mounting or dismounting, are displayed.

#### DEVADDR(device-address)

optionally, specifies that only the device with the specified drive ID is displayed.

*device-address* indicates a three or four digit device address.

#### DRVLOC(drive-location)

optionally, specifies that only the single drive with a matching drive location ID is displayed.

*drive-location* indicates the drive location id:

For a real drive, use format *R:AA:LL:PP:DD* where:

- *AA* indicates the acs ID.
- *LL* indicates the LSM number.
- *PP* indicates the panel.

- *DD* indicates the drive number.

For a virtual drive, use format *V:vtssname:nnnn* where *nnnn* indicates the VTSS drive identifier.

**LIBLOC(drive-id)**

optionally, specifies that only the real drive with the specified drive ID is displayed.

*drive-id* indicates the drive ID, in the format *AA:LL:PP:DD* where:

- *AA* indicates the acs ID.
- *LL* indicates the LSM number.
- *PP* indicates the panel.
- *DD* indicates the drive number.

**LIBRARY**

optionally, specifies that only library (non-virtual) drives are displayed.

**LSMloc(lsm-id)**

optionally, specifies that only drives in the specified LSM is displayed.

*lsm-id* indicates the LSM ID, in the format *AA:LL* where:

- *AA* indicates the acs ID.
- *LL* indicates the LSM number.

**MODEl(model-type)**

optionally, specifies that only drives with the specified model type are displayed.

*model-type* indicates the model type.

See [Appendix A, "MEDia, RECTech, and MODEl Values"](#) for a list of valid *model-type* values.

**RECTech(recording-technique)**

optionally, specifies that only drives with the specified recording technique are displayed.

*recording-technique* indicates the recording technique.

See [Appendix A, "MEDia, RECTech, and MODEl Values"](#) for a list of valid *recording-technique* values.

**VIRTual**

optionally, specifies that only virtual drives (not library drives) are displayed.

## Display EXceptns

**Interfaces:**

- Console or PARMLIB only
- UUI Support: No

**Subsystem Requirements:**

Active HSC at FULL service level

**Description**

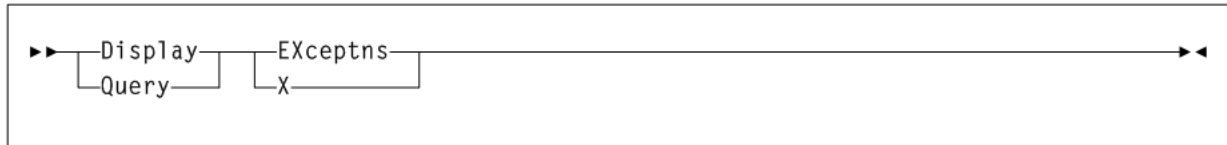
The `Display EXceptns` command displays hardware status.



## Syntax

Figure 3–43 shows Display EXceptns command syntax.

**Figure 3–43 Display EXceptns command syntax**



## Parameters

As shown in Figure 3–43, the Display EXceptns command includes the following parameters:

### EXceptns or X

displays hardware status.

Errors are reported in either LSM *AA:LL* or *AA:LL:CC* format where *AA* indicates the hexadecimal value for the ACS (00-FF), *LL* is hexadecimal value for the LSM (00-17), and *CC* indicates the hexadecimal value for the CAP identifier (00 through 0B).

## Output

Display EXceptns displays status for the following:

- all LMUs
- all LSMs
- all CAPs
- all robotic hands
- all pass-thru ports (PTPs)
- all stations

Errors are reported in *AA:LL:CC* format, where *AA* indicates the decimal value for the ACS (00-99), *LL* is decimal value for the LSM (00-99), and *CC* decimal value for the CAP identifier (00 through 11). The following error messages are possible:

**Table 3–2 Display EXceptns Error Messages**

Message	Description
CAP Not Operational	The CAP is not operational.
CAP Door is Open	The CAP door is open.
CAP Door is Unlocked	The CAP door is unlocked.
CAP is Reserved	The CAP is reserved.
Pass Thru Port is Inoperative	The pass thru port is inoperative.
Robot Hand is Inoperative	The robot hand is inoperative.
Robot Hand needs maintenance	The robot hand is in need of maintenance.
LSM is Not Ready	The LSM is in a not ready state.
LSM is Offline	The LSM is in an offline state.
LSM is Offline Pending	The LSM is in an online pending state.

**Table 3–2 (Cont.) Display EXceptns Error Messages**

Message	Description
LSM is in Maint Mode	The LSM is in maintenance mode.
LSM Door is Open	The LSM door is open.
LMU Compat 10 or less; not all functions supported	The LMU will not be able to respond correctly to some of the status queries until its microcode is updated. This condition will also cause SLS0662I LMU Response Error to be issued just before SLS4610I.
Station <i>nn</i> has Inactive Connection:	A hardware connection to the LMU is not usable. This may be normal for your configuration if the connection has never been made and is not necessary. The station number is in hex, so Station 0A is the tenth station, and Station 10 is really the sixteenth.
Station <i>nn</i> Inoperative; Host not responding:	The station number is in hex, so Station 0A is the tenth station, and Station 10 is really the sixteenth.

The following messages are summaries:

- No CAP problems were detected
- No Pass Thru Port problems were detected
- No Robot Hand problems were detected
- No LSM problems were detected
- No Station problems were detected

Message summaries only reflect that the LMU did not detect hardware errors. Something could be wrong with the software configuration, or with something that the LMU could not detect.

HSC processing continues and no user response is required.

## Display LINKSto

### Interfaces:

- Console or PARMLIB
- UUI Support: No

### Subsystem Requirements:

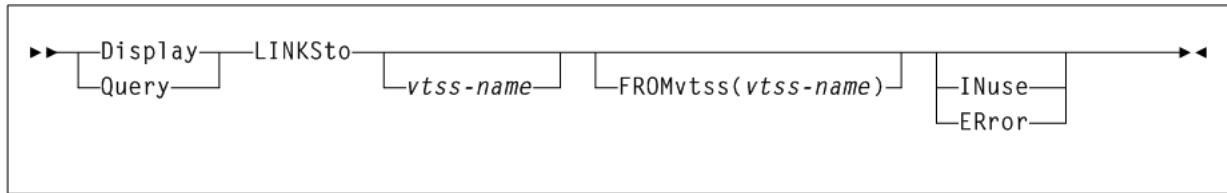
Active HSC at BASE or FULL service level

### Description

The Display LINKSto command displays replication link and CLINK status information.

### Syntax

Figure 3–44 shows Display LINKSto command syntax.

**Figure 3–44 Display LINKSto command syntax****Parameters**

As shown in [Figure 3–44](#), the Display LINKSto command includes the following parameters:

**vtss-name**

optionally, the name of the replication link for which status is displayed. This name represents a link to a target VTSS subsystem (*vtss-name*). If this parameter is not specified, then all targets are displayed.

**FROMvtss(vtss-name)**

optionally, specifies the *vtss-name* that owns, or is the source VTSS, for any specified replication link name. If this parameter is not specified, then all source VTSSs for the LINK name are displayed.

**INuse**

optionally, displays only LINKs that are in use.

**ERror**

optionally, displays only LINKs in an error state.

**Output**

The following is an example of the output produced by the Display LINKSto command.

This example displays status information for all links to target DVTSS18. This includes RLINK DVTSS18 status and CLINK status information of the owning (or source) VTSS, any active or pending enhanced replications for the link, and a list of all configured connections for the link. There is no channel interface information output for the source VTSS.

**Example 3–16 Display LINKSto output**

```

D LINKSTO DVTSS18

/SLS6603I Linksto information: 702
To VTSS Location VTD From Connect Status Usage Host
DVTSS18 -Cluster ---- DVTSS16 Links On: 1 Free
          AA0A 0A C0K On-Sync Free
          AA0B 0B C1K On-Sync Free
  
```

**Fields**

Display LINKSto command output includes the following fields:

**To VTSS**

The name of the replication link. This represents the link to a target VTSS, ownership of which belongs to the source VTSS.

**Location**

One of the following values:

**-Cluster**

indicates that the link is used for replication within a TapePlex.

**tp\_name**

The name of the tapeplex to which the link is connected. This indicates that the link will be used for Cross-TapePlex Replication operations.

**VTD**

One of the following values:

**v\_addr**

The address of the VTD on the target VTSS. This indicates that CLINKs connect the local VTSS's. Replication can be over CLINKs or replication links.

----

Indicates that connectivity is over replication links.

**From**

The name of the source VTSS. This is the VTSS that owns the replication link to the target VTSS.

**Connect**

One of the following connection types/values:

**Link**

indicates a replication link connection.

**vid/cif**

A composite of the device id that is assigned to the link and the channel interface.

**Status**

One of the following values, followed by the number of connections that apply.

**Online**

The link is online and available for replication.

**Offline**

The link is offline and unavailable for replication.

**On-Sync**

The link is online and available for synchronous replication.

**Off-Fail**

The link is in an offline-failed state.

**Fvts-Off**

The from, or source, VTSS is offline.

**Usage**

One of the following connection values:

**Free**

The link is idle.

**Assigned**

The link is assigned to a host, not replicating.

**Replicating**

The link is actively replicating.

**Host**

The name of the host that the link is assigned to.

**Display LMUPDEF****Interfaces:**

- Console or PARMLIB only
- UUI Support: No

**Subsystem Requirements:**

Active HSC at BASE or FULL service level

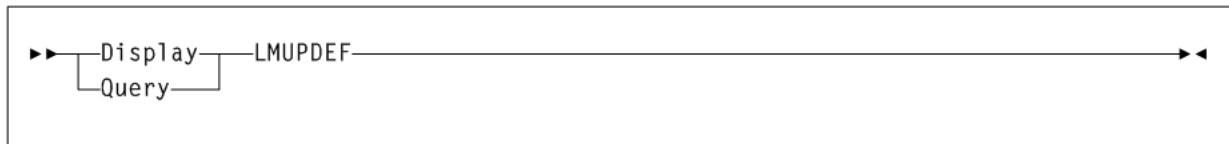
**Description**

The Display LMUPDEF command displays information about the definition data set containing LMUPATH statements, that is currently loaded.

**Syntax**

Figure 3–45 shows Display LMUPDEF command syntax.

**Figure 3–45** Display LMUPDEF command syntax

**Parameters**

None.

**Display LOCKs****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

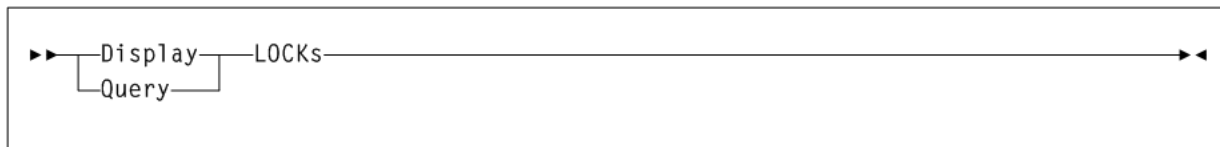
Active HSC/VTCS

**Description**

The Display LOCKs command displays VTCS lock status.

**Syntax**

Figure 3–46 shows Display LOCKs command syntax.

**Figure 3–46 Display LOCKS command syntax****Parameters**

None.

**Output**

The following is an example of the output produced by the Display LOCKS command:

**Example 3–17 Display LOCKS output**

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

**Fields**

Display LOCKS command output includes the following fields:

**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 if 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 Lsm

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at FULL service level

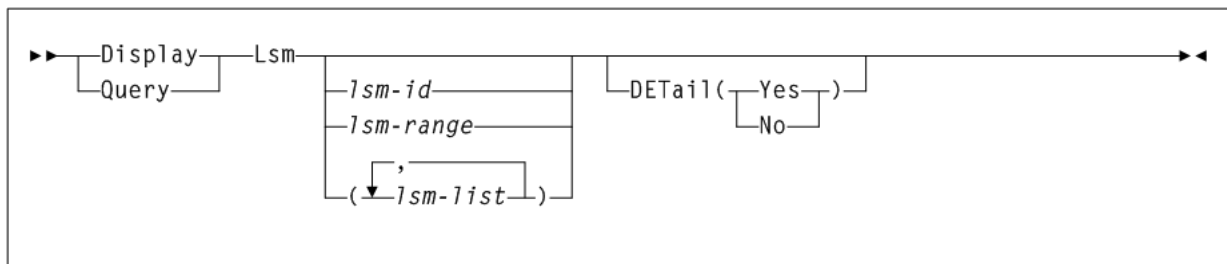
**Description**

The `Display Lsm` command displays LSM status.

**Syntax**

Figure 3-47 shows `Display Lsm` command syntax.

**Figure 3-47** *Display Lsm command syntax*

**Parameters**

As shown in Figure 3-47, the `Display Lsm` command includes the following parameters:

**lsm-id, lsm-range, or lsm-list**

optionally, one or more LSMs for which the system is to display information. If you do not supply an *lsm-id*, a status summary is displayed for every LSM in the library.

Each *lsm-list* element can be either a single LSM ID or a range of LSM IDs. The elements in a list must be separated by commas or blanks, and the entire list must be enclosed in parentheses.

An *lsm-id* format is *AA:LL*, where *AA* indicates the ACSid (decimal 00-99) and *LL* indicates the LSM number (decimal 00-99).

**DETail**

optionally, specifies whether detail information is displayed for each LSM.

**Yes**

All detailed information is displayed for each LSM.

**No**

Only summary information is displayed for each LSM.

When DETail (No) is specified, the current Ready or Not ready status is not determined; the LSM shows as Ready unless it is offline.

---

---

**Note:** If the DETail parameter is not specified, then detailed information is displayed only if specific LSM IDs are entered.

---

---

**Output**

The `Display Lsm` command displays status for the following:

- LSMid
- LSM type
- Online/offline status
- Ready/not ready status
- Automatic/manual status
- May include detail information:
  - Audits in progress (if any)
  - CAP status, priority and owning host if CAP is not drained.
  - Number of free cells and scratch volumes. (Note: The number of free cells in the LSM does not include free cells on frozen panels.)
  - Each frozen panel, showing the total number of cells and number of free cells on the panel.

**Display Message****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at BASE or FULL service level



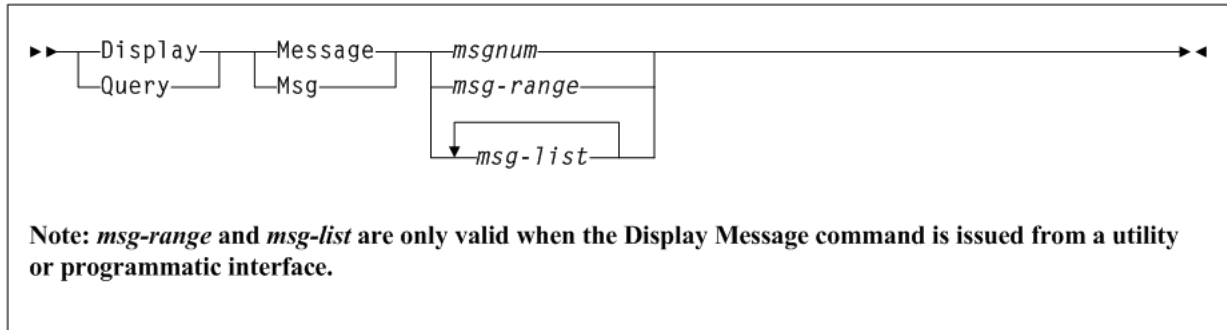
## Description

The `Display Message` command displays detailed HSC or VTCS message information.

## Syntax

Figure 3–48 shows `Display Message` command syntax.

**Figure 3–48** *Display Message command syntax*



## Parameters

As shown in Figure 3–30, the `Display Message` command includes the following parameters:

### Msg or Message

display detailed HSC or VTCS message information.

### msgnum

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

### msg-range or msg-list

a range or list of messages for which the system is to display status. Each `msg-list` element can be either a single `msgnum` or a `msg-range`. The elements in a list must be separated by commas or blanks, and the entire list must be enclosed in parentheses.

---

**Note:** `msg-range` and `msg-list` are only valid when the `Display Message` command is issued from a utility or programmatic interface.

---

## Display MGMTDEF

### Interfaces:

- Console or PARMLIB only
- UI Support: No

### Subsystem Requirements:

Active HSC at BASE or FULL service level

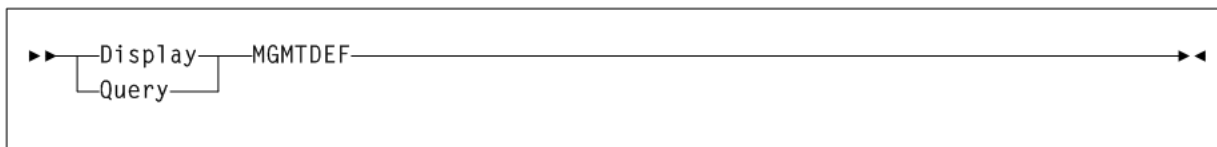
## Description

The `Display MGMTDEF` command displays information about active `MGMTclas` control statements.

### Syntax

Figure 3–49 shows Display MGMTDEF command syntax.

Figure 3–49 Display MGMTDEF command syntax



### Parameters

None.

## Display MIGrate

### Interfaces:

- Console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC/VTCS

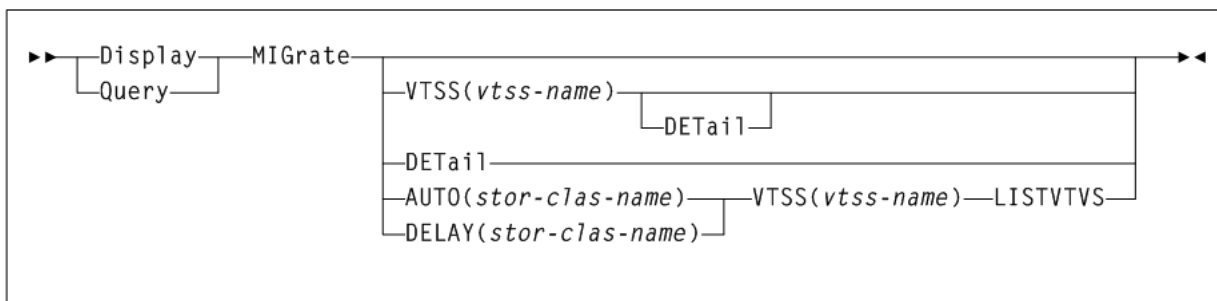
### Description

The Display MIGrate command displays migration status.

### Syntax

Figure 3–50 shows Display MIGrate command syntax.

Figure 3–50 Display MIGrate command syntax



### Parameters

As shown in Figure 3–50, the Display MIGrate command includes the following parameters:

#### VTSS(vtss-name)

Optionally, display migration status for a specified VTSS.

*vtss-name* indicates the VTSS name.

#### DETAil

Optionally, display detailed migration status by Storage Class.

**AUTO(stor-clas-name)**

Optionally, display the number of VTVs for the specified storage class, and their time remaining on the *AUTO* queue. The time remaining is referred to as the age.

*stor-clas-name* indicates the Storage Class name.

**DELAY(stor-clas-name)**

Optionally, display the number of VTVs for the specified storage class, and their time remaining on the *DELAY* queue. The time remaining is referred to as the age.

*stor-clas-name* indicates the Storage Class name.

**VTSS(vtss-name)**

Display migration status for a specified VTSS (*vtss-name*).

**LISTVTVS**

List the specific VTVs on the *AUTO* queue. *VTSS* must also be specified.

**Output (Display Migrate)**

The following is an example of the output produced by the *Display MIGrate* command:

**Example 3–18 Display MIGrate output**

```
/SLS0000I D MIG VTSS(HBVTSS16)
VTSSNAME: HBVTSS16
Active migration tasks: 4
Immediate migrate: Max wait: 5 MINUTES
Immediate delay queue: Active
Auto migrate: Host: EC20 MIGRATION TARGET: 70%
```

**Fields**

*Display MIGrate* command output includes the following fields:

**VTSSNAME**

the VTSS migrating the VTVs.

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

**IMMEDIATE DELAY QUEUE**

indicates whether there are VTVs on the Immediate Migrate Queue.

**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.

**Output (Display Migrate DETail)**

The following is an example of the output produced by the *Display MIGrate DETail* command:

**Example 3–19 Display MIGrate DETail output**

```

/SLS0000I D MIG DET
VTSSNAME: DVTSS16
Active migration tasks: 2
Immediate migrate: Max wait: 1 minutes
Immediate delay queue: Not active
Auto migrate: Not active

```

Storage Class	ACS	Max/Onl RTDs	---SCH-- Lim Pref	Req Act	--Auto-- GB	---Immed--- Delay	GB	Weight /Skip
S1	00	16 16	1 0	1	-	1	9	50/ 0
S2	00	16 16	1 0	1	-	1	9	50/ 0

**Fields**

Display MIGrate command output includes the following fields:

**STORAGE CLASS**

the Storage Class associated with the migration.

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. For migrations to VLEs, this is the VLE name. If this column indicates **\*\*ANY\*\***, then migrations to any location are allowed, subject to any other restrictions.

**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.

**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.

### Output (Display MIGrate AUTO)

The following is an example of the output produced by the `Display MIGrate AUTO` command:

#### Example 3–20 Display MIGrate AUTO output

```
/SLS0000I D MIG AUTO(S1) VTSS(DVTSS16) LISTVTVS

/SLS6603I VTVs awaiting migration:

DX0601 DX0920

2 volumes with an age range of 3-5 in the queue
```

### Output (Display MIGrate DELAY)

The following is an example of the output produced by the `Display MIGrate DELAY` command:

#### Example 3–21 Display MIGrate DELAY output

```
/SLS0000I D MIG DELAY(S2) VTSS(DVTSS16) LISTVTVS

/SLS6603I VTVs awaiting migration:

DX0800 DX0900

2 volumes with an age range of 3-5 in the queue
```

## Display MNTD

### Interfaces:

- Console or PARMLIB only
- UUI Support: No

### Subsystem Requirements:

Active HSC at BASE or FULL service level

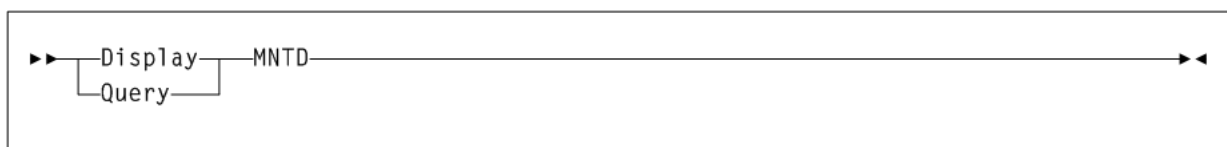
### Description

The `Display MNTD` command displays current settings for all HSC mount options set by the `MNTD` command.

### Syntax

Figure 3–51 shows `Display MNTD` command syntax.

Figure 3–51 Display MNTD command syntax



**Parameters**

None.

**Display MONitor****Interfaces:**

- Console or PARMLIB only
- UUI Support: No

**Subsystem Requirements:**

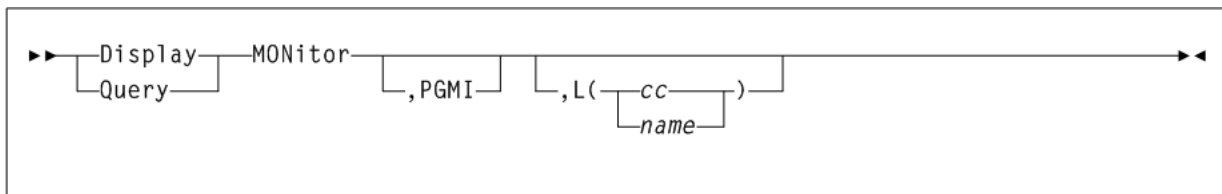
Active HSC at BASE or FULL service level

**Description**

The Display MONitor command displays a list of the monitoring consoles.

**Syntax**

Figure 3–52 shows Display MONitor command syntax.

**Figure 3–52 Display MONitor command syntax****Parameters**

As shown in Figure 3–52, the Display MONitor command includes the following parameters:

**,PGMI**

optionally, displays the status of the monitoring of move requests received by the programmatic interface.

**,L**

optionally, displays the type of requests being monitored by the specified console. Console IDs and console names must be two to eight characters long.

**cc**

the console ID.

**name**

the console name.

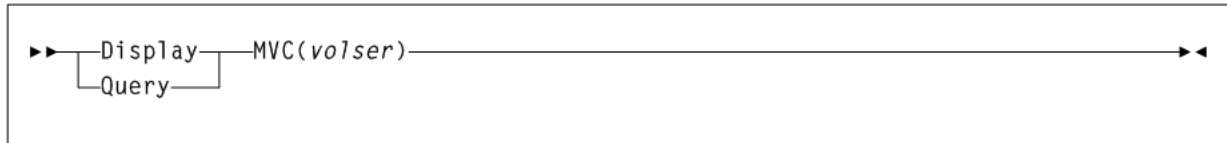
If both PGMI and L are omitted, the status of all monitoring is displayed.

**Display MVC****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

**Description**The `Display MVC` command displays information about a specific MVC.**Syntax**Figure 3–53 shows `Display MVC` command syntax.**Figure 3–53** *Display MVC command syntax***Parameters**As shown in Figure 3–53, the `Display MVC` command includes the following parameters:**volser**

the volser of the MVC.

**Output**The following is an example of the output produced by the `Display MVC` command:**Example 3–22** *Display MVC output*

```

Volser:                702532
Media:                 T1B000T1
ACSid:                 00
Size(Mb):              1000000
Migrated Count:       19
Vtv Count:             19
%Used:                 7.41
%Fragmented:          0.00
%Available:            92.59
%Usable:                0.00
Times Mounted:         77
Last Mounted:          2010Mar15 10:40:54
Last Migration:        2010Jan14 14:41:04
Last Drain/Reclaim:   2010Feb23 13:55:26
Owner:                 ALP
VTSS:                  HBVTSSY
MVCPOOL:               ALPMVC
Security access:       No profile
Status:                Initialized
Format:                Partitioned

```

**Fields**The `Display MVC` command output includes the following fields:

**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 containing the MVC, or the VLE containing remote MVC volumes or VMVCs in VLEs. This will not appear if the MVC has been ejected from the library. **\*\*MANY\*\*** indicates that an MVC with a specific volser has been discovered in more than one location. MVCs in this state are marked unusable.

**SIZE (MB)**

the size of the MVC in megabytes.

**MIGRATED COUNT**

the number of VTVs migrated.

**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 VERIFIED**

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

This date reflects the last time that VTCS knew the MVC contents were valid where the date is the most recent of the following:

- The date/time the MVC was verified with MEDVER.fy.



- The date/time the MVC was migrated to from an empty state.

**LAST DRAIN/RECLAIM**

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

**OWNER**

the Storage Class that owns the MVC. An 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.

---

---

**WARNING:** If the error occurred on a VOLSAFE (WORM) tape, do not drain the MVC without the recommendation of Oracle/StorageTek Support. Once a WORM tape is drained, it can no longer be used for migration and must be replaced.

---

---

- If the RTD caused the problem, use the `MVCMMAINT` utility to reset the MVC state.

---

---

**Note:** One or more of the following messages is issued for `IN ERROR` status: `SLS6686`, `SLS6687`, `SLS6688`, `SLS6690`. Refer to Oracle's ELS publication *ELS Messages and Codes* for recovery procedures for these messages.

---

---

### 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 an `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 `MVCMMAINT` 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.

---

---

**WARNING:** If the error occurred on a VOLSAFE (WORM) tape, do not drain the MVC without the recommendation of Oracle/StorageTek Support. Once a WORM tape is drained, it can no longer be used for migration and must be replaced.

---

---

- 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.

---

---

**WARNING:** If the error occurred on a **VOLSAFE (WORM) tape**, do not drain the MVC without the recommendation of Oracle/StorageTek Support. Once a WORM tape is drained, it can no longer be used for migration and must be replaced.

---

---

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 have the appropriate SAF rules set to enable VTCS to update the MVC. Correct the error and use the `MVCMaint` utility to set `READONLY (OFF)`. Refer to Oracle's ELS publication *Installing ELS* for information about defining a security subsystem user ID for HSC, SMC, and VTCS.

#### **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.

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, read the MVC to the end of the tape, 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.

---

---

**WARNING:** If the error occurred on a VOLSAFE (WORM) tape, do not drain the MVC without the recommendation of Oracle/StorageTek Support. Once a WORM tape is drained, it can no longer be used for migration and must be replaced.

---

---

#### **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.

#### **DEDUP requested on last mount**

DEDUP (YES) was requested on the last mount.

#### **FORMAT**

the VTV format, standard or partitioned.

## **Display MVCPool**

#### **Interfaces:**

- Console or utility
- UI Support: Yes

#### **Subsystem Requirements:**

Active HSC at FULL service level

#### **Description**

Display `MVCPool` displays information about a specific MVC pool.

---

---

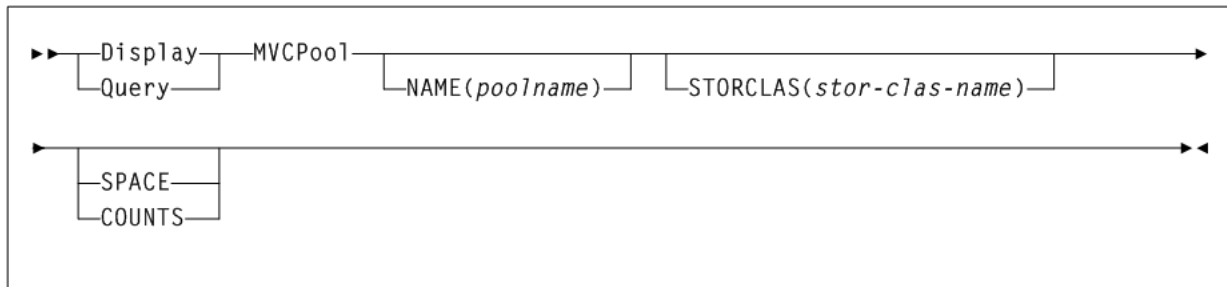
#### **Note:**

- When this command is issued, MVC location information is refreshed to ensure an accurate, up to date result.
  - MVCs in the following states are excluded from the MVC pool totals: EJECT, READONLY, DATACHECK, and Logically Deleted.
- 
-

## Syntax

Figure 3–54 shows Display MVCPool command syntax.

**Figure 3–54** Display MVCPool command syntax



## Parameters

As shown in Figure 3–54, the Display MVCPool command includes the following parameters:

### NAME(pool-name)

display information for the specified named MVC pool.

*pool-name* indicates the MVC pool name as defined using the MVCPool control statement. Refer to Oracle's ELS publication *ELS Legacy Interfaces Reference* for more information.

Alternatively, you can specify ALL to display information for all named MVC pools (including the default pool DEFAULTPOOL).

### STORCLAS(stor-clas-name)

display information about MVCs by Storage Class.

*stor-clas-name* indicates the name of a specific Storage Class for which you want MVC usage displayed.

This may be for current valid Storage Classes that you defined on the STORCLAS control statement, or for Storage Classes which used to exist to which media is still assigned. See "STORCLAS Control Statement" on page 3-238 for more information.

To display all Storage Classes, specify ALL or omit the STORCLAS parameter.

### SPACE

optionally, display information about MVCs using the SPACE output format. The SPACE parameter is ignored when the D MVCPool is from a utility.

### COUNTS

optionally, display information about MVCs using the COUNTS output format. The COUNTS parameter is ignored when the D MVCPool is from a utility.

---

**Note:** The Display MVCPool command from a utility will produce the combined SPACE or COUNTS output.

---

## Output

The following is an example of the output produced by the Display MVCPool command with pool name POOL1 specified:

**Example 3–23 Console Display MVCPool output (NAME POOL1)**

MVCPOOL (POOL1) INFORMATION

INITMVC NO	MVCFREE	THRESHOLD	INPTHRSH	MAXMVC	START	RECLAIM
2		30	43	4	10	DEMAND
ACS/ Location	Media	FREE-MVCS VOLS GB		RECLAIM-MVCS VOLS	GB	USED-MVCS VOLS GB
00	ECART	120	96	2	0.5	90 45
TMVSB	STK1R	30	600	1	3.5	25 350
00	TOTAL	150	696	3	4.0	115 395

The following is an example of the output produced by the Display MVCPool command with no pool name specified:

**Example 3–24 Console Display MVCPool output (no NAME specified)**

MVCPOOL INFORMATION

MVCPOOL (POOL1) INFORMATION

INITMVC NO	MVCFREE	THRESHOLD	INPTHRSH	MAXMVC	START	RECLAIM
2		30	43	4	10	DEMAND
ACS/ Location	Media	FREE-MVCS VOLS GB		RECLAIM-MVCS VOLS	GB	USED-MVCS 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
01	ECART	90	144	15	6.2	322 485
01	ZCART	35	700	3	11.3	43 675
01	TOTAL	125	844	18	17.5	365 1160
NON-LIB	STK2P	22	1100	0	0	12 1565
NON-LIB	TOTAL	22	1100	0	0	12 1565

The following figure shows an example of the output produced by the Display MVCPool SPACE console command with no pool name specified:

**Example 3–25 Console Display MVCPool SPACE output (no NAME specified)**

ACS/ Location	Media	Available Gb	Used Gb	Fragmented Gb	Unavailable Gb
00	T1C000T2	1,081,197	1,050,309	646,823	35,551
00	***Total	1,081,197	1,050,309	646,823	35,551
01	T1C000T2	1,167,239	1,046,045	604,593	581
01	***Total	1,167,239	1,046,045	604,593	581

The following figure shows an example of the output produced by the Display MVCPool COUNTS console command with no pool name specified:

**Example 3–26 Console Display MVCPool COUNTS output (no NAME specified)**

ACS/ Location	Media	Total MVC Count	Free MVC Count
00	T1C000T2	621	106
00	***Total	621	106
01	T1C000T2	618	126
01	***Total	618	126

**Example 3–27 Utility Display MVCPool output (no NAME specified)**

ACS/ Location	Media	Total MVC Count	Free MVC Count	Available GB	Used GB	Fragmented GB	Unavailable GB
00	T1C000T2	621	106	1,081,197	1,050,309	646,823	35,551
00	***Total	621	106	1,081,197	1,050,309	646,823	25,551
01	T1C000T2	618	126	1,167,239	1,046,045	604,593	581
01	***Total	618	126	1,167,239	1,046,045	604,593	581

**Fields**

Display MVCPool command output includes the following fields:

**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.

**INPTHRSH**

the fragmented space threshold (as a percentage) that determines when an MVC in partitioned format is eligible for dynamic reclaim processing.

**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.

**RECLAIM**

specifies the space reclamation setting for the MVC pool.

**ACS**

- For local tape drives, this column displays the ACS or LSM to which the drive is attached.

- For remote RTDs, this column displays *name: number* where *name* indicates the name of the TapePlex and *number* indicates the number of the ACS in decimal.
- For VLEs, this column displays the VLE name.
- NLIB indicates MVCs that are now outside the library.
- **\*\*MANY\*\*** indicates that an MVC with a specific volser has been discovered in more than one location. MVCs in this state are marked unusable.
- ACSs with no attached RTDs are flagged.

**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.

**Total MVC**

a count of the number of MVCs in the MVC pool.

**Free MVC**

a count of the number of FREE MVCs in the MVC pool. Free MVCs have 100% usable space.

**Available**

the size (Gb) of available MVC size in the MVC pool. This is the space that is available for migration.

**Used**

the size (Gb) of space used in the MVC pool.

**Fragmented**

the size (Gb) of wasted space in the MVC pool.

**Unavailable**

the size (Gb) of unavailable size in the MVC pool. This space is not available for migration. MVCs with the following states are included in this total:

- DATACHECK
- FULL
- BROKE
- READONLY
- RETIRED
- Warranty Expired



## Display OPTion

### Interfaces:

- Console or PARMLIB only
- UUI Support: No

### Subsystem Requirements:

Active HSC at BASE or FULL service level

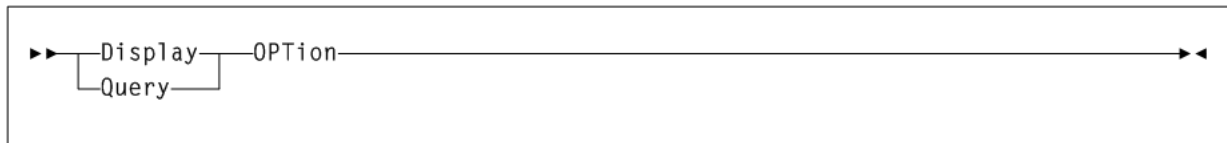
### Description

The Display OPTion command displays current settings for general HSC options set by the OPTion command.

### Syntax

Figure 3–55 shows Display OPTion command syntax.

**Figure 3–55 Display OPTion command syntax**



### Parameters

None.

## Display PATH

### Interfaces:

- Console or PARMLIB only
- UUI Support: No

### Subsystem Requirements:

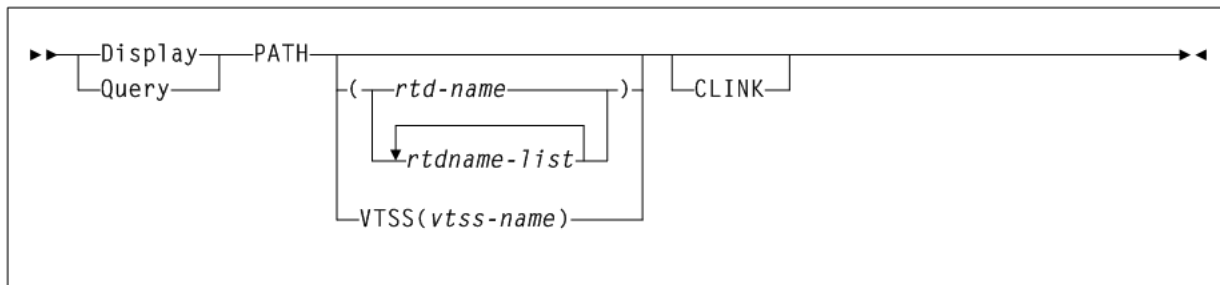
Active HSC/VTCS

### Description

The Display PATH command displays the status of paths from VTSSs to either RTDs or virtual libraries.

### Syntax

Figure 3–56 shows Display PATH command syntax.

**Figure 3–56 Display PATH command syntax****Parameters**

As shown in [Figure 3–56](#), the `Display PATH` command includes the following parameters:

**PATH**

optionally, restricts the display to the specified paths.

*rtdname* or *rtdname-list* indicates the names assigned to the paths for accessing RTDs for Virtual Libraries from a VTSS.

**VTSS**

optionally, restricts the display to the paths from an individual VTSS.

*vtss-name* indicates the name of the VTSS to which the paths are configured.

**CLINK**

optionally, include paths allocated for CLINKs in the display output. Paths for both cluster replication and electronic export are displayed.

**Output**

The following is an example of the output produced by the `Display PATH` command:

**Example 3–28 Display PATH output**

VTSSNAME	Type	RTD NAME	CHANIF	Device	Target	State
HBVTSS18	RTD	SS182A01	01 C0E	2A01		Online
	RTD	SS182A06	00 C0I	2A06		Online
	CLINK	HBVTSS19	06 C1A		CLUSTER1	On-Async
	VLIB	SS18IP0	04 I0A:0	V000	WILDCAT1	Online
	VLIB	SS18IP2	04 I1A:0		WILDCAT2	Online
HBVTSS19	EEXP	HBVTSS16	05 I0I:0		PRIMARY	On-Sync
	RTD	SS192A02	01 C0E	2A02		Online
	RTD	SS192A07	00 C0I	2A07		Online
	VLIB	SS19IP0	04 I0A:0	V010	WILDCAT1	Online
	VLIB	SS19IP2	04 I1A:0		WILDCAT2	Online
	EEXP	HBVTSS16	05 I0I:0		PRIMARY	On-Sync

**Fields**

`Display PATH` command output includes the following fields:

**VTSSNAME**

the name of the VTSS to which the path is attached.

**Type**

The type of the path. This can be one of the following:

- RTD – A direct connection to a RTD
- VLIB – A data path connection to a Virtual Library
- CLINK - A CLINK used for cluster replications within the tapeplex
- EEXP – A CLINK used for electronic export to another tapeplex

**RTDNAME**

The name allocated to the path. For paths defined via the RTDPATH statement in the configuration, this is the name assigned through the RTDNAME parameter. For a CLINK or EEXP type path, this column will contain the target VTSS name.

**CHANIF**

the logical device ID assigned to the path and the CHANIF or IPIF value that was specified for the path.

**Target**

the destination subsystem for the path. For a VLIB type path, this contains the STORMNGR name assigned to the Virtual Library. For a CLINK type path, this contains the name of the cluster. For the EEXP type paths, this contains the tapeplex name to which the path is connected.

**Device**

the destination device for the path. For an RTD type path, this contains the MVS address of the tape device. If this is a VLIB path that is working in compatibility mode, this will contain the logical device address assigned to the path.

**State**

The current state of this path. This can be one of the following:

**Online**

the path is online and available for use.

**On-Async**

the path is online and available for use for asynchronous replication of VTVs.

**On-Sync**

the path is online and available for use for synchronous replication of VTVs.

**Offline**

the path or the device to which it connects has been varied offline.

**Maint**

the path or the device to which it connects has been varied into an offline maintenance mode. This could be as a result of repeated failures.

**Display Queue****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

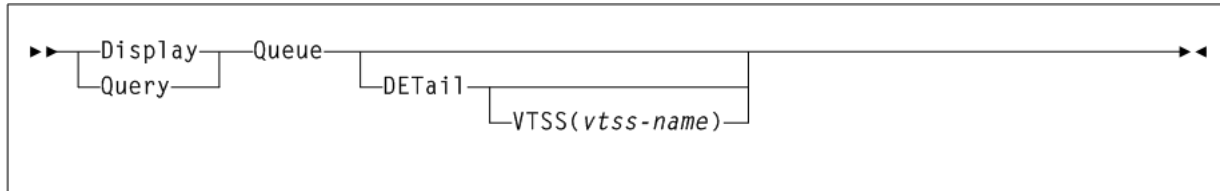
**Description**

The `Display Queue` command displays status of queued VTCS processes.

**Syntax**

Figure 3–57 shows `Display Queue` command syntax.

**Figure 3–57** *Display Queue command syntax*

**Parameters**

As shown in Figure 3–57, the `Display Queue` command includes the following parameters:

**DETail**

optionally, display detailed status.

**VTSS(vtss-name)**

optionally, display processes for the specified VTSS.

*vtss-name* indicates the VTSS name.

**Output**

The following is an example of the output produced by the `Display Queue` command:

**Example 3–29** *Display Active output (no detail)*

```
MIGRATES=3 RECALLS=2 RECLAIMS=0
```

**Display REPlicat****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

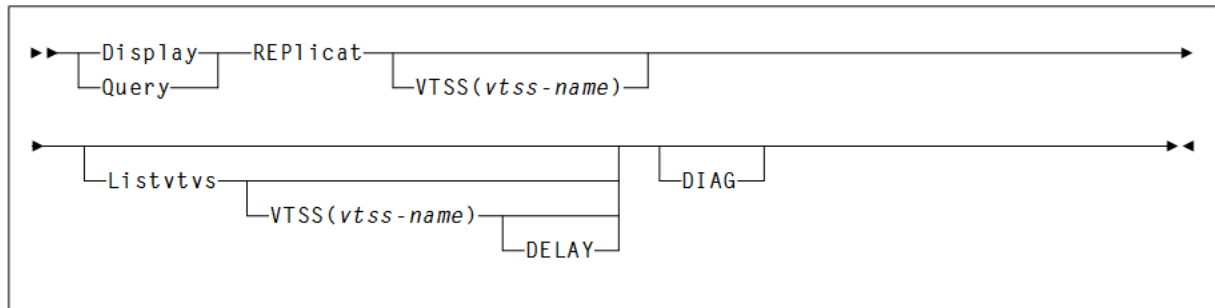
Active HSC/VTCS

**Description**

The `Display REPlicat` command displays VTV replication status.

**Syntax**

Figure 3–58 shows `Display REPlicat` command syntax.

**Figure 3–58** Display REPLICat command syntax**Parameters**

As shown in [Figure 3–58](#), the Display REPLICat command includes the following parameters:

**Listvtvs**

optionally, lists volsers of VTVs awaiting replication or electronic export from the specified VTSS, and indicates an age range, or number of minutes the VTVs have been queued.

**VTSS(vtss-name)**

optionally, lists volsers for only the specified VTSS.

*vtss-name* indicates the VTSS name.

**DELAY**

optionally, lists VTVs for the specified VTSS that are being delayed for both asynchronous replication and asynchronous electronic export. Output also indicates the minimum and maximum age of the delayed VTVs.

**VTSS(vtss-name)**

optionally, displays the status of VTVs awaiting replication or electronic export from the specified VTSS.

*vtss-name* indicates the VTSS name.

**DIAG**

optionally, displays additional diagnostic information.

**Output**

The following is an example of the output produced by the Display REPLICat command issued without any optional parameters:

**Example 3–30** Display REPLICat output

VTSS	Name	Host	Qdepth
HBVTSSY	PLEX1819	EC20	0
		EC21	0
		ECCL	0
		ECCY	0
HBVTSS18	-Cluster	EC20	0
		EC21	0
		ECCL	0
		ECCY	0

\*SLS5013I Command completed (0)

**Fields**

Display REPLICAT command output includes the following fields:

**VTSS**

the Primary or sending VTSS name.

**NAME**

one of the following values:

**name**

the name of the TapePlex to which CLINKs are connected from the VTSS.

**-Cluster**

indicates that the VTSS is a member of a cluster and has CLINKs defined for replication to other VTSSs in the cluster.

**HOST**

the host attached to the Primary VTSS.

**Qdepth**

the total number of VTVs waiting to be replicated.

The next example shows output produced by the Display REPLICAT command issued with the LISTVTVS and DELAY parameters:

**Example 3–31 Display REPLICAT LISTVTVS DELAY output**

```
D REPL LISTVTVS VTSS(DVTSS16) DELAY

SLS6603I VTVs awaiting replication
VTVs to be replicated to TAPEPLEX SECONDARY
DX0200
1 volumes with an age range of 1-1 in the queue
VTVs to be replicated within cluster BIDIR
DX0201 DX0202 DX0203
3 volumes with an age range of 1-3 in the queue
```

---

---

**Note:** If a VTV is assigned a MGMTCLAS that has both REP(YES) and EEXPOL(EE) with REPDELAY specified, the VTV only displays in the replicated to TAPEPLEX output section. The replication within the cluster will occur once the replication to TAPEPLEX has occurred.

---

---

## Display Requests

**Interfaces:**

- Console or PARMLIB only
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at BASE or FULL service level

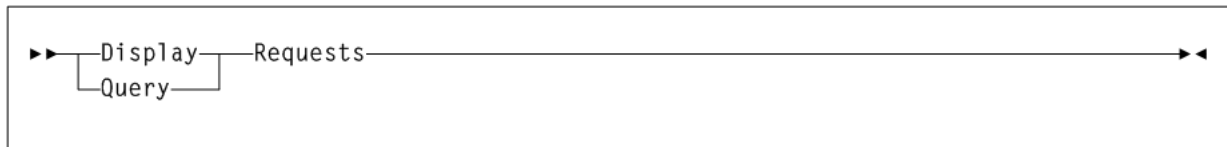
**Description**

The Display Requests command displays all pending LMU requests.

## Syntax

Figure 3–59 shows Display Requests command syntax.

**Figure 3–59 Display Requests command syntax**



## Parameters

None.

## Display RTD

### Interfaces:

- Console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC/VTCS

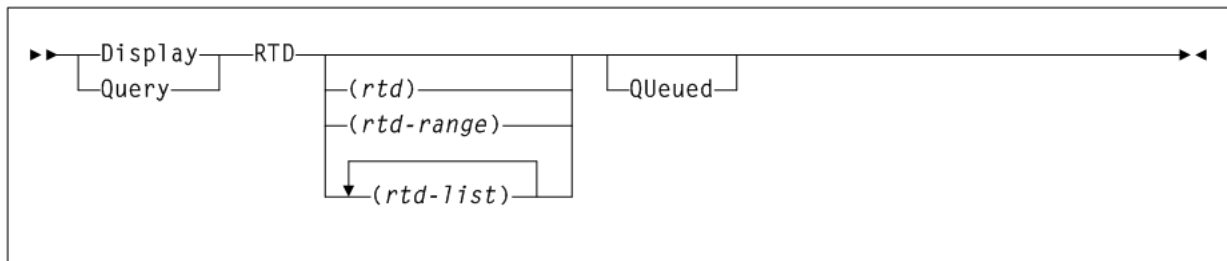
## Description

The Display RTD command displays usage information for the specified RTDs.

## Syntax

Figure 3–60 shows Display RTD command syntax.

**Figure 3–60 Display RTD command syntax**



## Parameters

As shown in Figure 3–60, the Display RTD command includes the following parameters:

### **rtd-id, rtd-range, or rtd-list**

optionally, the unit addresses of one or more RTDs.

### **Qeued**

optionally, display information about requests queued for the RTD.

## Output

The following is an example of the output produced by the Display RTD command:

**Example 3–32 Display RTD output**

RTD	MOUNT	ALLOC	HOST	VTSS	Status	Top ID	Top Host
B200	-	-	-	DHSS16	Offline	01265	ECC20
B201	-	-	-	DHSS16	Offline	02368	ECC20
B202	-	-	-	DHSS16	Online/free		
B201	-	-	-	DHSS16	Online/free		
0B79	-	-	-	DHSS16	Online/free		
0B7A	DMV051*	DMV051	EC20	DHSS18	Recall VTV		
1600					MVS1 :Migrate		
1601					MVS1 :Migrate		

**Fields**

Display RTD command output includes the following fields:

**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**

one of the following:

- The VTSS that is currently connected to the RTD.
- The VTSS that has paths to the Storage Manager containing the vRTD.
- - indicates that there are multiple VTSSs with paths to the Storage Manager containing the vRTD.

**Status**

One of the following RTD statuses:

**AUDIT MVC**

An MVC is being audited.

**BUSY**

The RTD is busy (non-specific task).

**FAIL/OFFLINE**

The RTD has been placed offline due to a failure.

**IDLE**

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

**INITIALIZE**

The host is verifying RTD status and availability.

**MAINTENANCE**

The RTD is in maintenance mode.



**MIGRATE VTV**

The RTD is migrating a VTV.

**OFFLINE**

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

**ONLINE/FREE**

The RTD is online and available.

**PATH MAINT**

One or more paths to the RTD have been varied into maintenance mode.

**PATH OFFLINE**

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

**PATH SUSPEND**

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

**PATHS OFFLINE**

There are no online paths to the Storage Manager containing the vRTD and all paths to the Storage Manager containing the vRTD from at least one online VTSS are offline.

**RECALL VTV**

The RTD is recalling a VTV.

**RECOVER RTD**

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

**RECOVERY**

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

**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, FICON and FIP.
- 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.

---

---

**Note:** An RTD can only be online if the CLINK is offline.

---

---

**UNKNOWN**

There are no online paths to the Storage Manager containing the vRTD and none of the following statuses apply:

- Paths Offline
- VTSS Offline
- VTSSs Offline

**UNLOAD MVC**

A forced unload of the RTD is occurring.

**VTSS Offline**

One of the following:

- The VTSS to which the RTD is connected is offline.
- There are no online paths to the Storage Manager containing the vRTD, `Paths offline` does not apply, and there is one offline VTSS with paths to the Storage Manager containing the vRTD.

**VTSSs Offline**

One of the following:

- The RTD is connected to multiple VTSSs, all of which are offline.
- There are no online paths to the Storage Manager containing the vRTD, `Paths offline` does not apply and there are multiple offline VTSSs with paths to the Storage Manager containing the vRTD.

**VTV TRANSFER**

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

**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 at the top of the queue to use this RTD. The **TOP HOST** columns indicates which host has the top claim upon the RTD.

**Output**

The following is an example of the output produced by the `Display RTD` command with the `QUed` parameter:

**Example 3-33 Display RTD output (Display RTD Queued)**

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

**Fields**

`Display RTD` command output includes the following fields:

**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 sufficiently 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 an MVC, this will be empty.

**STORAGE CLASS**

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

**Display SCRATCH****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at BASE or FULL service level

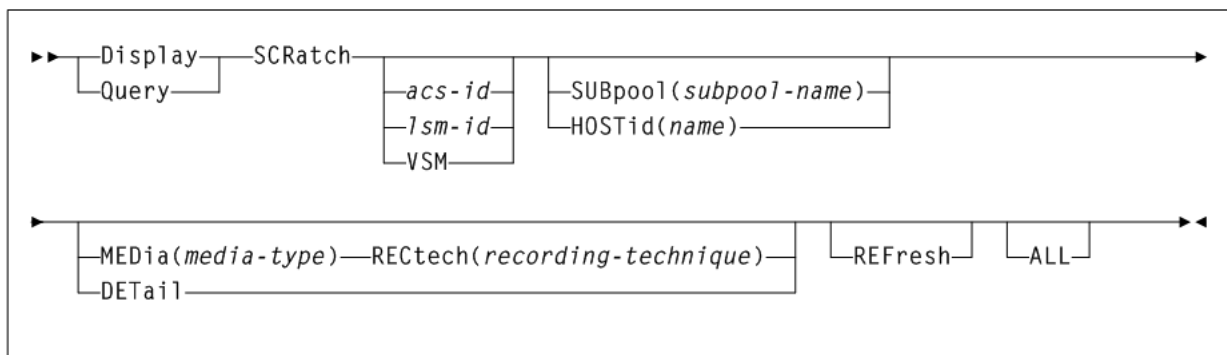
**Description**

The Display SCRATCH command displays scratch counts by subpool name, ACS id, LSM id, media type, recording technique, or owning host name.

**Syntax**

Figure 3-61 shows Display SCRATCH command syntax.

**Figure 3-61 Display SCRATCH command syntax**

**Parameters**

As shown in Figure 3-61, the Display SCRATCH command includes the following parameters:

**acs-id**

optionally, the ACS for which the system is to display scratch count information. If you specify *acs-id* without an LSM number, selected scratch counts for the entire ACS are displayed.

**lsm-id**

optionally, the LSM for which the system is to display scratch count information.

**VSM**

optionally, restricts the `Display SCRatch` display to subpools with virtual volumes. VSM is not allowed if *acs-id* or *lsm-id* are also specified.

---

---

**Note:** If *acs-id* or *lsm-id* is specified, it must be the first parameter after the `SCRatch` keyword. The `VSM` keyword is not positional.

---

---

**SUBpool(subpool-name)**

optionally, restricts the scratch count display to the named subpool only. If `SUBpool` is specified, then LSM and ACS totals will reflect only the named subpool. `SUBpool` is not allowed if `HOSTid` is also specified.

*subpool-name* indicates the name of the subpool. Specify up to 13 characters.

**HOSTid(name)**

optionally, restricts the scratch count display to subpools that are accessible from the named `HOSTid`. If `HOSTid` is specified, then LSM and ACS totals will reflect only subpools that can be accessed from the named `HOSTid`. `HOSTid` is not allowed if `SUBpool` is also specified.

*name* indicates the **SMFID** or the **SMC SERVER ALIASHOST ID** of the host. Specify up to 8 characters.

**MEDIA(media-type)**

optionally, restricts the scratch count display to the volumes compatible with the specified media name. If `MEDIA` is specified, then LSM and ACS totals will reflect only the compatible media types.

*media-type* indicates the media type.

See [Appendix A, "MEDIA, RECTech, and MODEL Values"](#) for a list of valid *media-type* values.

**RECTech(recording-technique)**

optionally, restricts the scratch count display to volumes compatible with the specified recording technique. If `RECTech` is specified, then LSM and ACS totals will reflect only the compatible recording techniques.

*recording-technique* indicates the recording technique.

See [Appendix A, "MEDIA, RECTech, and MODEL Values"](#) for valid recording-technique values.

**DETail**

optionally, lists scratch counts by all media and recording techniques. If specified, then an additional level of media and recording technique detail is displayed for every subpool. `DETail` is not allowed if either `MEDIA` or `RECTech` are also specified.

**REFresh**

optionally, updates VSM scratch counts prior to the display occurring.

---



---

**Note:** The REFresh parameter causes additional I/Os to the CDS and must be used sparingly.

---



---

**ALL**

optionally, lists scratch counts for subpools that have both a 0 scratch count and 0 threshold value. Normally, the Display SCRatch command will only list subpools that have either a non-zero scratch count or non-zero threshold value. Specification of ALL overrides this filter and results in the display of all defined scratch pools.

**Output**

The following are examples of the output produced by the Display SCRatch command.

The following example shows counts for all LSMs in all ACSs:

**Example 3–34 Display SCRatch output (all)**

```
DISPLAY SCRATCH
```

```
SLS2638I Scratch Summary
```

ACS/LSM	Subpool Name	Label	Media	Rectech	Count
VSM	SPV001	SL	VIRTUAL	VIRTUAL	930
VSM	SPV002	SL	VIRTUAL	VIRTUAL	1036
VSM	SPM001	SL	VIRTUAL	VIRTUAL	25
VSM	Total		VIRTUAL	VIRTUAL	1991
LSM 00:00	SP001	SL	All		13
LSM 00:00	SP002	SL	All		31
LSM 00:00	SPM001	SL	All		235
LSM 00:00	Non-Subpool		All		11
LSM 00:00	Total		All		290
LSM 00:01	SP001	SL	All		9
LSM 00:01	Non-Subpool		All		6
LSM 00:01	Total		All		15
ACS 00	SP001	SL	All		22
ACS 00	SP002	SL	All		31
ACS 00	SPM001	SL	All		235
ACS 00	Non-Subpool		All		17
ACS 00	Total		All		305

The following example shows Display SCRatch counts for a single subpool:

**Example 3–35 Display SCRatch output (subpool)**

```
DISPLAY SCRATCH SUBPOOL(SPM001)
```

```
SLS2638I Scratch Summary
```

ACS/LSM	Subpool Name	Label	Media	Rectech	Count
VSM	SPM001	SL	VIRTUAL	VIRTUAL	25
LSM 00:00	SPM001	SL	All		235
ACS 00	SPM001	SL	All		235

The following example shows Display SCRATCH counts for a single ACS:

**Example 3-36 Display SCRATCH output (ACS)**

DISPLAY SCRATCH ACS 00

SLS2638I Scratch Summary

ACS/LSM	Subpool Name	Label	Media	Rectech	Count
LSM 00:00	SP001	SL	All		13
LSM 00:00	SP002	SL	All		31
LSM 00:00	SPM001	SL	All		235
LSM 00:00	Non-Subpool		All		11
LSM 00:00	Total		All		290
LSM 00:01	SP001	SL	All		9
LSM 00:01	Non-Subpool		All		6
LSM 00:01	Total		All		15
ACS 00	SP001	SL	All		22
ACS 00	SP002	SL	All		31
ACS 00	SPM001	SL	All		235
ACS 00	Non-Subpool		All		17
ACS 00	Total		All		305

The following example shows Display SCRATCH count detail for a single LSM:

**Example 3-37 Display SCRATCH output (LSM)**

DISPLAY SCRATCH LSM 01:00 DETAIL

SLS2638I Scratch Summary

ACS/LSM	Subpool Name	Label	Media	Rectech	Count
LSM 01:00	SP002	SL	ECART	36TRACK	1
LSM 01:00	SP002	SL	STANDARD	18TRACK	5
LSM 01:00	SP002	SL	STANDARD	36TRACK	5
LSM 01:00	Non-Subpool		All		8
LSM 01:00	Total		All		19

## Display SEN

**Interfaces:**

- Console or PARMLIB only
- UUI Support: No

**Subsystem Requirements:**

Active HSC at BASE or FULL service level

## Description

The `Display SEN` command displays `SEN LISTEN` requests and their status. This display can display a specific request, using the requestor and listener names, a list of related requests using only the requestor name, or all `SEN` requests.

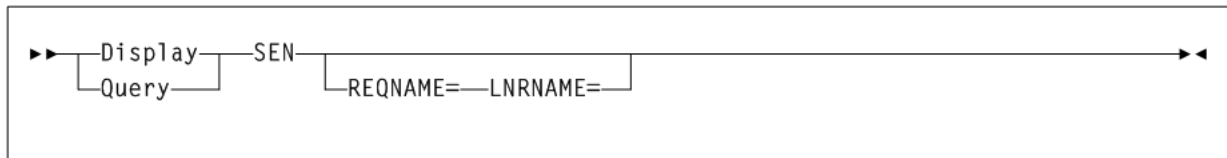
Displayed information includes the following:

- 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 3–62 shows `Display SEN` command syntax.

**Figure 3–62** *Display SEN command syntax*



## Parameters

As shown in Figure 3–62, the `Display SEN` command includes the following parameters:

### **REQNAME=**

optionally, specifies the name of the requestor identified on the `SEN` request.

### **LNRNAME=**

specifies the name of the listener routine identified on the `SEN` request.

## Supported HSC SEN Events

The following table lists supported HSC `SEN` events.

**Table 3–3** *HSC SEN Events*

Event Name	Numeric Equate
<code>libvol_insert_event</code>	18
<code>libvol_delete_event</code>	19
<code>hsc_termination_event</code>	20
<code>lsmrail_added_event</code>	25
<code>lsmrail_removed_event</code>	26
<code>libdrive_added_event</code>	27
<code>libdrive_removed_event</code>	28

## Supported VTCS SEN Events

The following table lists supported VTCS `SEN` events.

**Table 3–4 VTCS SEN Events**

Event Name	Numeric Equate
vtss_performance_event	01
vtss_chanif_performance_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
vtv_movement_event	13
vtv_scratch_event	14
vtv_replicate_event	15
tv_unlink_from_mvc_event	16
clink_vary_event	17

## Display SERVER

### Interfaces:

- Console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC/VTCS

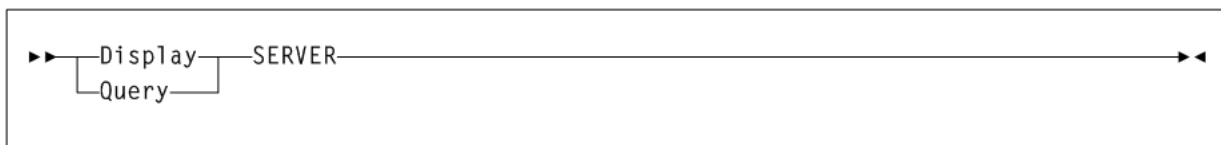
### Description

Display SERVER displays service level and database status.

### Syntax

Figure 3–63 shows Display SERVER command syntax.

**Figure 3–63 Display SERVER command syntax**



### Parameters

None.



## Output

The following is an example of the output produced by the `Display SERVER` command:

### Example 3–38 Display SERVER output

```
SLS1030I HSC service level FULL active
SLS0007I Data Base Status: 125
CDS          DSN          Active
Primary     ELS.DBASEPRM   Yes
Secondary   ELS.DBASESEC   Yes
Standby     (not configured)  No
```

## Display SRVlev

### Interfaces:

- Console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC at FULL service level

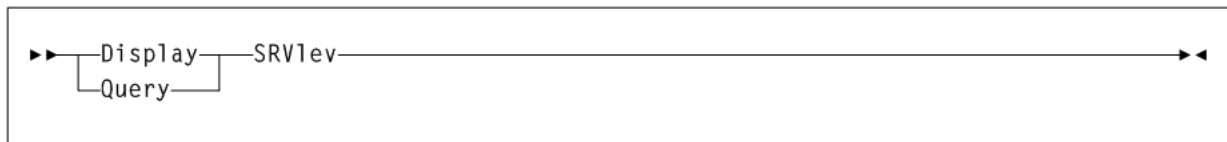
### Description

The `Display SRVlev` command displays the current HSC service level for the host issuing the command.

### Syntax

Figure 3–64 shows `Display SRVlev` command syntax.

Figure 3–64 Display SRVlev command syntax



### Parameters

None.

## Display Status

### Interfaces:

- Console or PARMLIB only
- UUI Support: No

### Subsystem Requirements:

Active HSC at BASE or FULL service level

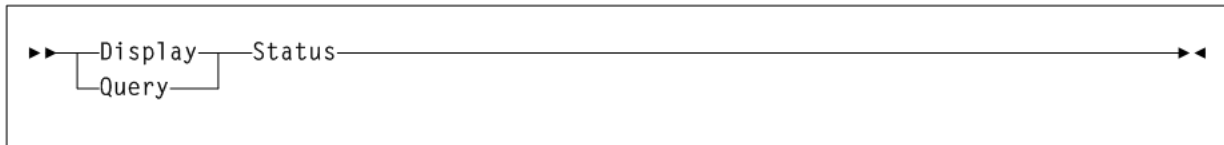
## Description

The `Display Status` command displays the status of pending requests currently active on the host issuing the command.

## Syntax

Figure 3–65 shows the syntax for the `Display Status` command.

**Figure 3–65** *Display Status command syntax*



## Parameters

None.

## Output

The `Display Status` command displays the following information to help you resolve problems during regular execution and termination of the HSC:

- requests: Dismount, Eject, Enter, Modify, Mount, Move, Vary, View
- information about the request, including:
  - the requester (such as Operator, PGMI, Clean, job name of utility)
  - physical element (such as CAPid, Drive, LSM, Station)
  - associated element for the request (such as CAPid belonging to an audit)
  - ready status of each queue, and whether a given queue is being purged or terminated
  - whether a switch is in progress for an ACS

---



---

### Note:

- Use the `Display Drives` command for information about current and pending mount activity, and the `Display Requests` command for information about queued LSM requests.
  - When an audit is running that is a full panel or more in scope, the current cell location points to the first cell in the panel being audited.
- 
- 

## Display STORCLas

### Interfaces:

- Console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC/VTCS

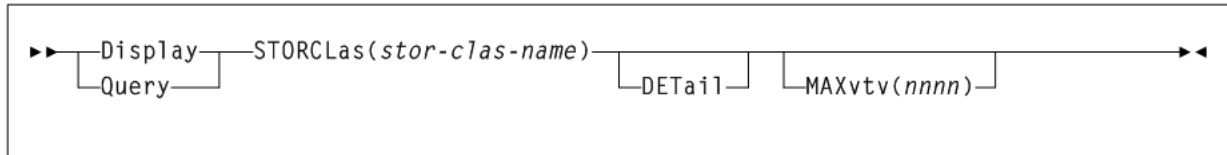
## Description

The `Display STORCLas` command displays information about a Storage Class.

## Syntax

Figure 3–66 shows `Display STORCLas` command syntax.

**Figure 3–66** *Display STORCLas command syntax*



## Parameters

As shown in Figure 3–66, the `Display STORCLas` command includes the following parameters:

### **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 *vtss-name* or `!ERROR`, in order to allow details of migrations to these Storage Classes to be displayed.

### **DETail**

optionally, the output lists VTVs currently queued for automatic migration or immediate migration with this Storage Class.

### **MAXvtv(nnnn)**

optionally, indicates the maximum number of VTVs to be listed in a single automatic migration or immediate migration list (for a VTSS).

*nnnn* indicates a maximum number of VTVs, from 0-9999. If not specified, a default value of 100 is used.

`MAXvtv(nnnn)` implies `DETail`.

---

**Caution:** When you specify `MAXvtv(nnnn)`, high values can cause temporary system degradation due to the number of WTO (write to operator) messages issued.

---

## Output

The following is an example of the output produced by the `Display STORCLas` command:

### **Example 3–39** *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

As shown in [Example 3–39](#), `Display STORCLas` output displays the following:

- 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 STORMNgr

### Interfaces:

- Console or PARMLIB only
- UUI Support: No

### Subsystem Requirements:

Active HSC/VTCS

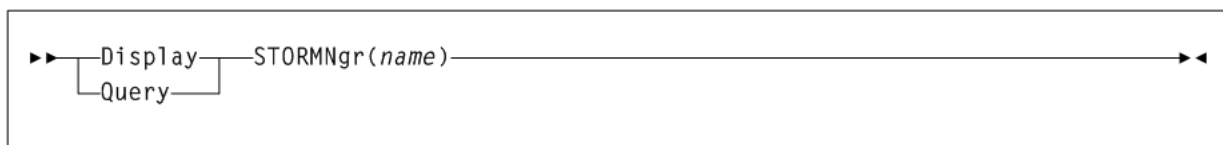
### Description

The `Display STORMNgr` command displays the status of an external storage manager and the paths defined to it from the VTSSs.

### Syntax

[Figure 3–67](#) shows `Display STORMNgr` command syntax.

**Figure 3–67** *Display STORMNgr command syntax*



### Parameters

As shown in [Figure 3–67](#), the `Display STORMNgr` command includes the following parameters:

#### **name**

the storage manager name.

### Output

The following is an example of the output produced by the `Display STORMNgr` command:

#### **Example 3–40** *Display STORMNgr output*

Name	Type	Version	Status
VLELIB0	VLE	1.1	Online

Features: Unknown

Device	Name	VTSS	CHANIF
V000	SS16IP0	DVTSS16	08 IOA:0
V002	SS16IP1	DVTSS16	09 IOI:0
V001	SS16IP2	DVTSS16	0A IIA:0
V003	SS16IP3	DVTSS16	0B III:0
V010	SS17IP0	DVTSS17	08 IOA:0
V012	SS17IP1	DVTSS17	09 IOI:0
V011	SS17IP2	DVTSS17	0A IIA:0
V013	SS17IP3	DVTSS17	0B III:0

## Fields

Display `STORMNgr` command output includes the following fields:

### Name

the name of the external storage manager

### Type

the type of Storage Manager:

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

### Version

the software version currently executing on the Storage Manager

### Status

the current status of the Storage Manager

### Features

a list of significant features installed or supported by the Storage Manager

### Device

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

### Name

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

### VTSS

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

### CHANIF

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

## Display TASKS

### Interfaces:

- Console or utility

- UI Support: Yes

**Subsystem Requirements:**

Active HSC at BASE or FULL service level

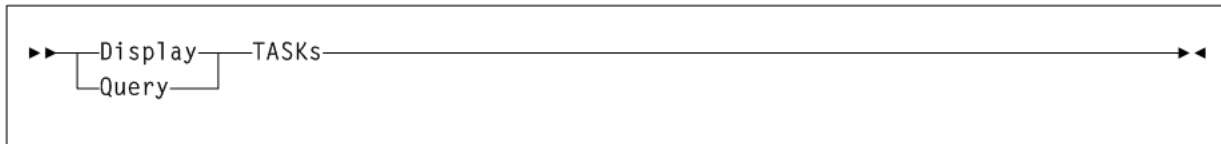
**Description**

The `Display TASKs` command displays task status.

**Syntax**

Figure 3–68 shows `Display TASKs` command syntax.

**Figure 3–68** *Display TASKs command syntax*



**Parameters**

None.

**Output**

The following is an example of the output produced by the `Display TASKs` command:

**Example 3–41** *Display TASKs output*

TASK NBR	TASK TYPE	SLOT ID	VTSS	RTD	CURRENT PROCESS	LOCKS HELD	WAITQ COUN	PENDQ COUNT
000	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		

**Fields**

`Display TASKs` command output includes the following fields:

**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
- Ctc - Cross TapePlex Communication Manager
- 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 THReshId****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at `BASE` or `FULL` service level

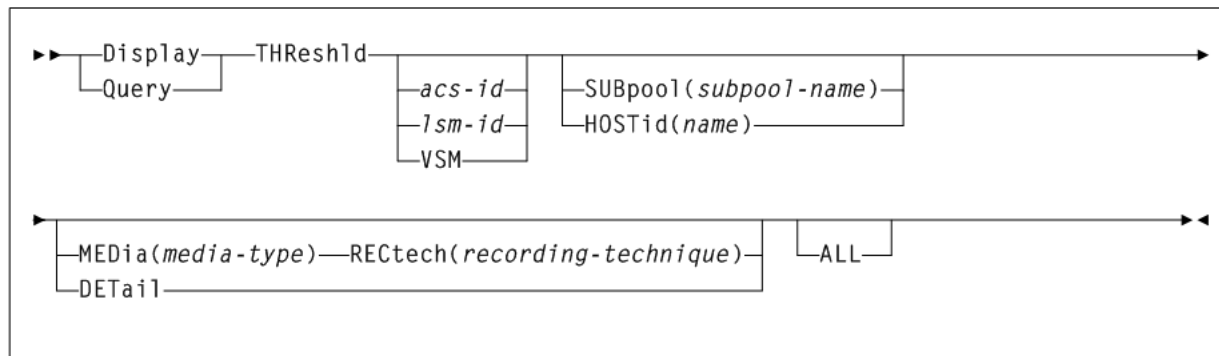
**Description**

`Display THReshId` displays scratch count and scratch threshold information by subpool name, ACS id, LSM id, media, and recording technique.

**Syntax**

Figure 3–69 shows `Display THReshId` command syntax.

Figure 3–69 Display THReshld command syntax



### Parameters

As shown in Figure 3–69, the Display THReshld command includes the following parameters:

#### THReshld

display scratch count and threshold information which can include scratch counts and scratch thresholds by subpool name, ACS id, LSM id, media, and recording technique.

#### acs-id

optionally, the ACS for which the system is to display scratch count and scratch threshold information. If you specify *acs-id* without an LSM number, selected scratch counts for the entire ACS are displayed.

#### lsm-id

optionally, the LSM for which the system is to display scratch count and scratch threshold information.

#### VSM

optionally, restricts the Display THReshld display to subpools with virtual volumes. VSM is not allowed if *acs-id* or *lsm-id* are also specified.

---

**Note:** optionally, restricts the Display THReshld display to subpools with virtual volumes. VSM is not allowed if *acs-id* or *lsm-id* are also specified.

---

#### SUBpool(subpool-name)

optionally, restricts the scratch count and scratch threshold display to the named subpool only. If SUBpool is specified, then LSM and ACS totals will reflect only the named subpool. SUBpool is not allowed if HOSTid is also specified.

*subpool-name* indicates the subpool name. Specify up to 13 characters.

#### HOSTid(name)

optionally, restricts the scratch count and scratch threshold display to subpools that are accessible from the named HOSTid. If HOSTid is specified, then LSM and ACS totals will reflect only subpools that can be accessed from the named HOSTid. HOSTid is not allowed if SUBpool is also specified.

*name* indicates the **SMFID** or the **SMC SERVER ALIASHOST ID** of the host. Specify up to 8 characters.



**MEDia(media-type)**

optionally, restricts the scratch count and scratch threshold display to the volumes compatible with the specified media name. If MEDia is specified, then LSM and ACS totals will reflect only the compatible media types.

*media-type* indicates the name of the desired media to display. See [Appendix A, "MEDia, RECtech, and MODel Values"](#) for valid media name values.

MEDia is not valid with the VSM parameter.

**RECtech(recording-technique)**

optionally, restricts the scratch count and scratch threshold display to volumes compatible with the specified recording technique. If RECtech is specified, then LSM and ACS totals reflect only the compatible recording techniques.

*recording-technique* indicates the name of the desired recording technique to display. See [Appendix A, "MEDia, RECtech, and MODel Values"](#) for valid media name values.

RECtech is not valid with the VSM parameter.

**DETail**

optionally, lists scratch counts and scratch threshold information by all media and recording techniques. If specified, then an additional level of media and recording technique detail is displayed for every subpool. DETail is not allowed if VSM, MEDia or RECtech are also specified.

**ALL**

optionally, lists scratch counts for subpools that have both a 0 scratch count and 0 threshold value. Normally, the Display THRESHold command will only list subpools that have either a non-zero scratch count or non-zero threshold value. Specification of ALL overrides this filter and results in the display of all defined scratch pools.

**Output**

The following is an example of the output produced by the Display THRESHold command:

**Example 3-42 Display THRESHold command output**

```
DISPLAY THRESHOLD
```

```
SL52649I Threshold Value Summary
```

ACS/LSM	Subpool Name	Label	Media	Rectech	Count	Thresh
VSM	SPV001	SL	VIRTUAL	VIRTUAL	930	0
VSM	SPV002	SL	VIRTUAL	VIRTUAL	1036	0
VSM	SPM001	SL	VIRTUAL	VIRTUAL	25	0
VSM	Total		VIRTUAL	VIRTUAL	1991	0
LSM 00:00	SP001	SL	All		13	0
LSM 00:00	SP002	SL	All		31	0
LSM 00:00	SPM001	SL	All		235	0
LSM 00:00	Total		All		290	0
LSM 00:01	SP001	SL	All		9	0
LSM 00:01	Total		All		15	0
ACS 00	SP001	SL	All		22	0
ACS 00	SP002	SL	All		31	0
ACS 00	SPM001	SL	All		235	0

```
ACS 00 Total                All                305                0
=====
```

## Display Volser

### Interfaces:

- Console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC at BASE or FULL service level

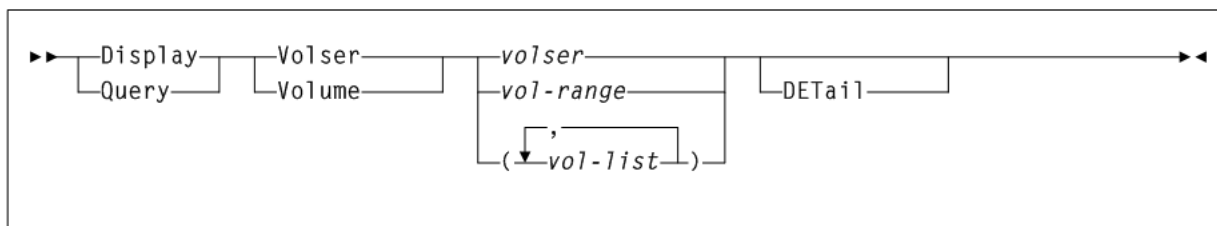
### Description

The `Display Volser` command displays volume information.

### Syntax

[Appendix 3-70](#) shows `Display Volser` command syntax.

**Figure 3-70** *Display Volser output example*



### Parameters

As shown in [Figure 3-70](#), the `Display Volser` command includes the following parameters:

#### Volser or Volume

displays volume locations for the specified volumes. The information displayed includes:

- status of volume (selected or unselected) and its location
- the source, destination, and home location (in the case of an errant volume)

*volser*, *vol-range*, or *vol-list* indicates one or more VOLSERs for which the system is to display status. Each *vol-list* element can be either a single VOLSER or a VOLSER range. You can display a range of up to 100 volumes.

List elements must be separated by commas or blanks, and the entire list must be enclosed in parentheses.

#### DETail

optionally, displays all available information about the specified volume(s).

### Output

The `Display Volser` command displays the following information:

- home cell location

- whether the volume is a scratch cartridge
- whether the volume is selected
- owning host (displayed if the volume is selected)
- drive address or drive ID (displayed if the volume is mounted)
- whether an external label is present
- whether the media label is readable by the robotic vision system (displayed if the external label exists)
- when the volume was last inserted into the library
- when the volume was last selected
- select count
- media type for the volume
- recording technique for the volume
- whether or not the media label was readable (N/A appears if the value was not read from the LMU)
- how the media type has been determined for the volume. The values that appear in this field include:

**YES**

The media label and the VOLPARM defined for the volume agree.

**NO**

The media label and the VOLPARM defined for the volume do not agree.

**VOLATTR Only**

A VOLPARM has been defined, but the LMU has not determined the media type.

**Label Only**

The LMU has determined the media type, but a VOLPARM has not been defined.

**Undefined**

A VOLPARM has not been defined, and the LMU has not determined the media type.

- whether or not the volumes are unusable (i.e., spent cleaning cartridges)
- volume density for STK1 and STK2 media volumes.
- Media Warranty Life (MWL) percentage, stored on the cartridge MIR and collected during dismount processing. MWL percentage indicates how much of the media life has been used.

---



---

**Note:** To collect and report media warranty life, tape libraries and transports must meet the following requirements:

- SL8500 or SL3000 libraries
- LMU compatibility level 21 or higher
- T9x40: all media and models at firmware level 1.42 or higher (except 9840B)
- T10000: all models and media at firmware level 1.38 or higher

Refer to Oracle's ELS publication *Managing HSC and VTCS* for more information about media warranty life.

---



---

## Display VOLUME\_Info

### Interfaces:

- Console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC/VTCS

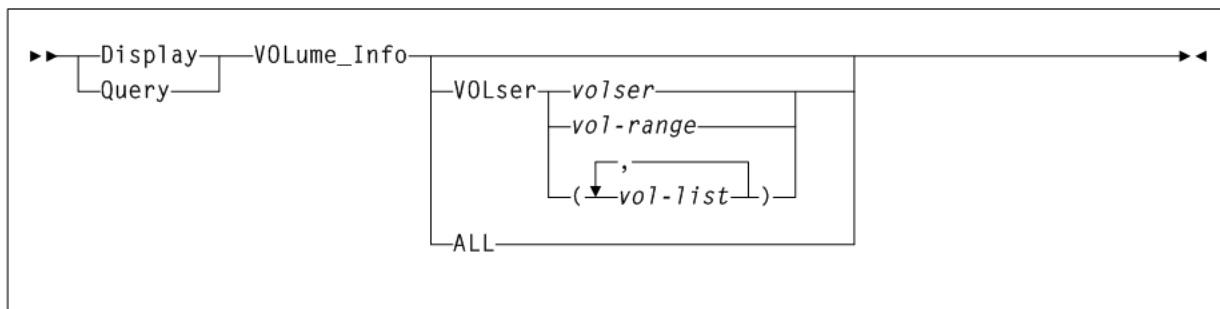
### Description

The `Display VOLUME_Info` command displays information for real and/or virtual volumes.

### Syntax

Figure 3-71 shows `Display VOLUME_Info` command syntax.

**Figure 3-71** *Display VOLUME\_Info command syntax*



### Parameters

As shown in Figure 3-71, the `Display VOLUME_Info` command includes the following parameters:

#### VOLser

displays volume information for the specified volumes. The information displayed includes the following:

- status of volume (selected or unselected) and its location
- the source, destination, and home location (in the case of an errant volume)

*volser*, *vol-range*, or *vol-list* indicates one or more VOLSERS for which the system is to display status. Each *vol-list* element can be either a single VOLSER or a VOLSER range. You can display a range of up to 100 volumes.

List elements must be separated by commas or blanks, and the entire list must be enclosed in parentheses.

#### ALL

Only real (non-virtual) volumes are listed. Volumes are listed in random order.

## Display VSCRatch

#### Interfaces:

- Console or utility
- UUI Support: Yes

#### Subsystem Requirements:

Active HSC/VTCS

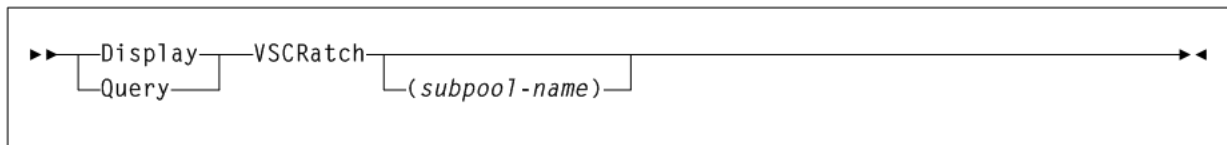
#### Description

The `Display VSCRatch` command displays diagnostic information for virtual scratch counts.

#### Syntax

Figure 3-72 shows `Display VSCRatch` command syntax.

**Figure 3-72** *Display VSCRatch command syntax*



#### Parameters

As shown in Figure 3-72, the `Display VSCRatch` command includes the following parameters:

##### **subpool-name**

optionally, a subpool name. Specify up to 13 characters.

If a subpool name is specified, the virtual scratch count display is restricted to the named subpool only.

#### Output

The following is an example of the output produced by the `Display VSCRatch` command:

##### **Example 3-43** *Display VSCRatch output*

Subpool-Name	Scratch Count	Label Type	<----- Sizes available ----->				
			0.4GB	0.8GB	2GB	4GB	32GB
M0	972	SL	Yes	Yes	Yes	Yes	N/A
M1	994	SL	Yes	Yes	Yes	Yes	N/A
M2	1,000	SL	Yes	Yes	Yes	Yes	N/A

**Fields**

Display VSCRatch command output includes the following fields:

**Subpool-Name**

the name of the scratch subpool. 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 scratch counts.

**Label Type**

one of the following scratch label types:

- SL indicates standard-labeled tape.
- AL indicates ANSI-labeled tape.
- NL indicates nonlabeled tape.
- NSL indicates nonstandard labeled tape.

**Sizes available**

valid VTV sizes (0.4 GB, 0.8 GB, 2 GB, 4 GB, 32 GB).

**Display VTD**

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

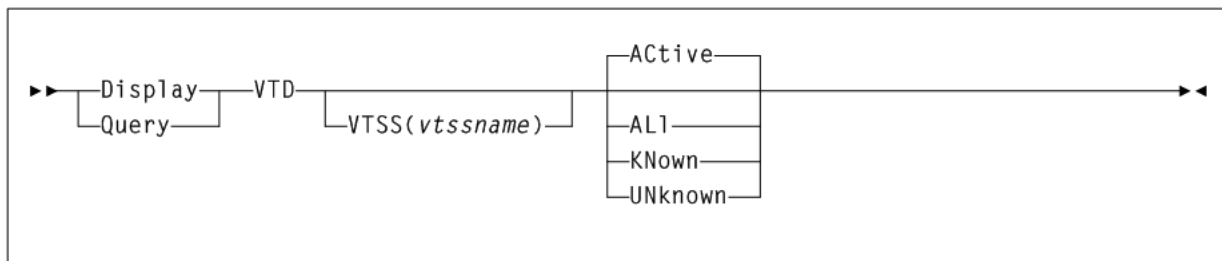
**Description**

The Display VTD command displays VTD status.

**Syntax**

Figure 3–73 shows Display VTD command syntax.

**Figure 3–73 Display VTD command syntax**



## Parameters

As shown in [Figure 3–69](#), the `Display THReshld` command includes the following parameters:

### VTSS(*vtss-name*)

Optionally, display status for the VTDs connected to the specified VTSS.

*vtss-name* indicates the VTSS name.

### Active

Display status for VTDs that have a VTV mounted. This can include both known and unknown VTDs.

### ALI

Display status for all VTDs.

### Known

Display status for all VTDs that have a known MVS address on this system.

### Unknown

Display status for all VTDs that do not have a known MVS address on this system.

## Output

The following is an example of the output produced by the `Display VTD` command:

### Example 3–44 Display VTD output

Drive	Location	VTV	Status	Usage	Client address
A800	HBVTSS16	X00778	Available	ECAM only	
A801	HBVTSS16		Available	ECAM only	
A802	HBVTSS16		Available	ECAM only	
A803	HBVTSS16		Available		EC21:B412

## Fields

`Display VTD` command output includes the following fields:

### Drive

The MVS device address of the VTD. If the device address is prefixed by a # sign then this is the address within the VTSS, the MVS address is unknown to this VTCS.

### 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.

#### 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.

**Imported**

The VTV volser shown in the VTV column has been electronically imported via the VTD. It is awaiting the confirmation request from the host that performed the export.

**Importing**

The VTV volser shown in the VTV column is being electronically imported via the VTD.

**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.

**Mount (other)**

The VTV volser shown in the VTV column is mounted on the VTD. The mount was not performed by the host on which the command was executed.

**Not Available**

The VTD is not available for work.

**Redrive Later**

A previous attempt to mount the VTV volser shown in the VTV column upon the VTD failed. It will be interpreted again within the next few minutes.

**Replicating**

The VTV volser shown in the VTV column is in the process of being replicated.

**Undefined**

The VTV volser shown in the VTV column is undefined.

**Unknown**

The VTV volser shown in the VTV column is unknown.

**Usage**

one of the following:

**Configuring**

Configuration of the VTD is in progress.

**ECAM init error**

Unable to initialise VTD through ECAM-t with VTSS.

**ECAM I/O error**

ECAM-t I/O has failed on this VTD.

**ECAM only**

The VTD is one of the VTDs connected to another VTSS for the process of replicating VTVs. In [Example 3-44](#), X00778 is the VTV being replicated.

**Not accessed**

The VTD is not found in the VTSS configuration.

**Not verified**

The VTD is defined with NOVERIFY in the configuration.



**Client address**

the address of the drive as reported by the client that last performed a mount on the VTD.

**Display VTSS****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

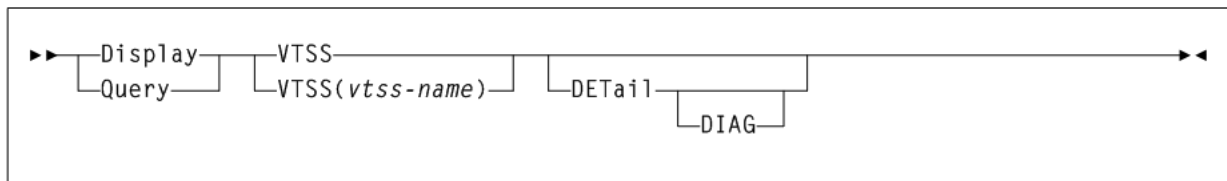
**Description**

The `Display VTSS` command displays VTSS status.

**Syntax**

Figure 3-74 shows `Display VTSS` command syntax.

**Figure 3-74** *Display VTSS command syntax*

**Parameters**

As shown in Figure 3-74, the `Display VTSS` command includes the following parameters:

**vtss-name**

optionally, indicates the VTSS for which to display status.

**DETail**

optionally, display detailed VTSS status information including supported features.

**DIAG**

optionally, display more detailed diagnostic information about the TCP/IP connections and control parameters in effect.

**Output**

The following is an example of the output produced by the `Display VTSS` command:

**Example 3-45** *Display VTSS output*

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

## Fields

Display VTSS command output includes the following fields:

### **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).

**RTDs**

indicates whether the VTSS has RTDs.

**Display VTSS DETail Output**

The following is an example of the output produced by the `Display VTSS DETail` command:

**Example 3–46 Display VTSS Detail output**

Vtss	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 MVCS CONNECT to 32 RTDS STACKED MIGRATES

**Fields**

`Display VTSS DETail` command output includes the following fields:

**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).

**ACCESS**

The connection method employed for the host, either TCPIP or CHANNEL ECAM-T protocol.

**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.

**PARTITIONED RTDs**

Partitioned RTDS enabled.

**T10KC Partition RTDs**

T10KC partitioned RTDs enabled.

**Write New VOL1 Label**

VTSS supports MVC labeling.

**CONCURRENT PORT I/O**

VTSS supports multiple I/Os on a single physical nearlink interface.

**MANY-TO-MANY CONNECT**

VTSS supports target VTD selection for a CLINK.

**IP REPLICATION**

VTSS supports IP CLINKs and therefore a VLE connection.

**CONCURRENT TAPE RECALL/MOUNT**

VTV mounts on VTDs can occur before the recall has completed.

**CONCURRENT TAPE RECALL/MOUNT \*DISABLED**

The Concurrent Tape Recall/mount feature is installed on the VTSS but has been disabled by either `CONFIG GLOBAL FASTRECL=NO` or `CONFIG VTSS NOERLYMT`.

**UNKNOWN VTSS FEATURE**

VTSS feature found not supported by VTCS software level.

**Enhanced replication Release 1**

VTSS supports enhanced synchronous replication.

**Supports VVV VTDs**

`VVV` indicates the number of VTDs currently supported by the VSM 6 (or later).

**NONE**

No VTSS features defined.

**Note:**

- If ETTB (Early Time to First Byte) is enabled, the following is displayed:

VTV mounts on VTDs can occur before the recall has completed.

- If ETTFB is disabled, the following is displayed:

The concurrent tape recall/mount feature is installed on the VTSS but has been disabled by either `CONFIG GLOBAL FASTRECL=NO` or `CONFIG VTSS NOERLYMT`.

**Display VTV****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

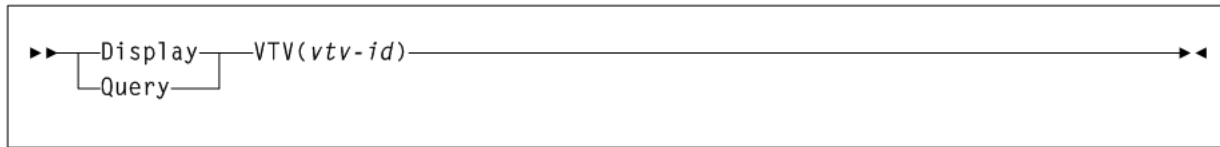
Active HSC/VTCS

**Description**

`Display VTV` displays information about a specific VTV.

**Syntax**

Figure 3–75 shows `Display VTV` command syntax.

**Figure 3–75 Display VTV command syntax****Parameters**

As shown in [Figure 3–75](#), the Display VTV command includes the following parameters:

**vtv-id**

the ID of the VTV.

**Output**

The following is an example of the output produced by the Display VTV command:

**Example 3–47 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          2008Feb21 09:32:22
Last Mounted           2008Feb21 09:32:23
Last Recalled          2008Apr05 08:15:47
First Migrated         2008Apr25 08:58:28
Last Used Date         2008Apr25 08:58:28
MVC(s)                 021681                021645
Block-id(s)            1A400D24                25401185
Management Class       M13                    SB
Migrate Pending        SA
Status                 Initialized

```

**Fields**

Display VTV command output includes the following fields:

**VOLSER**

the volser of the VTV specified in the query.

**VTSS**

the VTSS where the VTV currently resides.

If the replicated copy of the VTV is pinned, the following is displayed:

```
VTSS: vtss-name (pinned)
```

**MOUNTED**

The MVS device address of the VTD. If the device address is prefixed by a # sign then this is the address within the VTSS, the MVS address is unknown to this VTCS.

**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.

**Partition-Id(s)**

the starting partition id on the 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.

**REPLICATION STARTED**

If the VTV is being replicated between VTSSs in a cluster, the name of the target VTSS is displayed.

**REPLICATED**

If the VTV has been replicated to another VTSS in the TapePlex, the VTSS name displayed indicates where the replica can be found.

If the replicated copy of the VTV is pinned, the following is displayed:

Replicated: *vtss-name* (pinned)

**IMPORTING**

If the VTV is being mounted via a VTD, the VTD unit address is displayed.

**ELECTRONIC IMPORTED**

If the VTV has been imported from another TapePlex. This will report the TapePlex name that owns the VTV

**EXPORTED**

If the VTV has been exported to another TapePlex. The line will also list the TapePlexes to which the VTV copies has been successfully exported. An asterisks (\*) next to the name indicates an export that has been rejected by the target TapePlex.

**STATUS**

one or more of the following statuses:

**CONSOLIDATED**

VSM has consolidated the VTV.

**DR TEST VERSION**

The VTV was created on a DR Test system, for example, when running CDRT.

**DUPLEXED**

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

**EXPORT-NOT POSSIBLE**

Export of this VTV to a remote TapePlex was attempted and the request was rejected. Typically, this is due to a different copy of the VTV residing in the remote TapePlex.

**EXPORT-REJECTED**

Electronic export to one or more TapePlexes was actively rejected. This could be due to the target TapePlex not allowing import of the VTV, or a clash with copy status.

**EXPORT-REQUIRED**

This VTV should be electronically exported and is currently queued for processing.

**EXPORT-STARTED**

Electronic export is active for this VTV, but not yet complete.

**INITIALIZED**

VTCS has used the VTV at least once.

**MIGRATED**

VSM has migrated the VTV.

**MIGRATE PENDING**

VTV migration is pending. This status is displayed when a VTV is initially created, or when the VTV requires reconciling or archiving. In these latter cases, individual MVC copies may indicate *Reconcil* or *Deletion*.

**REPLICATION COMPLETE**

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



**REPLICATION REQUIRED**

This VTV should be replicated and is currently queued for processing.

**REPLICATION STARTED**

Replication is active for this VTV but not yet complete.

**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.

**AVOID EARLY MOUNT**

Concurrent recall/mount encountered an error with this VTV. No further concurrent recall/mount activity will be attempted for this VTV.

---

---

**Note:** VTVs that have incurred an ETTFB recall error have an error flag set in their VTV record in the CDS. This error indicates to VTCS that no further ETTFB recalls will occur for this VTV. When the error flag is on, the following is displayed:

```
concurrent recall/mount encountered an error with this VTV.  
No further concurrent recall/mount activity will be  
attempted for this VTV.
```

---

---

## DRAin

**Interfaces:**

- Console or PARMLIB only
- UUI Support: No

**Subsystem Requirements:**

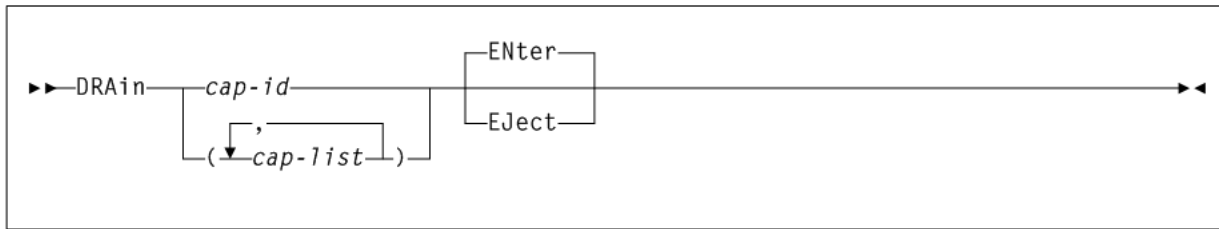
Active HSC at FULL service level

## Description

The `DRAin` command terminates an eject or enter operation on one or more CAPs, which frees the CAP resources for use by other processes.

## Syntax

Figure 3–76 shows `DRAin` command syntax.

**Figure 3–76 DRAIn command syntax**

## Parameters

As shown in [Figure 3–76](#), the `DRAIn` command includes the following parameters:

### **cap-id or cap-list**

one or more CAPs to be drained. The CAPids specified on a `DRAIn` command must match the CAPids specified on the associated `EJect` or `ENter` command.

The format of a `cap-id` is `AA:LL:CC`, where `AA:LL` indicates the LSMid, and `CC` indicates the CAP. See "[CAPid](#)" on page B-1 for a list of valid values.

Each `cap-list` element can be either an `lsm-id` or a `cap-id` that describes a specific CAP. The elements in a list must be separated by a comma or a blank, and the entire list must be enclosed in parentheses.

### **ENter**

specifies that an enter operation is to be terminated on the specified CAPs. `ENter` is the default.

### **EJect**

specifies that an eject operation is to be terminated on the specified CAPs.

## DRCHKPT

### **Interfaces:**

- SLUADMIN utility only
- UUI Support: No

### **Subsystem Requirements:**

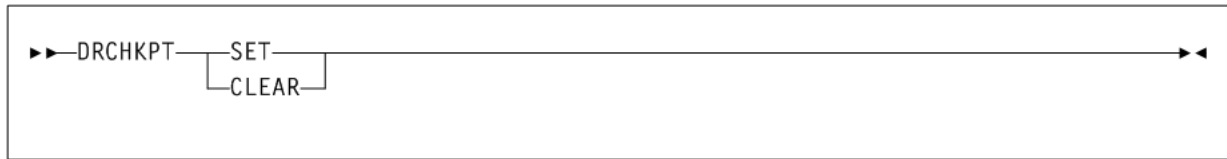
Active HSC/VTCS at FULL service level

## Description

The `DRCHKPT` command establishes the system recovery point (checkpoint) from which MVC content can be recovered from for a period of time.

## Syntax

[Figure 3–77](#) shows `DRCHKPT` command syntax.

**Figure 3–77 DRCHKPT command syntax**

## Parameters

As shown in [Figure 3–77](#), the DRCHKPT command includes the following parameters:

### SET

Sets a date/time stamp in the active CDS which establishes the recovery point. Beginning at this recovery point, MVC content is guaranteed for a period of time in the future (for example, until another DRCHKPT utility is run).

### CLEAR

Removes any recovery point set in the active CDS. Once removed, MVC content cannot be guaranteed.

## DRCLEAN

### Interfaces:

- SLUADMIN utility only
- UUI Support: No

### Subsystem Requirements:

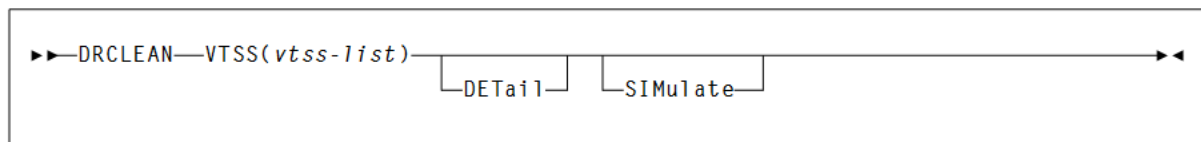
Active HSC/VTCS at FULL service level

## Description

The DRCLEAN utility specifies list of one or more VTSSs from which the DRTEST environment VTV copies will be deleted.

## Syntax

[Figure 3–78](#) shows DRCLEAN command syntax.

**Figure 3–78 DRCLEAN command syntax**

## Parameters

As shown in [Figure 3–78](#), the DRCLEAN command includes the following parameters:

### VTSS(vtss-list)

specifies a list of VTSSs from which the DR Test VTV copies will be deleted. Each VTSS listed is processed in parallel. Results are written to the report by VTSS in the order of

completion, which might not necessarily be the order of specification in the command. This is the only required parameter.

*vtss-list* indicates one or more VTSSs.

**DETail**

optionally, provides a line of output in the report for every VTV successfully deleted. Any errors encountered in processing are always reported. Totals for all VTVs successfully deleted and all error situations are always produced at the end of the run. The default is not to report each successful VTV deletion and just to provide the summary of results at the end of the run.

**SIMulate**

optionally, run the DRCLEAN utility in simulation mode. All processing continues as normal except that the actual VTV deletion is not executed. You may run this and see what would occur, including any coherency or query errors arising, without actually deleting any VTVs from the VTSS.

## DRMONitr

**Interfaces:**

- SLUADMIN utility only
- UI Support: No

**Subsystem Requirements:**

Active HSC/VTCS at FULL service level

## Description

The DRMONitr utility stalls job stream processing to ensure that critical data reaches its target destination. Once all identified data is accounted for, the utility ends.

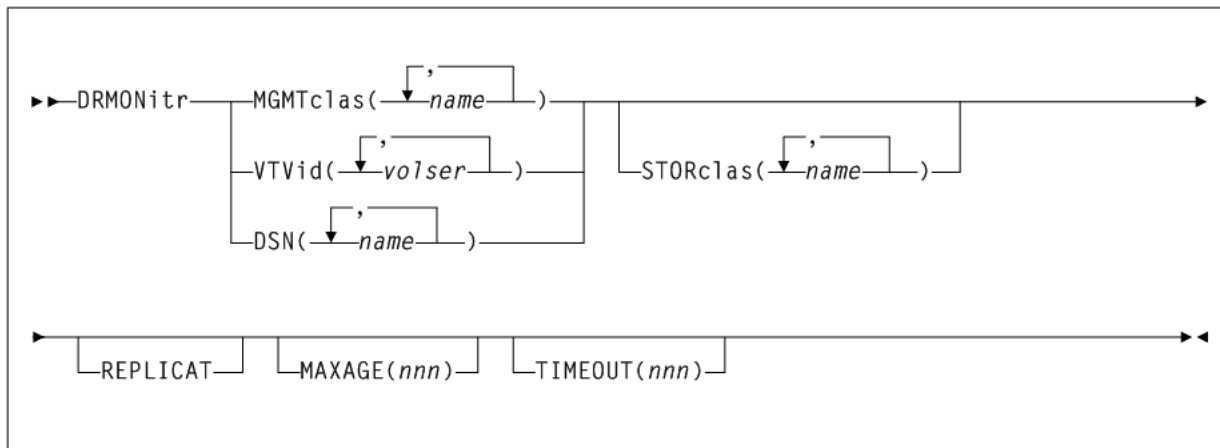
DRMONitr can monitor for MVC and remote TapePlex copies being complete. It can also monitor for cluster replication being complete, in support of tapeless environments.

DRMONitr reports any VTV exception conditions found during execution.

## Syntax

[Figure 3–79](#) shows DRMONitr command syntax.

Figure 3–79 DRMONitr command syntax



## Parameters

As shown in Figure 3–79, the DRMONitr command includes the following parameters:

### MGMTclas(name)

optionally, specifies a list of Management Classes to be monitored for VTV migrate/copy completion.

*name* indicates one or more Management Classes.

Specified Management Classes must be defined to the active VTCS system and must specify immediate migrate. For tapeless environments, specified Management Classes must specify REPLICAT.

### VTVID(volser)

optionally, specifies a list and/or range of VTV volsers to be monitored for migrate/copy completion.

*volser* indicates one or more volume serial numbers.

The VTV's Management Class must be defined to the active VTCS system and must specify immediate migrate, i.e. the Management Class must specify IMMEdmig (KEEP), IMMEdmig (DELETE) or IMMDELAY (*n*).

For tapeless environments, the VTV's Management Class must specify REPLICAT.

### DSN(name)

optionally, specifies a list of data sets to be monitored for migrate/copy completion.

The VTVs containing these data sets must have Management Classes defined to the active VTCS system and must specify immediate migrate. For tapeless environments, the VTV's Management Class must specify REPLICAT.

### STORclas(name)

optionally, specifies the Storage Class locations that the identified VTV data is to be delivered to.

*name* indicates one or more Storage Class locations.

Specified Storage Classes must be defined to the active VTCS system. Although this parameter is optional, the STORclas parameter should only be omitted in tapeless environments.

**REPLICAT**

optionally, specifies that VTV cluster replication is to be monitored for completion. Replication must be specified on the VTV's Management Class policy for the VTV to be selected for monitoring. If replications are monitored in a tape environment, their completion overrides any Storage Class requirement.

**MAXAGE(nnn)**

optionally, specifies the maximum VTV age (time since last update), in hours, when monitoring migrations by Management Class (a `MGMTclas` value is required).

*nnn* indicates the age in hours. Valid values are 1 to 999. There is no default.

If the VTV's age (time since last update) is greater than `MAXAGE`, the VTV is not monitored. `DRMONitr` reports the number of skipped VTVs due to `MAXAGE`, if specified.

If you do not specify `MAXAGE`, any active VTV belonging to the specified Management Classes, regardless of age, is monitored for migrate or copy completion.

**TIMEOUT(nnn)**

optionally, specifies the maximum time, in minutes, for `DRMONitr` to run.

*nnn* indicates the maximum time. Valid values are zero to 999. If you do not specify a `TIMEOUT` value, there is no limit on the time the utility can run (stall).

A `TIMEOUT` value of zero does not stall the `DRMONitr` utility. This special case reports on incomplete copies of specified VTV data.

If a non-zero `TIMEOUT` value is exceeded, `DRMONitr` ends with RC 8 and generates an error message. If the `TIMEOUT` value specified was zero, a return code of 4 is set if a VTV's copy is detected as incomplete, otherwise a return code 0 is set.

---



---

**Note:** You must specify either `STORclas` or `REPLICAT` or both to specify the destination criteria. Otherwise `DRMONitr` ends with RC 8 and generates an error message. Additionally, Oracle recommends that you do not specify (or monitor) Storage Classes in tapeless environments because migrations are not possible.

---



---

## DRTEST

The `DRTEST` command sets up the environment for DR testing and optionally, starts and stops the test.

Issue this command from the `SLUADMIN` utility to set up the DR testing environment. Once the environment is set, you can issue `DRTEST` from a utility or the console to start or stop testing.

Issue this command with any of the options listed in the following table. Each option is described individually, and in more detail, on the pages to follow.

**Table 3–5 DRTEST Options**

Option	Description
CREATE	creates a DR test CDS, normally run at the DR test site
PRIMEprd	updates the production CDS without creating a <code>DRTEST</code> CDS
RESET	removes all DR test settings in the production CDS
START	starts the DR test on the production site. <code>DRTEST START</code> cannot run successfully until the production CDS is primed using <code>DRTEST PRIMEprd</code> or <code>DRTEST CREATE</code> .

**Table 3–5 (Cont.) DRTEST Options**

Option	Description
STOP	stops the DR test on the production site

**Note:**

- You can use a combination of these options provided you have correct environment and JCL requirements. For example, DRTEST STOP and DRTEST RESET can be run in the same job, however, DRTEST STOP must be run before DRTEST RESET.
- Refer to Oracle's ELS publication *ELS Disaster Recovery and Offsite Data Management Guide* for information about how to use the DRTEST utility.

**DRTEST CREATE****Interfaces:**

- SLUADMIN utility only
- UI Support: No

**Subsystem Requirements:**

Active HSC not required

**Description**

The DRTEST CREATE command creates a DR test CDS, normally run at the DR test site.

---

**Note:** The NOUPDprd parameter is only valid if the DRTEST PRIMEprd command is executed prior to the DRTEST CREATE command. See "DRTEST PRIMEprd" on page 3-176 for more information.

---

Consider the following:

- DR test cannot be active. This refers to the production CDS status.
- Active HSC/VTCS is not required.
  - If HSC/VTCS is active, SLSCNTL DD statement(s) identifying the production CDS are optional, and if included, must match the active HSC/VTCS.
  - If HSC/VTCS is not active, SLSCNTL DD statement(s) identifying the production CDS are required.
- SLSNEW DD statement(s) identifying the DRTEST CDS(s) are required.

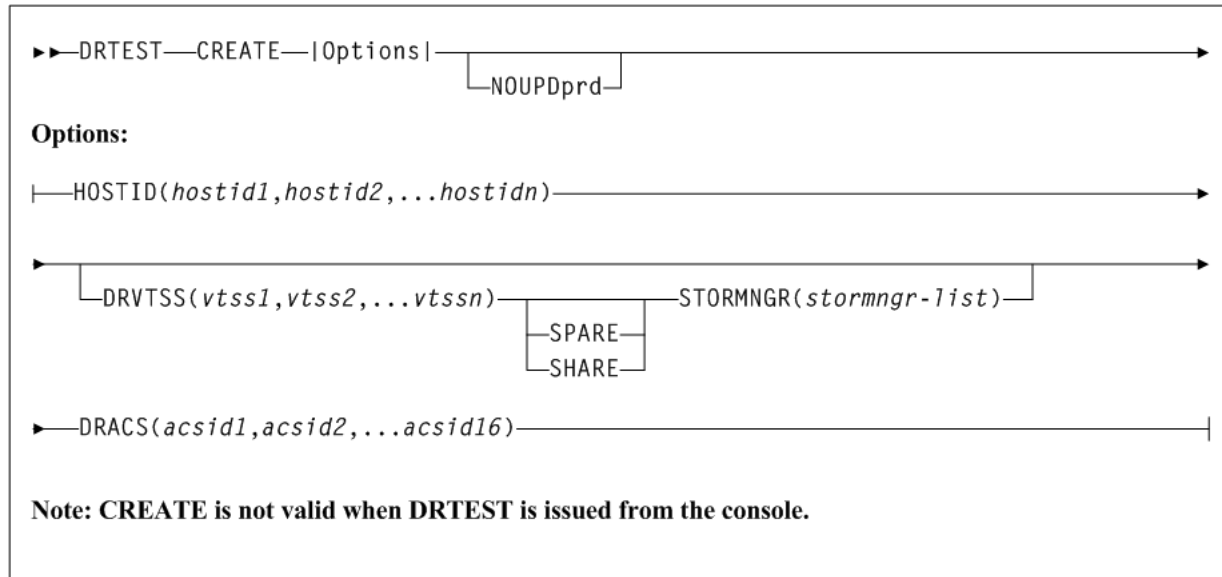
Sample Control Cards:

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

**Syntax**

Figure 3–80 shows DRTEST CREATE command syntax.

Figure 3–80 DRTEST CREATE command syntax



### Parameters

As shown in Figure 3–80, the DRTEST CREATE command includes the following parameters:

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

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

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

Optionally, specifies the VTSS names available to the DR test site. Unless you also specify the SHARE or SPARE parameter, these VTSSs are offline to the production site and online to the DR test site when the DR test starts.

#### SPARE

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

#### SHARE

Optionally, specifies that the DR test VTSS(s) are shared with the production site. During the DR test, both the test and production site may create VTVs in the DR test VTSS (based on the VOLPARM definitions). The DR test cannot modify VTVs that are not in a DR test subpool. The parameters DRACS and STORMNGR cannot be specified if SHARE is specified.

#### STORMNGR(*stormngr-list*)

Optionally, specifies the subsystem names of the VLEs attached to the DR test site TapePlex.

*stormngr-list* indicates the list of subsystem names. For more information, refer to Oracle's ELS publication *Configuring the Host Software for VLE*.



**DRACS(acsid1, acsid2,...acsid16)**

Optionally, specifies one or more ACSs available to the DR test site. Multiple ACS IDs must be specified as a list, and not as a range of ACS IDs. The ACSs are online to the DR test site but can remain connected (and usually are) to the production site when the DR test starts.

- If the `DRVTSS SHARE` parameter is specified, the `DRACS` parameter is not allowed.
- If the `STORMNGR` parameter is specified, the `DRACS` parameter is allowed but not required.

**NOUPDprd**

Optionally, does not update the input production CDS(s).

If this parameter is specified, you can use a single CDS production copy as input and that copy can be either the actual primary production CDS, a backup copy, or a mirrored copy.

This parameter is only valid if the `PRIMEprd` function has already been run on the production CDS. See "[DRTEST PRIMEprd](#)" on page 3-176 for more information.

**Additional JCL Requirements**

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statements apply to the `DRTEST CREATE JCL`:

**SLSNEW1**

specifies the new primary copy of the test HSC CDS.

**SLSNEW2**

specifies the new secondary copy of the test HSC CDS.

**SLSNEW3**

specifies the new standby copy of the test HSC CDS.

**SLSJRN00 – SLSJRNnn**

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

**SLSSTATN**

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

Each entry follows the HSC `SET SLISTATN` utility format. See "[SET SLISTATN](#)" on page 3-334 for more information. The following figure shows an example of the `SLSSTATN` file:

**Example 3–48 SLSSTATN File**

```
SET SLISTATN(00E7,00E8),FORACS(01),FORHOST(PRI1)
SET SLISTATN(00E7,00E8),FORACS(01),FORHOST(PRI1)
SET SLISTATN(00E7,00E8),FORACS(01),FORHOST(PRI2)
SET SLISTATN(00E7,00E8),FORACS(01),FORHOST(PRIY)
```

**SLSVTSS**

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

---



---

**Caution:** Improper use of CONFIG RESET against the DR test CDS may render the DR test environment inoperable!

---



---

Each entry follows the format of the VTSS, VTD, RTD, and HOST statements of the VTCS CONFIG utility. See "CONFIG" on page 3-37 for more information. The following figure shows an example of the SLSVTSS file:

**Example 3-49 SLSVTSS File**

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

**DRTEST PRIMEprd****Interfaces:**

- SLUADMIN utility only
- UI Support: No

**Subsystem Requirements:**

Active HSC not required

**Description**

DRTEST PRIMEprd updates the production CDS without creating a DRTEST CDS.

**Considerations:**

- DR test cannot be active. This refers to the production CDS status.
- Active HSC/VTCS is not required.
  - If HSC/VTCS is active, SLSCNTL DD statement(s) identifying the production CDS are optional, and if included, must match the active HSC/VTCS.

- If HSC/VTCS is not active, `SLSCNTL` DD statement(s) identifying the production CDS are required.
- `SLSNEW` DD statement(s) identifying the DRTEST CDS(s) are required.

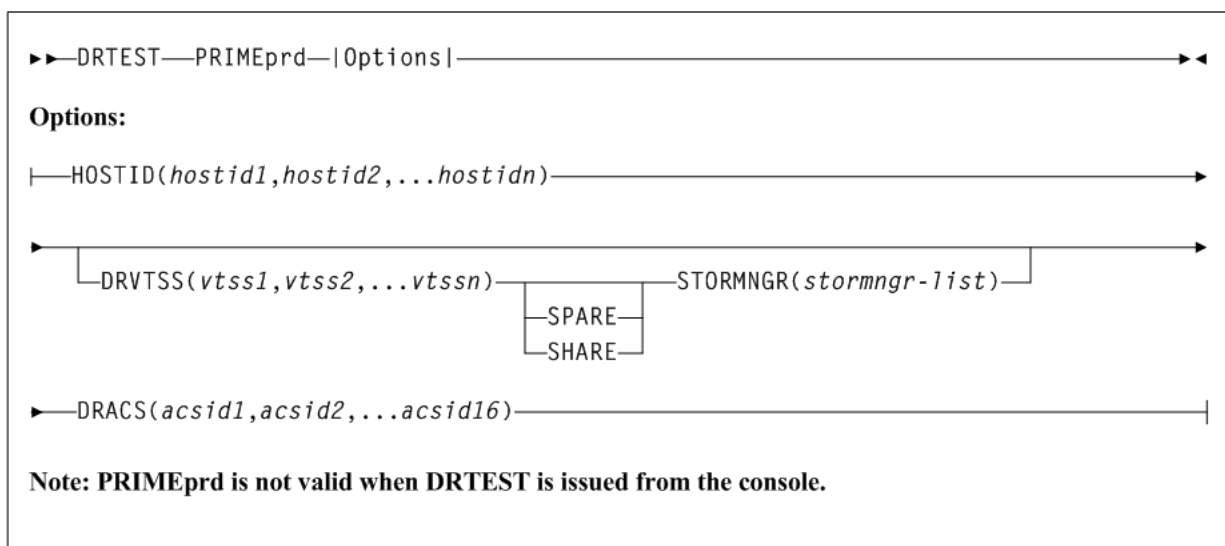
Sample Control Cards:

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

## Syntax

Figure 3–81 shows `DRTEST PRIMEprd` command syntax.

**Figure 3–81 DRTEST PRIMEprd syntax**



## Parameters

As shown in Figure 3–81, the `DRTEST PRIMEprd` command includes the following parameters:

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

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

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

Optionally, specifies the VTSS names available to the DR test site. Unless you also specify the `SHARE` or `SPARE` parameter, these VTSSs are offline to the production site and online to the DR test site when the DR test starts.

### SPARE

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

**SHARE**

Optionally, specifies that the DR test VTSS(s) are shared with the production site. During the DR test, both the test and production site may create VTVs in the DR test VTSS (based on the `VOLPARM` definitions). The DR test cannot modify VTVs that are not in a DR test subpool. The parameters `DRACS` and `STORMNGR` cannot be specified if `SHARE` is specified.

**STORMNGR(stormngr-list)**

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

*stormngr-list* indicates the list of Subsystem names. For more information, refer to Oracle's ELS publication *Configuring the Host Software for VLE*.

**DRACS(acsid1, acsid2,...acsid16)**

Optionally, specifies one or more ACSs available to the DR test site. Multiple ACS IDs must be specified as a list, and not as a range of ACS IDs. The ACSs are online to the DR test site but can remain connected (and usually are) to the production site when the DR test starts.

- If the `DRVTSS SHARE` parameter is specified, the `DRACS` parameter is not allowed.
- If the `STORMNGR` parameter is specified, the `DRACS` parameter is allowed but not required.

**DRTEST RESET****Interfaces:**

- `SLUADMIN` utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

**Description**

The `DRTEST RESET` command removes all DR test settings in the production CDS.

**Considerations:**

- DR test cannot be active. This refers to the production CDS status.
- Active HSC/VTCS is not required.
  - If HSC/VTCS is active, `SLSCNTL DD` statement(s) identifying the production CDS are optional, and if included, must match the active HSC/VTCS.
  - If HSC/VTCS is not active, `SLSCNTL DD` statement(s) identifying the production CDS are required.
- `SLSNEW DD` statement(s) identifying the `DRTEST` CDS(s) are required.

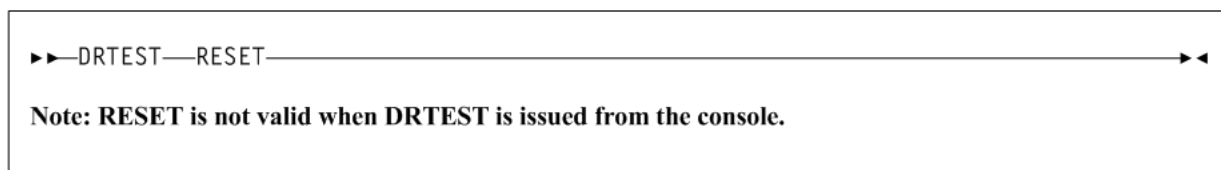
**Sample Control Cards:**

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

## Syntax

Figure 3–82 shows DRTEST RESET command syntax.

**Figure 3–82 DRTEST RESET command syntax**



## Parameters

None.

## DRTEST START

### Interfaces:

- console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC at FULL service level

### Description

The DRTEST START command starts the DR test on the production site.

---

**Note:** DRTEST START cannot run successfully until the production CDS is primed using DRTEST PRIMEprd or DRTEST CREATE.

---

### Considerations:

- DR test cannot be active. This refers to the production CDS status.
- Active HSC/VTCS is required.
- SLSCNTL DD statement(s) identifying the production CDS are optional, and if included, must match the active HSC/VTCS.
- SLSNEW DD statement(s) identifying the DRTEST CDS(s) are not required, and if included, are ignored.

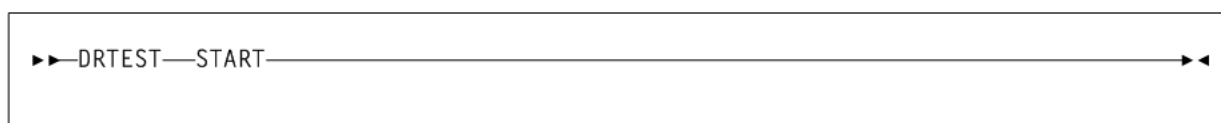
### Sample control cards:

```
DRTEST START
```

## Syntax

Figure 3–83 shows DRTEST START command syntax.

**Figure 3–83 DRTEST START command syntax**



**Parameters**

None.

**DRTEST STOP****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at FULL service level

**Description**

The DRTEST STOP command stops the DR test on the production site.

**Considerations:**

- DR test cannot be active. This refers to the production CDS status.
- Active HSC/VTCS is required.
- SLSCNTL DD statement(s) identifying the production CDS are optional, and if included, must match the active HSC/VTCS.
- SLSNEW DD statement(s) identifying the DRTEST CDS(s) are not required, and if included, are ignored.

**Sample control cards:**

DRTEST START

---



---

**Note:** DRTEST STOP and DRTEST RESET can be run in the same job, however, DRTEST STOP must be run before DRTEST RESET.

---



---

**Syntax**

Figure 3–84 shows DRTEST STOP command syntax.

**Figure 3–84 DRTEST STOP command syntax**

▶▶DRTEST—STOP————▶▶

**Parameters**

None.

**EEXPORT****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC not required

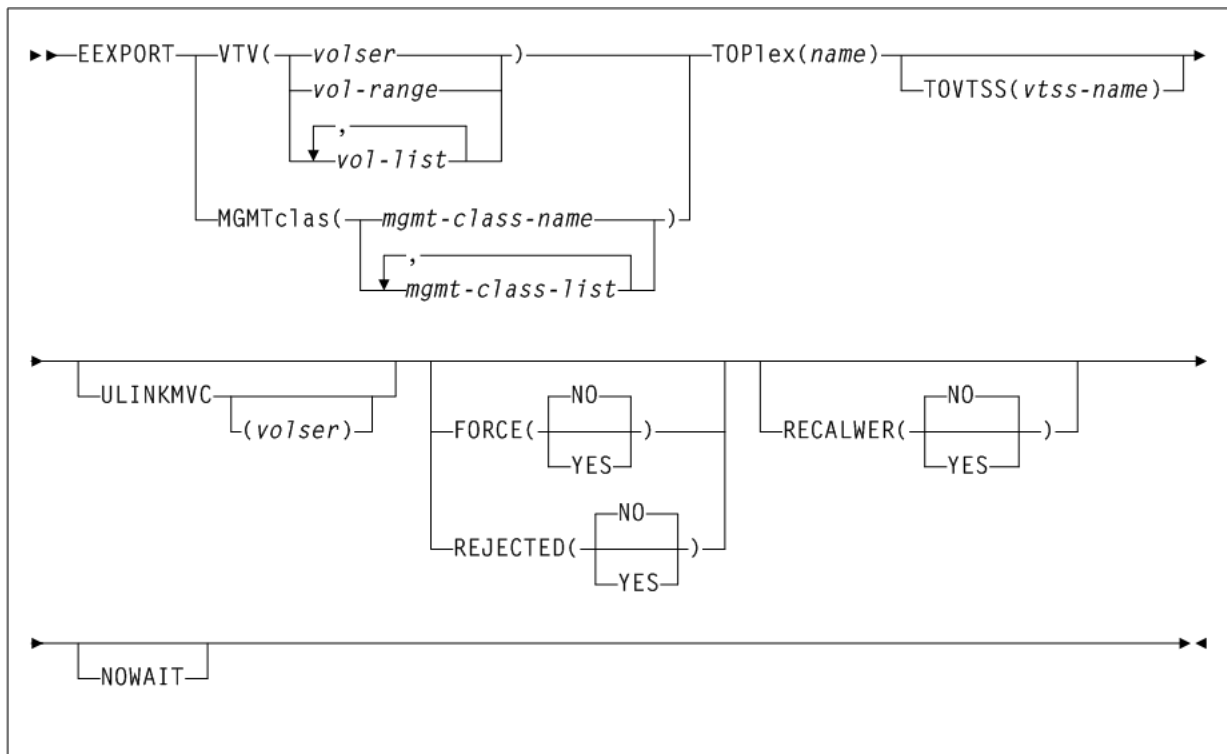
## Description

The EEXPORT command is used to perform electronic export functions. Use this command to manually export VTVs or re-drive rejected exports.

## Syntax

Figure 3–85 shows EEXPORT command syntax.

Figure 3–85 EEXPORT command syntax



## Parameters

As shown in Figure 3–188, the EEXPORT command includes the following parameters:

### VTV(volser, vol-range, or vol-list)

specifies one or more VTVs to electronically export.

*volser*, *vol-range*, or *vol-list* indicate the volsers of one or more VTVs. You can specify a maximum of 2,000 volsers.

### MGMTclas(mgmtclas-name or mgmtclas-list)

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

*mgmt-class-name* or *mgmt-class-list* indicates the Management Class name. This name must be 1 to 8 alphanumeric characters beginning with an alpha character and must follow SMS naming conventions.

**TOPlex(name)**

specifies the name of the TapePlex to which copies of the VTV are sent via electronic export. There must be at least one CLINK within the configuration that provides a route to the TapePlex.

*name* indicates the target TapePlex name.

**TOVTSS(vtss-name)**

optionally, specifies the VTSS name to be preferred as the receiving VTSS within the target TapePlex specified by the `TOPlex` parameter.

*vtss-name* indicates the VTSS name.

---

---

**Note:** This is a preference only. Validation of the value is not performed.

---

---

**ULINKMVC(volser)**

specifies whether MVC copies on the target TapePlex are to be deleted as part of the export process.

*volser* indicates a volser from the MVC copy to be deleted from the target VTVs. If not specified, then all MVC copies of the VTV are deleted.

This parameter only applies when the VTV version sent matches the version in the target TapePlex. If the versions sent is more recent, then the behavior is implicit.

This parameter can be used if the only MVC copies of a VTV in the target TapePlex become damaged. A replacement copy of the VTV can be supplied to the TapePlex via the `EEXPORT` command. The damaged MVC copies are replaced by additional migrations from the VTV copy sent.

**FORCE**

specifies whether VTCS exports the VTV to locations with disregard for other criteria.

**YES**

Export VTVs to the TapePlex specified in the `TOPlex` parameter, regardless of whether the VTVs have been previously exported to the TapePlex, or the management class points to a storage class that specifies the target TapePlex.

**NO**

Export VTVs to the TapePlex specified in the `TOPlex` parameter, but ignore any VTVs already marked as exported, and exclude any VTVs for which the management class does not point to a storage class that specifies the TapePlex in the `TOPlex` parameter. This is the default.

**REJECTED**

specifies whether VTCS only selects VTVs recorded in the CDS as being rejected by the target TapePlex. `REJECTED` and `FORCE` are mutually exclusive.

**NO**

Ignore any VTVs already marked as exported, and exclude any VTVs for which the management class does not point to a storage class that specifies the TapePlex in the `TOPlex` parameter. This is the default.

**YES**

Only select VTVs previously rejected by the target TapePlex.



**RECALWER**

specifies whether VTCS recalls VTVs with read data checks.

**NO**

Do not recall VTVs with read data checks. This is the default.

**YES**

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.

## Eject

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at FULL service level

## Description

The `EJECT` command moves volumes to a Cartridge Access Port (CAP) for removal.

You can designate a single CAP or a list of specific CAPs. Specifying one or more CAPs limits cartridge ejection to those CAPs.

Ejections in a multiple ACS configuration may specify CAPs in each ACS if volumes reside in each ACS. If a list of CAPs is not provided, HSC selects the highest priority CAP available for each ACS (see "[CAPPref](#)" on page 3-21).

You may elect to submit multiple `EJECT` requests, each of which may specify a particular CAP. If you elect to specify a CAP in a multi-ACS configuration, then the list of volumes must reside within the CAP ACS.

Multiple CAPs within one ACS can be allocated to the `EJECT` utility. Specifying multiple CAPs in a single `EJECT` utility typically improves performance by reducing pass-thrus.

If two CAPs are specified in the same LSM, cartridges in that LSM are ejected in sequential order. This is helpful for vaulting.

## Syntax

[Figure 3–86](#) shows `EJECT` command syntax.

Figure 3–86 Eject command syntax



## Parameters

As shown in [Figure 3–86](#), the Eject command includes the following parameters:

### TEXT(text)

optionally, specifies the address of a 44-byte string issued whenever the operator is requested to open the CAP for an Eject request. This text is displayed via message SLS1251E.

*text* indicates the text string.

### PMSG(nn)

optionally, specifies a user defined message to be displayed in the SLConsole CAP status screen for a CAP when the door is unlocked for cartridge removal.

*nn* is a two character numeric value from 00 to 99. Specify one of the following values:

- 00 - No message is displayed.

- 01 - Display Remove cartridges message.
- 02 - Display Lib not available message.
- 03 - Display Load cartridges message.
- 04-99 - Display customized user messages.

### Option 1 Parameters

As shown in Figure 3–86, the Eject command includes the following parameters:

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

one or more VOLSERS to be ejected. This can be a single VOLSER, a range of VOLSERS or a list of VOLSERS and/or VOLSER ranges in any combination.

If a list is specified, the elements must be delimited by commas or blanks, and the entire list enclosed in parentheses. A range of VOLSERS consists of a starting VOLSER and an ending VOLSER separated by a dash.

#### **acs-id, lsm-id, cap-id, or cap-list**

the ACS, LSM, or CAP for the cartridges to be ejected.

*acs-id* identifies the ACS containing the cartridges to eject. The HSC selects a non-zero preference CAP to use in the specified ACS.

*lsm-id* identifies the LSM containing the cartridges to eject. If the LSM has more than one CAP, an available CAP with the highest nonzero preference is selected.

*cap-id* identifies a specific CAP to use regardless of the assigned preference value. *cap-id* is of the form *AA:LL:CC*, where *AA:LL* indicates the LSMid, and *CC* indicates the CAP number. The CAP numbers are dependent on the LSM type.

*cap-list* requires explicitly specified CAPids separated by commas. A CAPid range is not allowed. CAPids must be separated by blanks or commas, and the entire list must be enclosed in parentheses. If no CAP is specified, one is chosen in each ACS.

---

**Note:** If neither *acs-id*, *lsm-id*, *cap-id*, or *cap-list* is specified, HSC selects a non-zero preference CAP in each ACS that contains volumes to be ejected.

---

#### **SEQ**

optionally, specifies whether or not CAP eject processing fills the CAP cells in the same order specified by the *volser*, *volser-range*, or *vol-list*.

If SEQ is not specified, but two CAPs are requested in the same LSM, and the LSM is not an SL8500, cartridges in the ACS are ejected in sequential order.

For best performance, SEQ(NO) is recommended.

#### **NO**

specifies to eject the requested volumes in the order of home cell location. Eject processing fills the CAP or magazine (for the SL8500) according to the distance of the home cell to the CAP or magazine; that is, volumes closest to the CAP or magazine are ejected first.

#### **YES**

specifies to eject cartridges to the CAP in the order the volumes are listed in the associated *vol-list*. The first cartridge requested appears in the topmost CAP

cell, the second cartridge requested appears in the next CAP cell, etc. until the CAP is full or all cartridges have been moved to the CAP.

**WAITcap**

optionally, specifies whether or not a list of ejecting volumes waits for an available CAP if one is not available.

**YES**

specifies that the eject process waits indefinitely for a CAP to become available. YES is the default.

**NO**

specifies that the eject process does not wait for a CAP if it is not available.

**Option 2 Parameters**

As shown in [Figure 3-86](#), the `Display Eject` command includes the following parameters:

**SCRATCH**

indicates that scratch volumes are to be ejected.

---



---

**Note:** If neither *acs-id*, *lsm-id*, *cap-id*, or *cap-list* is specified, HSC selects a non-zero preference CAP in each ACS that contains volumes to be ejected. HSC selects a non-zero preference CAP in ACS 00.

---



---

**acs-id, lsm-id, cap-id, or cap-list**

the ACS, LSM, or CAP for the cartridges to be eject.

*acs-id* identifies the ACS containing the cartridges to eject. The HSC selects a non-zero preference CAP to use in the specified ACS.

*lsm-id* identifies the LSM containing the cartridges to eject. If the LSM has more than one CAP, an available CAP with the highest nonzero preference is selected.

*cap-id* identifies a specific CAP to use regardless of the assigned preference value.

*cap-id* is of the form *AA:LL:CC*, where *AA:LL* indicates the LSMid, and *CC* indicates the CAP number. The CAP numbers are dependent on the LSM type.

*cap-list* requires explicitly specified CAPids separated by commas. A CAPid range is not allowed. CAPids must be separated by blanks or commas, and the entire list must be enclosed in parentheses. If no CAP is specified, one is chosen in each ACS.

**SUBpool(subpool-name)**

optionally, specifies the subpool from which scratch volumes are to be ejected. If `MEDIA` or `RECTech` are specified, cartridges are ejected for that media type or recording technique within the same subpool.

*subpool-name* indicates the name for the subpool.

**VOLCNT(count)**

optionally, specifies that a designated number of scratch volumes are to be ejected.

*count* indicates the number of scratch volumes to be ejected.

**MEDIA(media-type)**

optionally, specifies that scratch cartridges of the desired media are to be ejected.

*media-type* indicates the media type. See [Appendix A, "MEDia, RECtech, and MODel Values"](#) for a list of valid media-type values.

Note the following:

If `MEDia` is not specified, the next scratch cartridge is selected without regard to media type if `RECtech` does not exist. If both `MEDia` and `RECtech` are supplied, they must be compatible.

If `SUBpool` is not specified, then the next scratch cartridge is selected without regard to whether it belongs to a subpool, is defined to the default subpool using `VOLPARMS`, or belongs to no scratch subpool whatsoever.

#### **RECtech(recording-technique)**

optionally, specifies scratch cartridges of the desired recording technique are to be ejected. `RECtech` indicates the method used to record data tracks on the tape surface.

*recording-technique* indicates the recording technique. See [Appendix A, "MEDia, RECtech, and MODel Values"](#) for valid recording-technique values.

---



---

**Note:** If `RECtech` is not specified, the next scratch cartridge is selected depending on the media type (if supplied). If neither is supplied, the next scratch cartridge is selected without taking media type and recording technique into consideration. If both `RECtech` and `MEDia` are supplied, they must be compatible.

---



---

#### **WAITcap**

optionally, specifies whether or not a list of ejecting volumes waits for an available CAP if one is not available.

##### **YES**

specifies that the eject process waits indefinitely for a CAP to become available. YES is the default.

##### **NO**

specifies that the eject process does not wait for a CAP if it is not available.

## ENter

#### **Interfaces:**

- Console or utility
- UI Support: Yes

#### **Subsystem Requirements:**

Active HSC at `FULL` service level

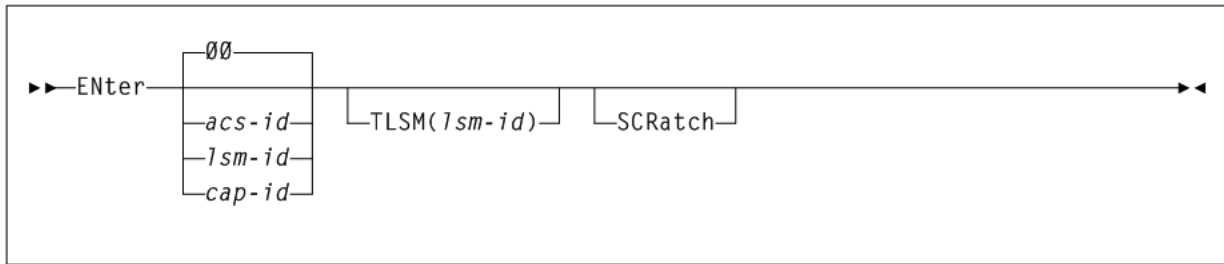
## Description

The `ENter` command makes a CAP in Automatic mode available for entering cartridges into an LSM.

## Syntax

[Figure 3–87](#) shows `ENter` command syntax.

Figure 3–87 ENter command syntax



## Parameters

As shown in [Figure 3–87](#), the ENter command includes the following parameters:

### acs-id

specifies the ACS where cartridges are to be entered. The HSC selects a nonzero preference manual mode CAP within the ACS to use.

### lsm-id

identifies one of the following:

- an LSM with a single CAP
- an LSM that contains multiple CAPs. The HSC selects an available manual mode CAP with the highest nonzero preference. CAPid 00 is selected if it is available, in manual mode, and has a preference value equal to or higher than other available manual mode CAPs in the LSM.

### cap-id

specifies a specific manual mode CAP to use regardless of availability or CAP preference. The format of a *cap-id* is *AA:LL:CC*, where *AA:LL* indicates the LSMid, and *CC* indicates the CAP. See "CAPid" on page B-1 for a list of valid values.

### TLSM(lsm-id)

specifies the LSM to receive the entered cartridges.

*lsm-id* indicates the following:

- an LSM with a single CAP
- an LSM that contains multiple CAPs. The HSC selects an available manual mode CAP with the highest nonzero preference. CAPid 00 is selected if it is available, in manual mode, and has a preference value equal to or higher than other available manual mode CAPs in the LSM.

The format of an LSMid is *AA:LL*, where *AA* indicates the ACSid (decimal 00-99) and *LL* indicates the LSM number (decimal 00-99).

### SCRatch

optionally enables you to put the volumes you enter into scratch status. If you do not specify that the volumes are to be given scratch status, the system enters them as nonscratch volumes.

## EXECParm

### Interfaces:

- PARMLIB only

- UUI Support: No

#### Subsystem Requirements:

None.

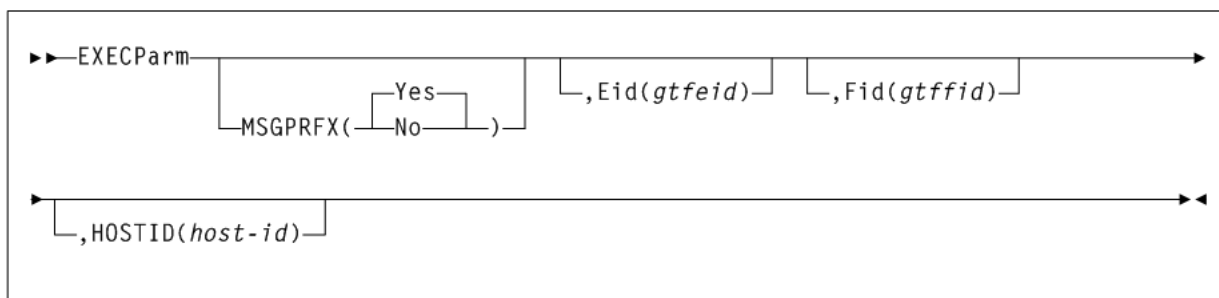
## Description

The EXECParm command provides an alternative method for specifying GTF event and format IDs normally specified in the subsystem startup procedure. It also provides you with the option of displaying the command prefix preceding WTO or WTOR messages.

## Syntax

Figure 3–88 shows EXECParm command syntax.

**Figure 3–88 EXECParm command syntax**



## Parameters

As shown in Figure 3–88, the EXECParm command includes the following parameters:

### MSGPRFX

optionally, specifies control of whether the command prefix is to precede WTO or WTOR messages to the operator.

#### Yes

indicates that the command prefix is to display preceding WTO or WTOR messages to the operator.

#### No

indicates that the command prefix is not to display preceding WTO or WTOR messages to the operator.

### Eid(gtfeid)

optionally, *gtfeid* specifies a GTF event ID.

PARAM='Eid(user-specified-event-id)' parameter is valid for use in the HSC initialization procedure as an alternative method of specifying the GTF event ID.

### Fid(gtffid)

optionally, *gtffid* specifies a GTF format ID.

PARAM='Fid(user-specified-format-id)' parameter is valid for use in the HSC initialization procedure as an alternative method of specifying the GTF format ID.

**HOSTID(host-id)**

optionally, *host-id* specifies the system ID associated with the request to process the EXECParm control statement.

---

---

**Note:** If the *host-id* specified does not match the host executing the command, the command is ignored and no message is issued.

---

---

## EXPORT

**Interfaces:**

- Utility only
- UUI Support: Yes

**Subsystem Requirements:**

- Active HSC/VTCS at FULL service level required when specifying the VTV or MGMTCLAS parameter.
- Active HSC/VTCS not required when specifying the MVC or STORCLAS parameter.

---

---

**Note:** The CDS used by the utility must not be accessed by any other currently active HSC/VTCS host(s). Otherwise, error message SLS6716E is issued and the utility fails.

---

---

## Description

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

---

---

**Note:**

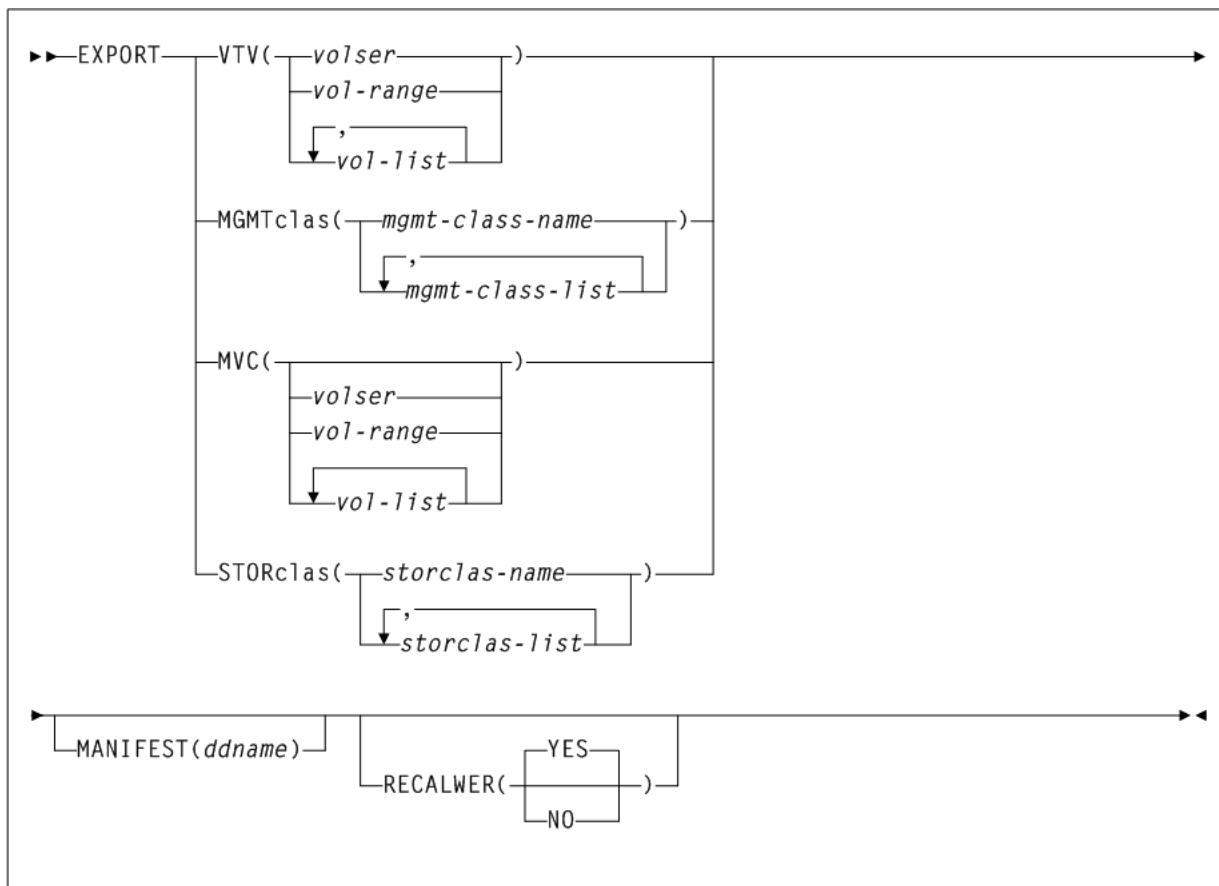
- EXPORT VTV and EXPORT MGMTCLAS cannot consolidate VTVs on vMVCs in a VLE.
  - EXPORT MVC and EXPORT STORCLAS can create a manifest for VTVs that have been migrated to vMVCs in a VLE.
- 
- 

## Syntax

Figure 3–89 shows EXPORT command syntax.



Figure 3–89 EXPORT command syntax



## Parameters

As shown in [Figure 3–69](#), the EXPORT command includes the following parameters:

### **VTV(volser, vol-range, or vol-list)**

specifies one or more VTVs to consolidate for export.

When an export by VTV is performed, the CDS does not maintain the MVC information in the VTV record. The MVC is marked as (E)port.

*volser*, *vol-range*, or *vol-list* indicates the volsers of one or more VTVs. You can specify an unlimited number of VTVs.

---

**Note:** EXPORT VTV cannot consolidate VTVs on vMVCs in a VLE.

---

### **MGMTclas(mgmtclas-name or mgmtclas-list)**

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

When an export by MGMTclas is performed, the CDS does not maintain the MVC information in the VTV record. The MVC is marked as (E)port.

*mgmt-class-name* or *mgmt-class-list* indicate the names of one or more Management Classes you defined on the MGMTclas control statement. See "[MGMTclas Control Statement](#)" on page 3-222 for more information.

---

---

**Note:** EXPORT MGMTclas cannot consolidate VTVs on VMVCs in a VLE.

---

---

**MVC(volser, vol-range, or vol-list)**

specifies one or more MVCs for export.

When an Export by MVC is performed, the CDS maintains the MVC information in the VTV record of the CDS. The MVC will be marked as (R) ead-Only.

*volser, vol-range, or vol-list* indicates the volsers of one or more MVCs.

**STORclas(stor-clas-name or stor-clas-list)**

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

When an Export by STORclas is performed, the CDS maintains the MVC information in the VTV record of the CDS. The MVC will be marked as (R) ead-Only.

*stor-clas-name* or *stor-clas-list* indicate the names of one or more Storage Classes that you defined on the STORclas control statement. See "[STORclas Control Statement](#)" on page 3-238 for more information.

---

---

**Note:** EXPORT STORclas can create a manifest for VTVs that have been migrated to vMVCs in a VLE.

---

---

**MANIFEST(ddname)**

optionally, specifies the output destination *dd-name* of the manifest file. The default is MANIFEST.

**RECALWER**

optionally, 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.

## Additional JCL Requirements

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statements apply to the EXPORT JCL:

**manifest file DD**

DD statement for the manifest file (optional).

## FMTLOG

**Interfaces:**

- SLUADMIN utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

## Description

The `FMTLOG` command is part of the HSC/VTCS transaction logging service.

This command pre-formats a log file to be used by the logging service. It pre-formats data blocks and writes certain control information in the first record of the file. You can format a maximum of two log files with a single `FMTLOG` utility command.

---

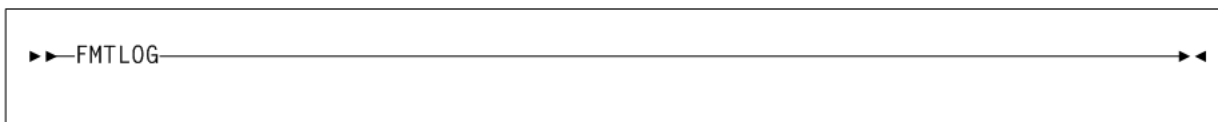
**Note:** You must pre-format a log file before activating it with the `SET LOGFILE` utility command. See "[SET LOGFILE](#)" on page 3-325 for more information. You must execute the `OFFload LOGfile` before executing `SET LOGFILE` to specify a different log file data set name. See "[OFFload](#)" on page 3-286 for more information.

---

## Syntax

Figure 3-90 shows `FMTLOG` command syntax.

**Figure 3-90** *FMTLOG command syntax*



## Parameters

None.

## Additional JCL Requirements

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statements apply to the `FORMAT LOGFILE` JCL:

### **SLSLOG1**

specifies the log file to be formatted.

### **SLSLOG2**

specifies the second log file to be formatted.

## IMPORT

### **Interfaces:**

- Utility only
- UI Support: Yes

### **Subsystem Requirements:**

Active HSC/VTCS not required

## Description

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

---

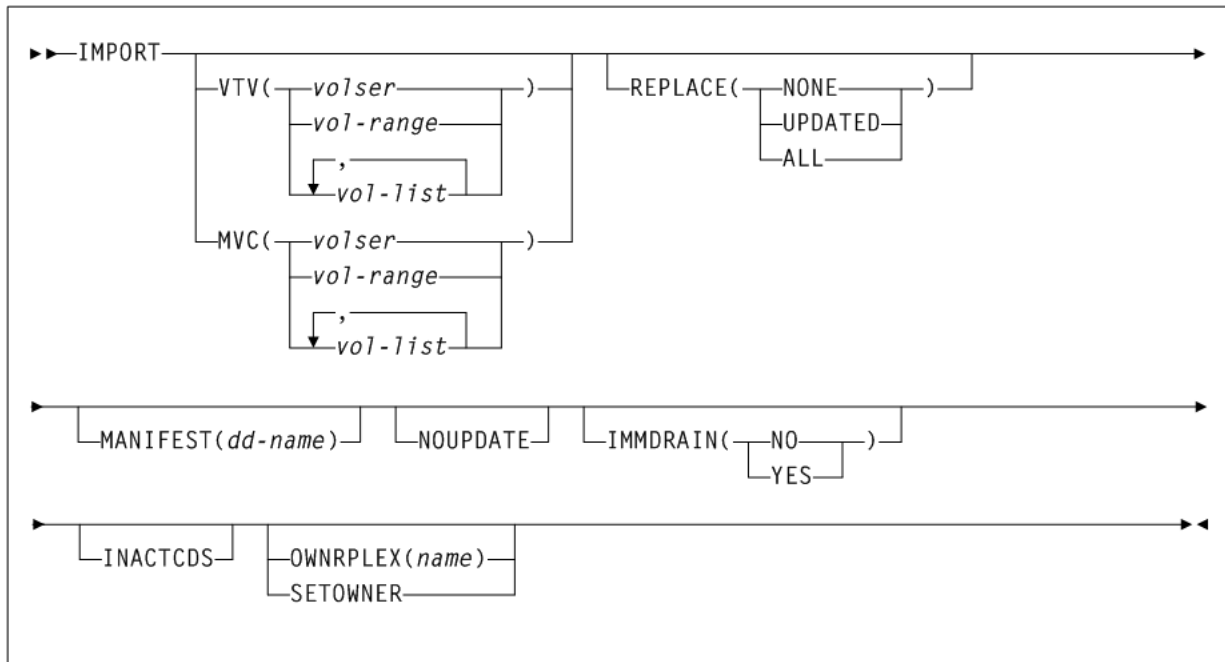
**Note:** Ensure that the “to” CDS has the same features (enabled by CDS level) as the “from” CDS. For example, if the “from” CDS has large VTV page sizes enabled and 2/4 Gb VTVs have been created, then the “to CDS” must have the same capabilities, otherwise the import fails.

---

## Syntax

Figure 3–91 shows `IMPORT` command syntax.

**Figure 3–91** *IMPORT* command syntax



## Parameters

As shown in Figure 3–91, the `IMPORT` command includes the following parameters:

### **VTV**(*volser*, *vol-range*, or *vol-list*)

specifies one or more VTVs to import.

*volser*, *vol-range*, or *vol-list* indicate the volsers of one or more VTVs.

### **MVC**(*volser*, *vol-range*, or *vol-list*)

specifies one or more MVCs to import.

*volser*, *vol-range*, or *vol-list* indicate the volsers of one or more MVCs.

### **REPLACE**

specifies whether VTCS 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.

**UPDATED**

Replace the VTV record if it is a more recent version of the VTV, or if the CDS has no migrated or resident versions of the VTV. If it is an older or the same version, do not replace the VTV record.

**ALL**

Replace any duplicate VTV records in the target CDS.

Considerations:

- 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.
- If a VTV record is replaced, all existing VTSS and MVC copies of the VTV are invalidated.
- 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.

**MANIFEST(dd-name)**

optionally, specifies the input dd name of the manifest file.

*dd-name* indicates the dd name of the manifest file. The default is `MANIFEST`.

**NOUPDATE**

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

**IMMDRAIN**

optionally, specifies whether VSM will immediately drain imported MVCs.

**NO**

Do not drain MVCs (the default).

**YES**

Drain MVCs.

**INACTCDS**

optionally, specifies that the import uses a different CDS from the CDS currently active on the HSC system where you are running the import job. Use the `SLSCNTL` definition statement in the `SLUADMIN JCL` to specify the alternate CDS.

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`.

If you specify the `INACTCDS` parameter, the CDS specified in the `SLSCNTL` definition statement in the `SLUADMIN JCL` must be different from the CDS being used by HSC/VTCS (if active).

**OWNRPLEX(name)**

specifies ownership of a VTV being imported.

*name* indicates the name of the TapePlex that owns the original copy of the VTV. This name must match one of the known TapePlex names in the configuration.

If this name matches that indicated for the `THISPLEX` parameter on the `CONFIG TAPEPLEX` statement, the electronic imported status is removed.

**SETOWNER**

specifies that ownership information is to be imported.

By default, any ownership information within the manifest is ignored. The `SERTOWNER` parameter specifies that the TapePlex ownership and export fields are imported. VTVs appear as if electronically imported on `Display` commands and reports.

**Additional JCL Requirements**

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statements apply to the `IMPORT JCL`:

**manifest file DD**

DD statement for the manifest file.

**INITialize****Interfaces:**

- `SLUADMIN` utility only
- UUI Support: No

**Subsystem Requirements:**

HSC at `FULL` service level

**Description**

The `INITialize` command enables you to batch enter cartridges into the library through the `CAP`, and invokes a user-specified program to write a magnetic label on each tape.

The robot accepts cartridges placed into the `CAP` and reads the Tri-Optic labels, and the HSC passes the `VOLSERs` to the user-specified program. The magnetically labeled cartridges can be ejected from the library or placed into cell locations. If the cartridges are stored in the library, the control data set is updated with the `VOLSER` and location information. The control data set can mark each cartridge as either scratch or nonscratch.

`INITialize` reads the external Tri-Optic labels and records them in a data set defined through the `CNTLDD` parameter. `CNTLDD` is a control statement parameter that describes the `DDname` to be used for `TMSTPNIT` control statements.

The `INITialize` utility does not initialize cartridges on its own; it invokes `TMS`, `TLMS`, or any other initialization utility, which performs the initialization. `CNTLDD` specifies, through the HSC to the `TMS` utility, the `DD` to be used for the input parameters the `TMS` utility requires.

---

---

**Caution:** This utility must be executed on an MVS system running SMC and using a local HSC server.

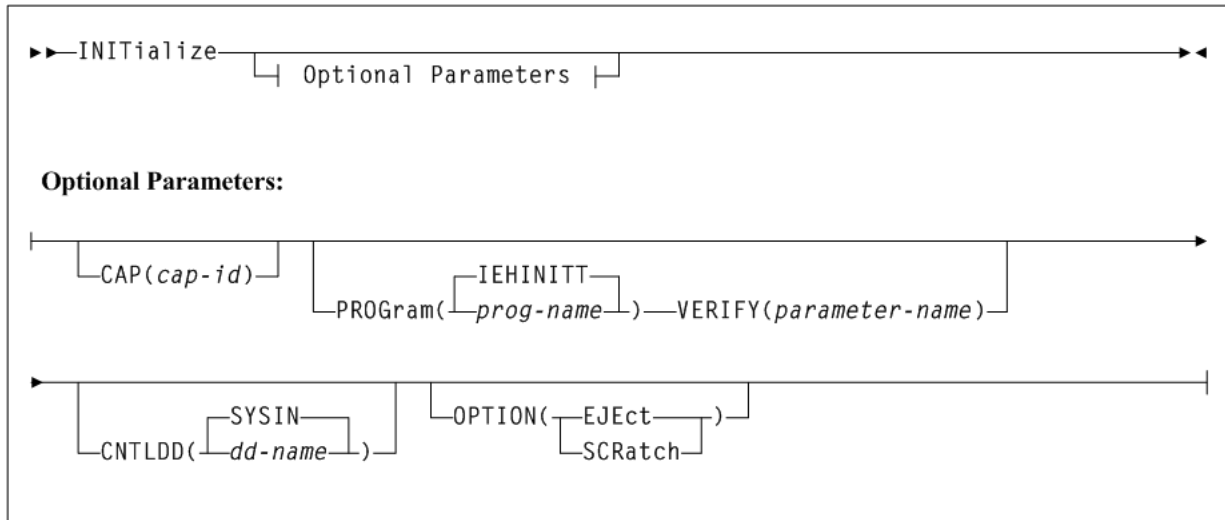
---

---

**Syntax**

[Figure 3-92](#) shows `INITialize` command syntax.

Figure 3–92 INITIALize command syntax



## Parameters

As shown in Figure 3–69, the INITIALize command includes the following parameters:

### CAP(cap-id)

optionally, specifies a particular CAP to be used for the operation.

If CAP is not specified, the utility chooses a CAP in the ACS implied by the device allocated to satisfy the SLSTAPE DD statement.

*cap-id* indicates the CAP. The format of a *cap-id* is *AA:LL:CC*, where *AA:LL* indicates the LSMid, and *CC* indicates the CAP. See "CAPid" on page B-1 for a list of valid values.

### PROGram(prog-name or IEHINITT)

optionally, specifies a program to be invoked to actually write the label (if not specified, IEHINITT is assumed).

*prog-name* indicates the program that is to perform the initialization.

### VERIFy(parameter-name)

optionally, specifies a parameter to be passed to the tape initialization program identified by the PROGram parameter. VERIFY is valid only when the PROGram parameter specifies TMSTPNIT.

*parameter-name* indicates the parameter passed to the TMSTPNIT program. For example, if the following parameter is entered:

```
VERIFY (ROBOT)
```

TMSTPNIT suppresses the TMS CAL0TN01 message for NL (nonlabeled) tapes. Refer to the appropriate CA-1 publication for valid parameter values.

### CNTLDD(dd-name or SYSIN)

optionally, specifies the DD statement to which initialization control statements are written, and from which the label program reads its control statements.

---

**Note:** This is not your control data set.

---

*dd-name* indicates the name of the DD statement. The *dd-name* option may only be specified when using CA-1 (TMS) prior to version 5.1. The default value (*SYSIN*) MUST be used with CA-1 (TMS) version 5.1 or later.

**OPTION**

optionally, specifies optional handling to be performed following the completion of label processing. If not specified, the default is retention of the new cartridges in the library with a non-scratch status.

**EJEct**

specifies that the labeled volumes are to be ejected from the library after the labeling process is complete.

**SCRatch**

specifies that the labeled volumes entering the library are to be added to the scratch list. If the volume is to be treated as scratch, the installation must inform the tape management system of the volume's scratch status.

## Additional JCL Requirements

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statements apply to the *INITIALIZE* JCL:

**SLSINIT**

substitution control card format for the invoked initialization program.

**SLSTAPE**

library transport on which the cartridge is mounted for labeling. To prevent a nonlibrary transport from being allocated, specify one of the following:

- an esoteric containing only library transports
- a specific library transport address
- a *TAPEREQ* statement to direct the allocation to a library transport
- a user exit (refer to Oracle's ELS publication *ELS Legacy Interfaces Reference*)

**CNTLDD or SYSIN**

contains volume serial labeling information and is passed as input to the initialization program (*IEHINITT* or the program specified by the *PROG* keyword parameter).

If *CNTLDD* is specified in the utility control statement, then the DD name identifies the data set containing the labeling information.

**SYSPRINT**

output messages from *IEHINITT*.

contains volume serial labeling information and is passed as input to the initialization program (*IEHINITT* or the program specified by the *PROG* keyword parameter).

If *CNTLDD* is specified in the utility control statement, then the DD name identifies the data set containing the labeling information.

## INVENTORY

**Interfaces:**

- Utility only



- UI Support: Yes

#### Subsystem Requirements:

Active HSC/VTCS

## Description

The `INVENTORY` command lists all VTVs on specified MVCs.

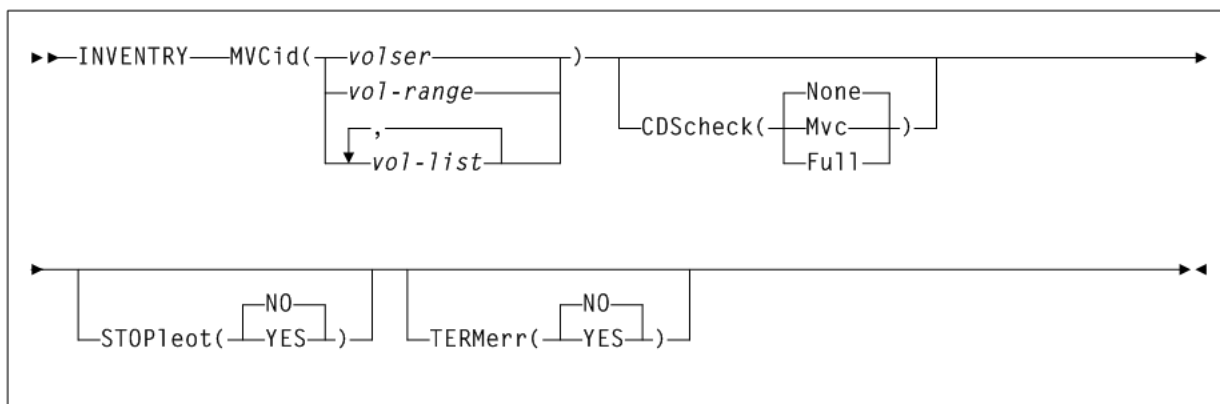
Additionally, you can use this command to do the following:

- Request that the contents of the specified MVCs are cross-checked with the information recorded in the CDS.
- Terminate 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.
- Terminate processing for an MVC the first time a VTV is discovered that is incompatible with the information in the CDS.

## Syntax

Figure 3–93 shows `INVENTORY` command syntax.

Figure 3–93 `INVENTORY` command syntax



## Parameters

As shown in Figure 3–93, the `INVENTORY` command includes the following parameters:

#### **MVCid(volser, vol-range, or vol-list)**

specifies the MVC(s) to be inventoried.

*volser*, *vol-range*, or *vol-list* indicates the volsers of one or more MVCs. You can also specify one or more ranges.

#### **CDScheck**

optionally, 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` indicates 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 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 that differs from the copy listed in SLS6933I, its details are listed via message SLS6934I. For each inventoried MVC, the CDS is checked to determine which CTCs are shown to have a copy on the MVC. Any MVCs that were not found on the MVC are listed in message SLS6935E.

**STOPIeot**

optionally, 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. This is the default.

**YES**

VTVs are processed on each inventoried MVC until one of the following occurs:

- The MVC's physical end-of-tape is reached.
- 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. This setting is invalid with CDScheck (None).

**TERMerr**

optionally, 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. This 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. This setting is invalid with CDScheck (None).

## Return Codes

The INVENTORY command includes the following return codes:

- 0 indicates that all requested updates completed successfully.
- 4 indicates that one or more errors were found. Errors are listed via message SLS6935E, SLS6936E, SLS6938E, SLS6939E, SLS6940E or SLS6941E.

- 8 indicates that one or more errors were found that are not specific to INVENTORY processing. For example, ECAM errors.

## Inventory Report

The following figure shows an example of an INVENTORY report produced by the following command:

```
INVENTORY MVC(021549)
```

In this example, the customer wishes to list all VTVs on MVC 021549 without cross-checking the VTVs found with the information recorded in the CDS.

In the sample output, each VTV found on the inventoried MVC is listed in message SLS6933I.

### Example 3–50 INVENTORY report

Inventory report for MVC 021549

```
SLS6933I MVC 021549 block 00000000: VTV Y00486 Created 2007Jul17 06:33:22 Migrated 2007Jul17
SLS6933I MVC 021549 block 05402F10: VTV Y00487 Created 2007Jul17 06:34:09 Migrated 2007Jul17
SLS6933I MVC 021549 block 0A405E1F: VTV Y00489 Created 2007Jul17 06:34:56 Migrated 2007Jul17
SLS6933I MVC 021549 block 0F408D2E: VTV Y00493 Created 2007Jul17 06:36:34 Migrated 2007Jul17
SLS6933I MVC 021549 block 10409E3C: VTV Y00492 Created 2007Jul17 06:35:46 Migrated 2007Jul17
```

## LIBGen

### Interfaces:

- SLUADMIN utility only
- UUI Support: No

### Subsystem Requirements:

Active HSC not required

## Description

The LIBGen command decompiles the hardware configuration, VOLPARM, VAULT and LOGFILE information from the Control Data Set and creates a set of LIBGEN macros that reflects that information. The LIBGen decompile utility can be used to generate an initial set of LIBGEN macros that can be modified when you are adding new hardware to your installation that requires the creation of a new CDS.

Refer to Oracle's ELS publication *Configuring HSC and VTCS* for detailed information about when to use this utility and how it functions.

## Syntax

Figure 3–94 shows LIBGen command syntax.

Figure 3-94 LIBGen command syntax



## Parameters

As shown in [Figure 3-94](#), the LIBGen command includes the following parameters:

### TAPELESS

Optionally, provides a method to generate CDScreat syntax from a CDS that currently contains library definitions, or to create LIBGEN macro syntax from a CDS currently defined as tapeless.

#### NO

Create CDScreat syntax from a CDS that currently contains library definitions.

#### YES

Create LIBGEN macro syntax from a CDS currently defined as tapeless.

If the TAPELESS parameter is omitted, the default is based on the input CDS; that is, a tapeless CDS generates CDScreat syntax and a CDS with library definitions generates LIBGEN macro syntax. When TAPELESS(NO) is specified for a tapeless CDS, ACSs and LSMs must be defined prior to using the LIBGEN macros to define a new CDS.

## Additional JCL Requirements

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statements apply to the LIBGen JCL:

### SLSLIBGN

output data set to accommodate the LIBGEN created by the utility.

The data set has these characteristics: LRECL=80, fixed-blocked format (multiple of 80). The data set can be assigned to print or to output to a DASD. It may be assigned to DUMMY if only the SLSPRINT output is desired.

### SLSPARM

Output VOLPARM card images from CDS.

SLSPARM is required when VOLPARM records exist in the CDS. If the SLSPARM DD is not coded in the JCL, message SLS0212I is issued and the utility ends with a return code 4.

### SLSSET

Output CDScreat, SET VOLPARM, SET VAULT or SET VAULTVOL, and/or SET LOGFILE control statements from the CDS. If the LIBGEN utility attempts to output one of the above types of control statements and the SLSSET DD statement is missing, a message SLS0212I is issued. If the statement to be produced is CDScreat, the utility terminates with a return code of 8; otherwise, processing continues and a return code of 4 is generated.

## Output

LIBGen command output includes the following:

- a valid LIBGEN file matching the existing control data set. The output file has the following characteristics:
  - All station and drive addresses are 4-character addresses.
  - In cases where multiple parameters point to the same label statement, the utility duplicates the statement with a unique label and points each parameter to a different, although identical statement.
  - Labels generated in the output LIBGEN are listed in Table 3–6, below.
  - If an EJECT password exists, it is not displayed. Instead, the following line is displayed:
 

```
EJCTPAS=???????
```
- messages associated with error conditions resulting from an unsuccessful execution of the utility.

**Table 3–6 LIBGen Output Labels**

Device	Label	Description
ACS	ACSaa	aa indicates the sequential decimal ACSid value (00 through 99), beginning with zero.
LSM	LSMaall	aa indicates the ACSid, and ll indicates the sequential decimal LSMid value (00 through 99), beginning with zero.
STATION	STaah	aa indicates the ACSid, and h indicates the sequential hexadecimal host index value (0 through F), beginning with zero.
PANEL	Paallpp	aa indicates the ACSid, ll indicates the LSMid, and pp indicates the sequential decimal panel number (0 through 10).
DRIVE	Daallpph	aa indicates the ACSid, ll indicates the LSMid, pp indicates the decimal panel number, and h indicates the hexadecimal host index value, beginning with zero.

## LMUPDEF

### Interfaces:

- Console or PARMLIB
- UUI Support: No

### Subsystem Requirements:

Active HSC at BASE or FULL service level

## Description

The LMUPDEF command specifies the definition data set that contains network LMU attachment (LMUPATH) statements.

If a definition file is changed by a LMUPDEF operator command, and if the change is to be permanent, you must update PARMLIB before restarting the HSC.

If the definition file is to be modified for a given shift or application, remember that the new definition remains in effect until another definition is loaded or the HSC is

recycled. The definition file then reverts to the PARMLIB specification. Thus, LMUPDEF can be used to temporarily change a definition file.

---



---

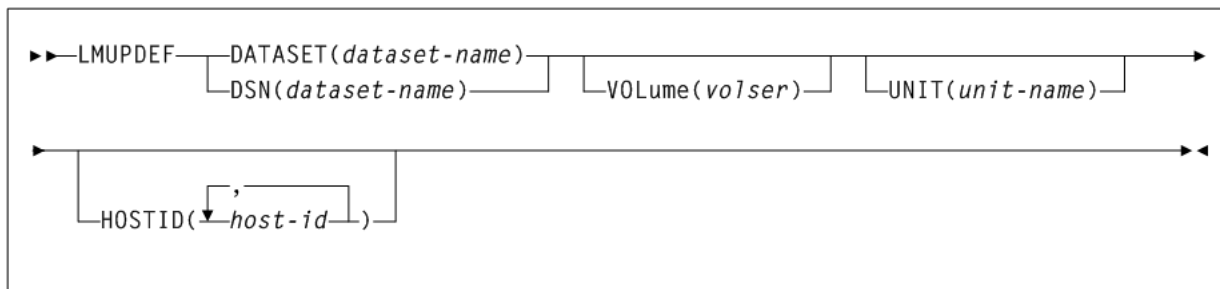
**Note:**

- Definition commands issued on one host are in effect only on that host. If different hosts use the same definition data set, the LMUPATH parameter statements are shared by those hosts.
  - If you issue multiple LMUPDEF commands or statements, the last one processed is currently active. You can determine which LMUPDEF statement is active by entering the `Display LMUPDEF` command.
- 
- 

## Syntax

Figure 3–95 shows LMUPDEF command syntax.

**Figure 3–95 LMUPDEF command syntax**



## Parameters

As shown in Figure 3–95, the LMUPDEF command includes the following parameters:

**DATASET(dataset-name) or DSN(dataset-name)**

specifies the name of the data set containing the LMUPATH statements to be processed and, optionally, an `OPTION TITLE` statement.

*dataset-name* indicates the name of the data set containing the LMUPATH address parameters. If the data set name includes a member name, *dataset.name* must be enclosed in quotes. For example:

```
DATASET('YOUR.DSN(MEMBER)')
```

---



---

**Note:**

- The definition data set may contain `VOLATTR`, `UNITATTR`, `TAPEREQ`, `LMUPATH`, and `OPTION TITLE` statements, but only `LMUPATH` and `OPTION TITLE` statements are processed.
  - If any other statement is encountered, an error message is issued and the statement is ignored.
  - See "[OPTION TITLE Control Statement](#)" on page 3-287 for more information about the `OPTION TITLE` control statement.
- 
-

**VOLume(volser)**

optionally, specifies the serial number of the DASD volume on which the data set resides. This parameter is optional. Specify the VOLume parameter if the data set is not cataloged, or if a data set on a volume other than the volume indicated by the catalog is to be used.

*volser* indicates the volume serial number for the definition data set.

**UNIT(unit-name)**

optionally, specifies the unit where the definition data set is located.

*unit-name* indicates the unit name. If the definition data set is not cataloged and this parameter is omitted, a unit name of SYSALLDA is the default.

**HOSTID(host-id)**

optionally, limits the execution of this control statement to the specified hosts. If one of the specified *host-ids* matches the host executing this control statement, the control statement is executed for that host. Otherwise, it is ignored. If this parameter is omitted, the control statement is executed on all hosts.

This parameter is valid only for use in PARMLIB, so that multiple systems can share a PARMLIB member containing TAPEREQ, VOLATTR, UNITATTR, or LMUPATH statements for different releases of HSC. If entered from the console, message SLS0018I is issued.)

*host-id* indicates the name of one or more hosts from which to execute this control statement. Multiple hosts must be separated by commas.

## LMUPATH Control Statement

The LMUPATH control statement defines network LMU/Library Controller (LC) attachments. It is loaded by the LMUPDEF command.

---

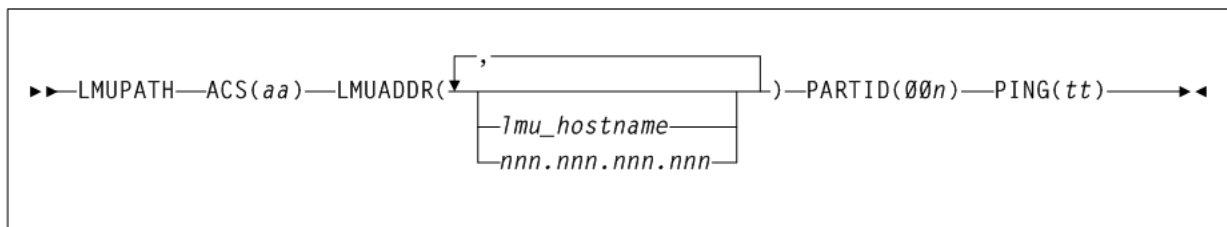
**Note:** If you are using the SL3000 or SL8500 partitioning feature, the PARTID parameter connects to a specific partition defined by the SL3000 or SL8500 library for the HSC host group. Refer to Oracle's ELS publication *Configuring HSC and VTCS* for information about how to use the partitioning feature.

---

### Syntax

Figure 3–96 shows LMUPATH control statement syntax.

**Figure 3–96** LMUPATH control statement syntax



### Parameters

As shown in Figure 3–96, the LMUPATH control statement includes the following parameters:

**ACS(aa)**

specifies the ACS name.

*aa* indicates the decimal ACSid value (00 through 99) to identify the LMU(s) used to communicate with a HSC.

**LMUADDR**

identifies an LMU/Library Controller (LC) by host name or IP address, for each ACS. To designate a single LMU/LC environment, specify one IP address or host name. To specify a dual LMU/LC environment or a dual IP connection to an SL8500, enter an additional IP address and/or host name.

The HSC automatically detects the type of connection, dual LMU for a 9330 or dual IP for an SL8500 library.

Users can intermix host name and IP addresses in one `LMUPATH` control statement. Each parameter entered must represent a different IP address.

**Imu-hostname**

a host name for the TCP/IP connection. The host name can be up to 24 characters long. The first character must be alphabetic.

**nnn.nnn.nnn.nnn**

an IP address for the LMU/LC. A maximum of 40 IP addresses can be specified.

**PARTID(00n)**

defines a partition ID for each ACS.

*00n* indicates a partition ID from 001 to 999.

- All three characters must be entered for the partition ID.
- For this release, only IDs 001 through 008 are supported.

---

---

**Note:** A partitioned SL3000 or SL8500 cannot use the multiple TCP/IP connection feature since partitioning applies only to a single SL3000 or SL8500 box.

---

---

**PING(tt)**

specifies the number of minutes in between requests sent from the HSC to the LMU. These requests are to keep the connection active, which prevents a firewall from closing the connection due to inactivity.

*tt* indicates the time in minutes from 00-99. The default is 5 minutes if this parameter is not defined, and entering 00 turns off this feature.

## LOGUTIL

**Interfaces:**

- SLUADMIN utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required



## Description

The LOGUTIL command initiates the CDS recovery process for VTCS. It is used in the following scenarios:

- The CDS has become unusable or corrupt and MVCs used since the date of the CDS backup need to be identified for MVCAUDIT purposes.
- A VTSS has suffered a catastrophic data loss.

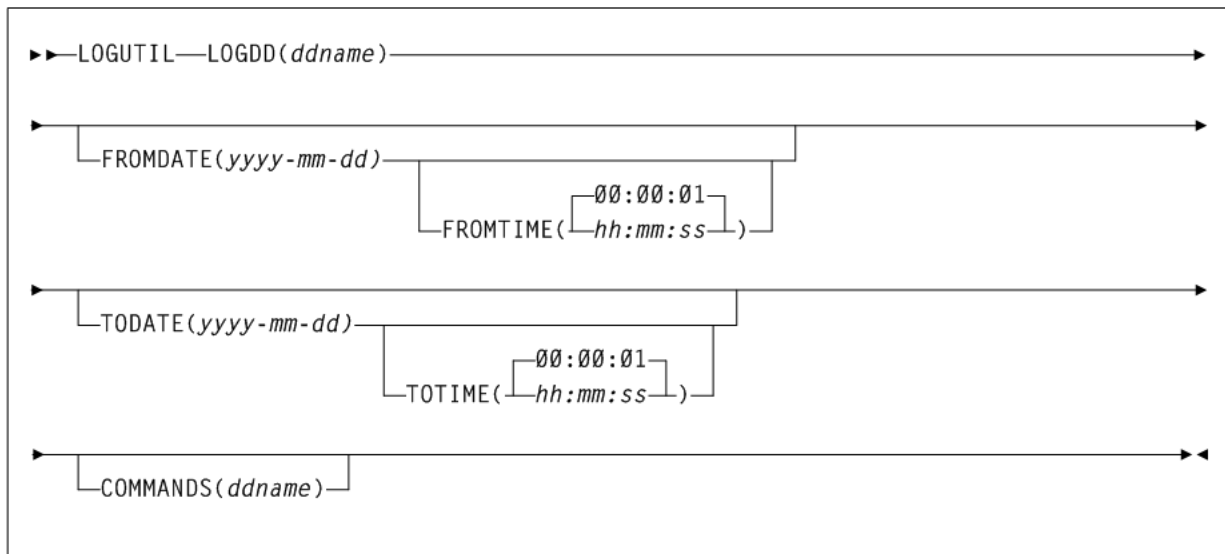
LOGUTIL calls the module that analyzes the input log files, and calls a recovery module to perform the appropriate recovery actions. Additionally:

- The LOGUTIL FOR\_LOSTMVC statement recovers VTVs that resided on a lost or damaged MVC.
- The LOGUTIL GENAUDIT statement initiates a re-synchronization of the CDS to VSM via MVCAUDIT statements.
- The LOGUTIL LOCATE\_VTV statement recovers older versions of VTVs.
- The LOGUTIL UNDELETE statement recovers deleted VTVs.

## Syntax

Figure 3–97 shows LOGUTIL command syntax.

**Figure 3–97 LOGUTIL command syntax**



## Parameters

As shown in Figure 3–97, the LOGUTIL command includes the following parameters:

### LOGDD(ddname)

specifies the DD name of the input log data sets.

*ddname* indicates the DD name.

### FROMDATE(yyyy-mm-dd)

optionally, specifies a starting date for logged events.

*yyyy-mm-dd* indicates the starting date.

---



---

**Note:** For FROMDATE and FROMTIME, specify local dates/times as seen from the MVS system on which LOGUTIL runs.

---



---

**FROMTIME(hh:mm:ss)**

optionally, specifies a starting time for logged events.

*hh:mm:ss* indicates the starting time. The default is 00:00:00.

**TODATE(yyyy-mm-dd)**

optionally, specifies an ending date and time for logged events.

*yyyy-mm-dd* indicates the ending date.

---



---

**Note:** For TODATE and TOTIME, specify local dates or times as seen from the MVS system on which LOGUTIL runs.

---



---

**TOTIME(hh:mm:ss)**

optionally, specifies an ending time for logged events.

*hh:mm:ss* indicates the ending time. The default is 00:00:01.

**COMMANDS(ddname)**

optionally, specifies the DD name of a data set that contains recovery commands.

*ddname* indicates the DD name.

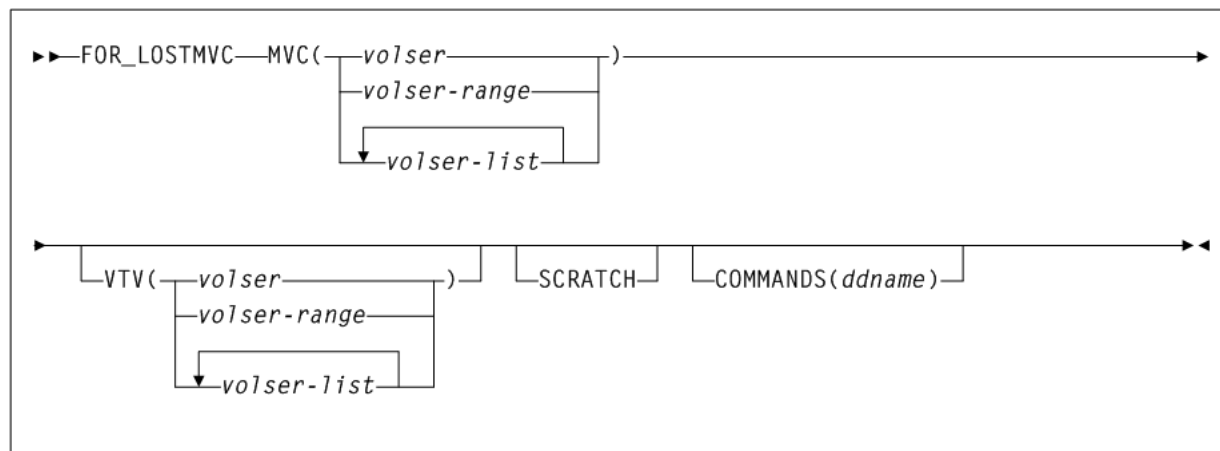
## LOGUTIL FOR\_LOSTMVC Statement

The LOGUTIL FOR\_LOSTMVC statement recovers VTVs that resided on lost or damaged MVCs.

**Syntax**

Figure 3-98 shows LOGUTIL FOR\_LOSTMVC statement syntax.

**Figure 3-98 LOGUTIL FOR\_LOSTMVC statement syntax**

**Parameters**

As shown in Figure 3-98, the LOGUTIL FOR\_LOSTMVC statement includes the following parameters:

**MVC(volser, volser-range, or volser-list)**

specifies the MVC(s) to be recovered. You can specify a maximum of 1000 MVCs.

*volser*, *volser-range*, or *volser-list* indicates the volser, volser range, or volser list to be processed.

**VTV(volser, volser-range, or volser-list)**

optionally, specifies specific VTV(s) to be recovered. You can specify a maximum of 300,000 MVCs.

Specified VTVs must reside on one or more of the recovery MVCs to be processed. When this parameter is coded, only VTVs specified are recovered (if they exist on the recovery MVCs).

*volser*, *volser-range*, or *volser-list* indicates the volser, volser range, or volser list to be processed.

This parameter behaves as a filter. If VTVs are specified that are not on the specified MVCs, they will not be recovered and will not be listed in the report output.

**SCRATCH**

optionally, specifies to specify scratch VTVs. By default, scratch VTVs are not processed as part of the recovery.

**COMMANDS(ddname)**

optionally, specifies the DD name of a data set that includes the recovery commands created when the COMMANDS parameter is specified on the LOGUTIL command.

*ddname* indicates the DD name.

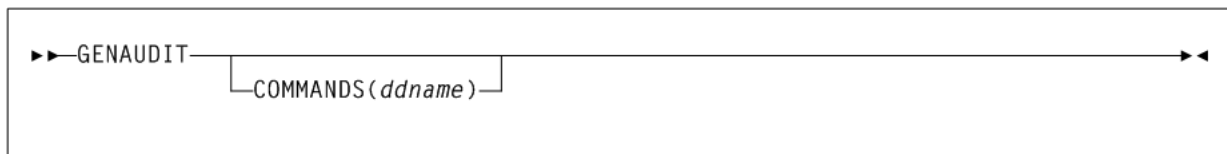
## LOGUTIL GENAUDIT Statement

The LOGUTIL GENAUDIT statement initiates a re-synchronization of the CDS to VSM through MVCAUDIT statements that are generated when the COMMANDS parameter is specified with the LOGUTIL command. The LOGUTIL GENAUDIT statement identifies MVCs for input to MVCAUDIT.

### Syntax

Figure 3-99 shows LOGUTIL GENAUDIT control statement syntax.

**Figure 3-99 LOGUTIL GENAUDIT statement syntax**



### Parameters

As shown in Figure 3-99, the LOGUTIL GENAUDIT statement includes the following parameters:

**COMMANDS(ddname)**

optionally, specifies the DD name of a data set that includes the recovery commands created when the COMMANDS parameter is specified on the LOGUTIL command.

*ddname* indicates the DD name.

The GENAUDIT COMMANDS parameter works with the LOGUTIL COMMANDS parameter as follows:

- If you specify LOGUTIL COMMANDS, the GENAUDIT statement outputs recovery commands to the specified data set even if you did not specify GENAUDIT COMMANDS.
- If you specify COMMANDS on both LOGUTIL and GENAUDIT, recovery commands are only output to the GENAUDIT COMMANDS data set.

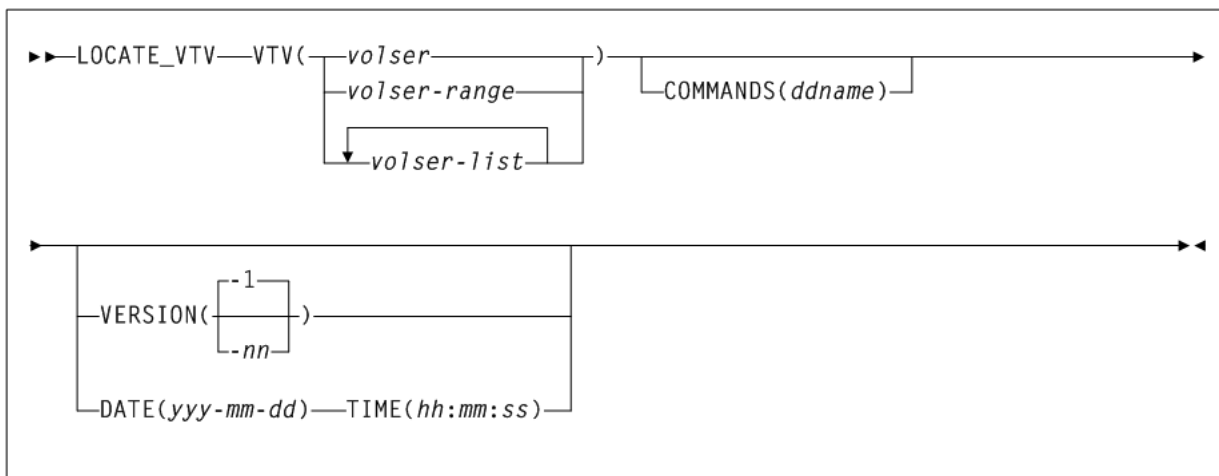
## LOGUTIL LOCATE\_VTV Statement

The LOGUTIL LOCATE\_VTV statement recovers a non-current version of a VTV.

### Syntax

Figure 3–100 shows LOGUTIL LOCATE\_VTV control statement syntax.

Figure 3–100 LOGUTIL LOCATE\_VTV statement syntax



### Parameters

As shown in Figure 3–100, the LOGUTIL LOCATE\_VTV control statement includes the following parameters:

#### VTV(volser, volser-range, or volser-list)

specifies specific VTV(s) to be recovered. You can specify a maximum of 1000 VTVs.

*volser*, *volser-range*, or *volser-list* indicates the volser, volser range, or volser list to be processed.

#### COMMANDS(ddname)

optionally, specifies the DD name of a data set that includes the recovery commands created when the COMMANDS parameter is specified on the LOGUTIL command.

*ddname* indicates the DD name.

The LOCATE\_VTV COMMANDS parameter works with the LOGUTIL COMMANDS parameter as follows:

- If you specify LOGUTIL COMMANDS, the LOCATE\_VTV statement outputs recovery commands to the specified data set even if you did not specify LOCATE\_VTV COMMANDS.

- If you specify `COMMANDS` on both `LOGUTIL` and `LOCATE_VTV`, recovery commands are only output to the `LOCATE_VTV COMMANDS` data set.

**VERSION(-nn)**

optionally, specifies the version of the VTV to recover.

`-nn` indicates the version. Valid values are -1 to -99. The default is `VERSION(-1)`, which is one version older than the current version.

---

**Note:** `VERSION` is mutually exclusive the `DATE` and `TIME` parameters.

---

**DATE(yyyy-mm-dd)**

optionally, specifies the date of the VTV version to recover. If you specify `DATE`, you must also specify `TIME`.

`yyyy-mm-dd` indicates the date, by year, month and day.

**TIME(hh:mm:ss)**

optionally, specifies the time of the VTV version to recover. If you specify `TIME` you must also specify `DATE`.

`hh:mm:ss` indicates the time, in hours minutes and seconds.

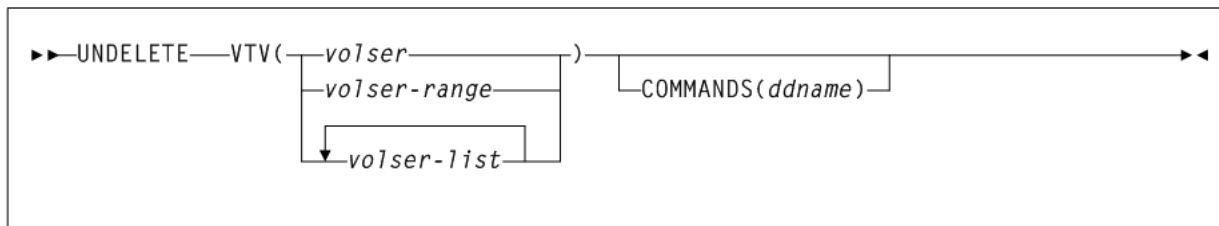
## LOGUTIL UNDELETE Statement

The `LOGUTIL UNDELETE` statement recovers deleted VTVs.

**Syntax**

Figure 3–101 shows `LOGUTIL UNDELETE` control statement syntax.

**Figure 3–101 LOGUTIL UNDELETE statement syntax**

**Parameters**

As shown in Figure 3–101, the `LOGUTIL UNDELETE` control statement includes the following parameters:

**VTV(volser, volser-range, or volser-list)**

specifies the VTV(s) to be recovered. You can specify a maximum of 1000 VTVs.

`volser`, `volser-range`, or `volser-list` indicates the volser, volser range, or volser list to be processed.

**COMMANDS(ddname)**

optionally, specifies the DD name of a data set that includes the recovery commands created when the `COMMANDS` parameter is specified on the `LOGUTIL` command.

`ddname` indicates the DD name.

The UNDELETE COMMANDS parameter works with the LOGUTIL COMMANDS parameter as follows:

- If you specify LOGUTIL COMMANDS, the UNDELETE statement outputs recovery commands to the specified data set even if you did not specify UNDELETE COMMANDS.
- If you specify COMMANDS on both LOGUTIL and UNDELETE, recovery commands are only output to the UNDELETE COMMANDS data set.

## MEDVERfy

### Interfaces:

- SLUADMIN utility only
- UUI Support: No

### Subsystem Requirements:

Active HSC/VTCS

## Description

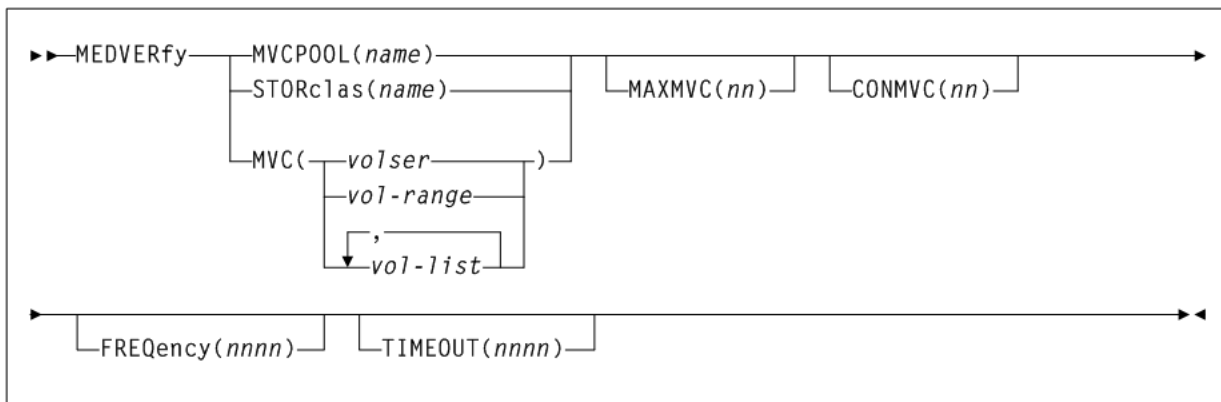
The MEDVERfy utility performs a Media Verification (MV) by verifying that VTV data can be read on MVCs or VMVCs (ELS 7.1 and VLE 1.2 and above only). For VLE, MEDVERfy ensures that deduplicated VMVCs can be “rehydrated” (reconstituted).

The utility reports on MVCs that pass or fail verification and also produces XML output.

## Syntax

Figure 3–102 shows MEDVERfy command syntax.

Figure 3–102 MEDVERfy command syntax



## Parameters

The following three, mutually exclusive, keyword parameters identify the MVCs to be verified:

### MVCPOOL(name)

optionally, specifies an MVCPOOL containing MVCs requiring data verification. An undefined MVCPOOL value aborts the utility with RC 8.

*name* indicates the MVC pool name.

**STORclas(name)**

optionally, specifies a storage class for MVCs requiring data verification. Based on time of last verify and/or any frequency specified on the command, the utility selects the best MVC candidates for verification processing.

An undefined `STORclas` value aborts the utility with RC 8.

*name* indicates the storage class name.

**MVC(volser, vol-range, or vol-list)**

optionally, specifies a list or range of MVCs to be verified. For individual MVCs whose `volser` falls outside those defined in the CDS, the utility aborts with RC 8.

*volser*, *vol-range*, or *vol-list* indicates a `volser`, `volser range`, or `volser list`.

---

**Note:** Based on time of last verify and/or any frequency specified on the command, the utility selects the best MVC candidates, within the specified selection criteria, for verification processing.

---

**MAXMVC(nn)**

optionally, specifies the maximum number of MVCs that will be processed by a single media verification task.

*nn* indicates the number of MVCs. Valid values are 1–99. If not specified, the default is 99.

**CONMVC(nn)**

optionally, specifies the maximum number of MVCs that VTCS concurrently processes for media verification.

*nn* indicates the number of MVCs. Valid values are 1–99. If not specified, the default is 1.

**FREQency(nnnn)**

optionally, establishes the verification scheduling interval, in days, that MVCs are to be verified. To be selected for verification, the number of days since the MVC's last verify date must be greater than the number of days specified.

*nnnn* indicates the interval, in days. Valid values are 1-9999. There is no default. If not specified, the MVC's last verify date, alone, is used for selecting MVCs to verify.

**TIMEOUT(nnnn)**

optionally, specifies the allowable time, in minutes, for the media verification utility to run.

*nnnn* indicates the allowable time, in minutes. Valid values are 1-9999. There is no default. If `TIMEOUT` is not specified, then there is no limit on utility run time.

---

**Note:** Similar to `RECLAIM` processing, if any MVCs are being processed when the `TIMEOUT` value is reached, `MEDVERfy` completes processing on those MVCs, and then stops.

---

## MEDVERfy Reports

The following figure shows an example of a Media Verify report that indicates successful verification for MVC DMV100:

**Example 3–51 MEDVERfy report (1 of 2)**

```
Media Verify - -----Media Verify request 6-----  
Media Verify - 1 MVC(s) selected for processing  
MVC DMV100 - verifying media  
MVC DMV100 - verify complete  
No exceptions to report
```

```
SLS0155I Condition code for utility function is 0
```

The report also lists MVCs not selected for MV processing under certain conditions. For example, in the following report, MVC DMV800 is skipped because it was mounted during verification selection:

**Example 3–52 MEDVERfy report (2 of 2)**

```
MEDVER MVC(DMV100,DMV800) CONMVC(1) TIMEOUT(180)
```

```
MVC DMV800 skipped, is mounted
```

```
Media Verify - -----Media Verify request 21-----  
Media Verify - 1 MVC(s) selected for processing  
MVC DMV100 - verifying media  
MVC DMV100 - verify complete  
No exceptions to report
```

```
No exceptions to report  
SLS0155I Condition code for utility function is 0
```

## MERGEcds

**Interfaces:**

- SLUADMIN utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC at BASE service level only

## Description

Use the MERGEcds command to do the following:

- Add or Change a configuration to reflect new ACSs or LSMs.
- Consolidate multiple CDSs into one CDS.
- Divide one CDS into multiple CDSs. MERGEcds copies volume information from each old CDS into the new CDS.
- Merge multiple ACSs into one ACS.
- Divide one ACS into multiple ACSs.
- Modify an ACSid and/or LSMid in order to renumber ACSs and LSMs.



- Modify VTSS names.
- Remove virtual or vault data from the CDS.
- Change the number of slots reserved for vaulted volumes.

---

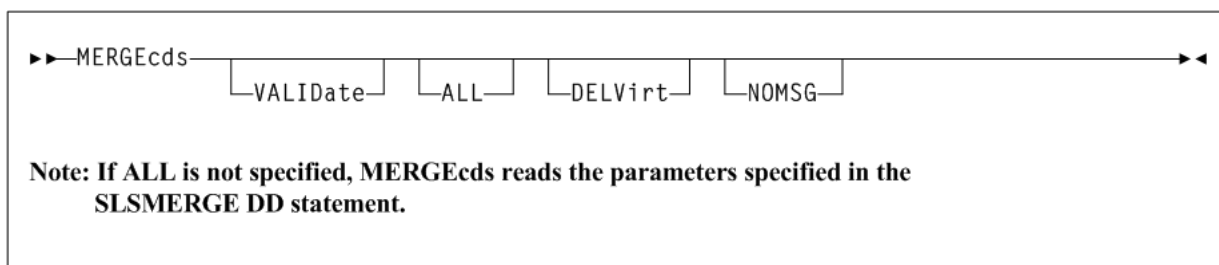
**Note:** Before using MERGEcds, refer to Oracle's ELS publication *Configuring HSC and VTCS* for detailed procedures used to run the utility.

---

## Syntax

Figure 3–103 shows MERGEcds command syntax.

**Figure 3–103 MERGEcds command syntax**



## Parameters

As shown in Figure 3–103, the MERGEcds command includes the following parameters:

### VALIDate

optionally, specifies to only validate that the configurations to be reconfigured or merged are compatible, but not perform the operation. MERGEcds VALIDate reports any duplicate, in-transit, and errant volumes.

### ALL

optionally, specifies to copy volume information for all ACSs and VTSSs from the "from" CDS to the "to" CDS. You can also specify the ALL parameter to convert a CDS to extended format.

- For a CDS merge, the ACS ID and LSM IDs, and VTSS names must match.
- If you are using VTCS, MVCs that are in the source (input) CDS but are not configured in the target (output) CDS, must be empty. An MVC is considered empty if it contains no current VTVs and, if previously used, has been drained.

---

**Note:** ALL and SLSMERGE DD are mutually exclusive. If you do not specify ALL, MERGEcds reads the parameters specified on the SLSMERGE control statement. These parameters specify the ACSs, LSMs, VTSSs, and Vaults whose volume information you want to merge or reconfigure. See "SLSMERGE Control Statement" on page 3-216 for more information.

---

### DELVirt

optionally, 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. Refer to Oracle's ELS publication *Managing HSC and VTCS* for instructions 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.

`DELvirt` has no effect unless `ALLVIRT` is specified on the `SLSMERGE` control statement.

**NOMSG**

optionally, suppresses message `SLS4245I`, which displays the volser of an MVC or VTV that was not copied to the "to" CDS.

`NOMSG` has no effect unless `DELvirt` is specified on the `MERGECDs` control statement and `ALLVIRT` is specified on the `SLSMERGE` control statement.

## Additional JCL Requirements

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statements apply to the `MERGEcds` JCL:

**SLSFCNTL**

specifies the current primary copy of the "from" HSC CDS.

**SLSFCTL2**

specifies the secondary copy of the CDS, if one is defined.

**SLSFSTBY**

optionally, specifies the standby copy of the CDS, if one is defined.

---

---

**Note:** If more than one of these statements are present, HSC determines the primary CDS from the Database Heartbeat (DHB) record of the "from" CDS. The "from" CDS is not modified. If a single CDS (`SLSFCNTL`) is specified, ensure that it is the primary CDS. No DHB validation occurs.

---

---

## SLSMERGE Control Statement

The `SLSMERGE` control statement specifies the "from" and "to" ACSs, LSMs, or Vaults to use for a merge, and enables you to rename a VTSS in the **Resident VTSS** field.

**Note:**

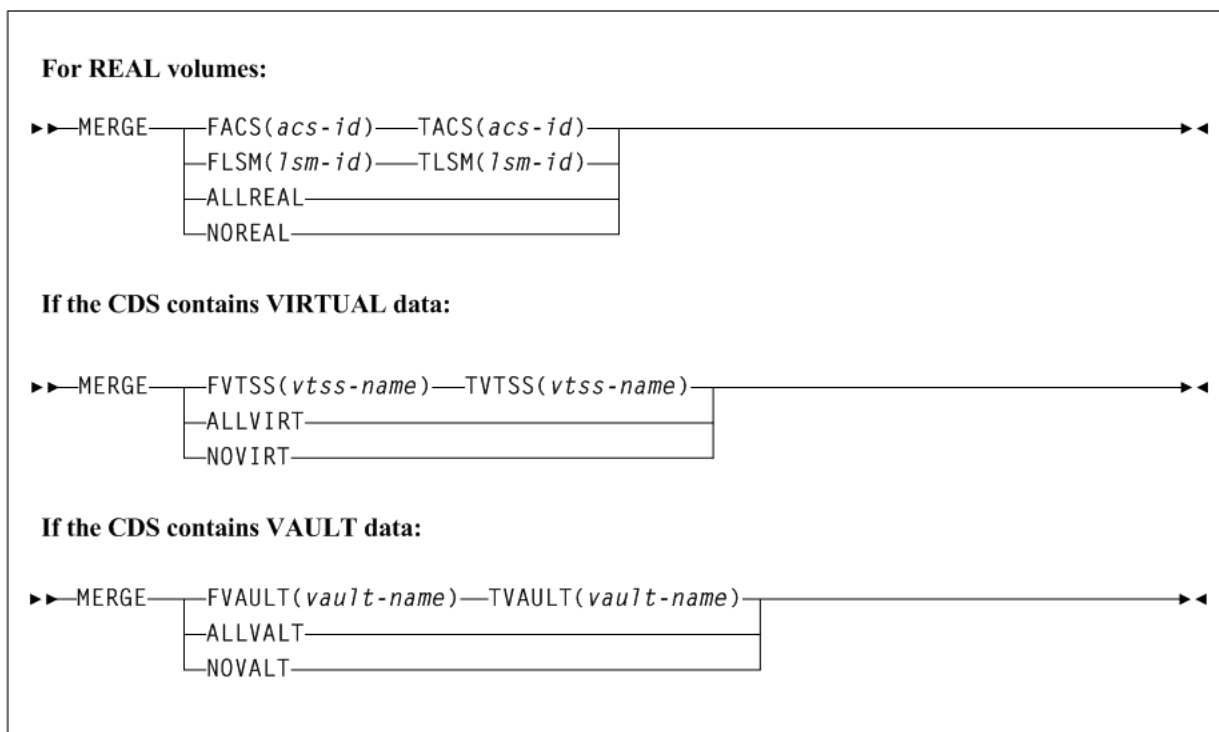
- The SLSMERGE control statement is optional and is mutually exclusive with the MERGEcds ALL parameter.
- If you use SLSMERGE control statements, you must include MERGE statements that account for all data in the “from” CDS.

For example, if your “from” CDS contains real, virtual, and vault data, you must supply a control card for each of these types to specify whether they are to be copied to the “to” CDS. If you do not want a particular type of data to be copied to the “to” CDS, issue the appropriate NO parameter (NOREAL, NOVIRT, or NOVALT) to exclude it.

**Syntax**

Figure 3–104 shows SLSMERGE control statement syntax.

**Figure 3–104 SLSMERGE control statement syntax**

**Parameters**

As shown in Figure 3–104, the SLSMERGE control statement includes the following parameters:

---

**Caution:** The “real” parameters (FACS, TACS, FLSM, TLSM) and “vault” parameters (FVAULT, TVault) act as selection criteria, while the “virtual” parameters (FVTSS, TVTSS) act only as rename criteria.

---

**Parameters (Real Volumes)**

For REAL volumes, the following parameters apply:

**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.

**ALLREAL**

Merge all real data only.

**NOREAL**

Do not merge real data.

**Parameters (CDS Containing Virtual Data)**

If the CDS contains VIRTUAL data, the following parameters apply:

**FVTSS(vtss-name)**

specifies the “from” VTSS name.

**TVTSS(vtss-name)**

specifies the “to” VTSS name.

You cannot use MERGEcds to remove data for certain VTSSs from the CDS.

FVTSS and TVTSS act only as rename criteria. When these parameters are specified, ALLVIRT is implied.

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. All VTSSs not specified in these statements are automatically merged to the same-named VTSS in the new CDS.

**ALLVIRT**

Merge all virtual data only.

**NOVIRT**

Do not merge virtual data.

**Parameters (CDS Containing VAULT Data)**

If the CDS contains VAULT data, the following parameters apply:

**FVAULT(vault-name)**

specifies the “from” vault.

**TVAULT(vault-name)**

specifies the “to” vault.

**ALLVALT**

Merge all vault data only.

**NOVALT**

Do not merge vault data.

## MERGMFST

**Interfaces:**

- Utility only
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC not required

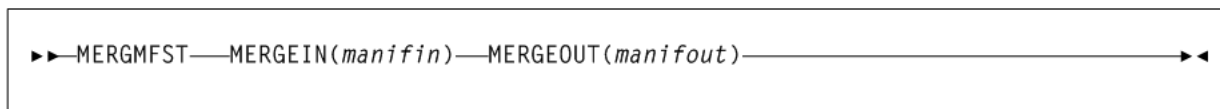
### Description

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

### Syntax

Figure 3–105 shows MERGMFST command syntax.

**Figure 3–105 MERGMFST command syntax**



### Parameters

As shown in Figure 3–105, the MERGMFST command includes the following parameters:

**MERGEIN(*manifin*)**

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

*manifin* indicates the DD name.

**MERGEOUT(*manifout*)**

specifies the DD statement for the merged manifest file.

*manifout* indicates the DD name.

### Additional JCL Requirements

In addition to the required JCL definition statements described in "Invoking SLUADMIN" on page 5-1, the following definition statements apply to the MERGMFST JCL:

**manifin DD**

DD statement for the input manifest file(s).

**manifout DD**

DD statement for the merged manifest file.

## METAdata

### Interfaces:

- Utility only
- UUI Support: Yes

### Subsystem Requirements:

Active HSC/VTCS

## Description

The `METAdata` command displays XML tags associated with a function that produces XML output.

---



---

**Note:** No text output is produced by the `METAdata` command. If neither XML nor CSV output is requested, the command will not produce any output.

---

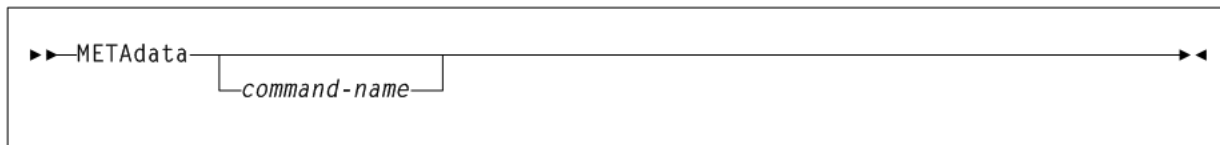


---

## Syntax

Figure 3–106 shows `METAdata` command syntax.

**Figure 3–106** *METAdata command syntax*



## Parameters

As shown in Figure 3–106, the `METAdata` command includes the following parameters:

### **command-name**

The command for which metadata is to be produced.

When the command contains "two parts" such as `Display Volume`, both portions should be entered. The `METAdata` command uses the same abbreviations for the `command-name` as the command itself, for example:

```
METAdata D V
```

## Tags Produced

The `METAdata` command itself does not support metadata.

The following tags are produced by the `METAdata` command:

### **<command\_name>**

The full name of the command.

### **<security\_level>**

Required security level if command authorization is in effect. Values are `QUERY`, `SET`, and `ADMIN`.

**<command\_tags>**

Header tag for all XML tags.

The following tags are produced by the `METAdata` command:

**<tag\_data>**

Header tag for each XML tag.

**<tag\_name>**

The XML tag name, for example, `volser`.

**<tag\_type>**

Values are header (header XML tag), data (XML data tag, normally associated with a value), and trailer (indicates the position of the trailer tag associated with a header tag).

**<occurrences>**

The maximum number of expected occurrences, for tags that may occur multiple times. This may be either a numeric value or "unlimited."

The following tags are produced by the `METAdata` command:

**<data\_type>**

Indicates the type of data expected in the tag value. Produced only for data tags. Values include: char, numeric, flag, hex, date, time.

**<maximum\_size>**

Indicates the maximum output data size.

## MGMTDEF

**Interfaces:**

- Console or `PARMLIB` only
- UUI Support: No

**Subsystem Requirements:**

Active HSC at `BASE` or `FULL` service level

## Description

The `MGMTDEF` command loads the following control statements from a specified definition data set:

- `MGMTclas`
- `MIGRSEL`
- `MIGRVTV`
- `MVCATTR`
- `STORclas`
- `STORLST`
- `STORSEL`
- `VTSSLST`
- `VTSSSEL`

---

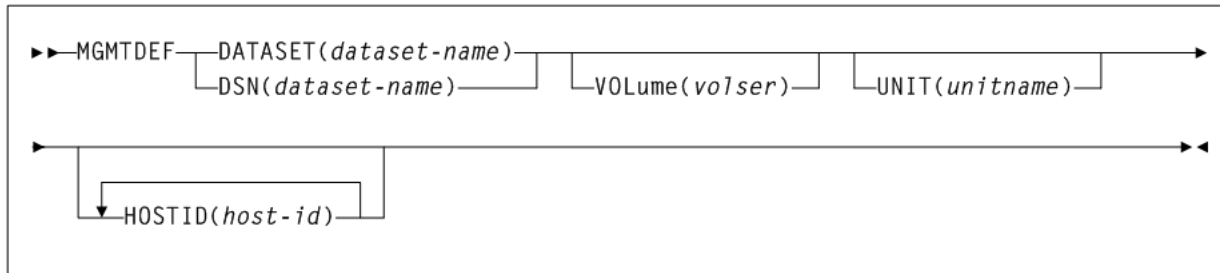
**Note:** When HSC/VTCS is active and the MGMTDEF command is issued to re-load MGMTclas control statements, the 'changed' control statements only affect newly created VTVs. Additional action is required to apply the changes to existing VTVs.

---

## Syntax

Figure 3–107 shows MGMTDEF command syntax.

**Figure 3–107 MGMTDEF command syntax**



## Parameters

As shown in Figure 3–107, the MGMTDEF command includes the following parameters:

### DATASET(dataset-name) or DSN(dataset-name)

specifies the definition data set that contains the MGMTclas and STORclas statements to load.

*dataset-name* indicates the data set name.

### VOLume(volser)

optionally, 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* indicates the DASD volser.

### UNIT(unitname)

optionally, specifies the DASD device where the definition data set resides.

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

### HOSTID(host-id)

optionally, 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* indicates the name of one or more hosts from which to execute the MGMTDEF command. Multiple hosts must be separated by commas.

## MGMTclas Control Statement

The MGMTclas control statement defines a VSM Management Class. It is loaded by the MGMTDEF command.



---

---

**Note:**

- By default, advanced VSM management features are automatically enabled. The `FEATures` command, included in previous ELS releases to enable these features, is no longer valid.
- ELS includes parameters designed to improve migration control. If you choose not to use these features, refer to Oracle's ELS publication *Legacy Interfaces Reference* for information about existing legacy `MGMTclas` parameters.
- When HSC/VTCS is active and the `MGMTDEF` command is issued to re-load `MGMTclas` control statements, the 'changed' control statements only affect newly created VTVs. Additional action is required to apply the changes to existing VTVs.

For example, If you change the `IMMDELAY` value, you must recycle the HSC to apply the change to existing VTVs. If you change the `MIGpol` value from one storage class to two, you must recall the VTV to allow for creation of the second migrated copy.

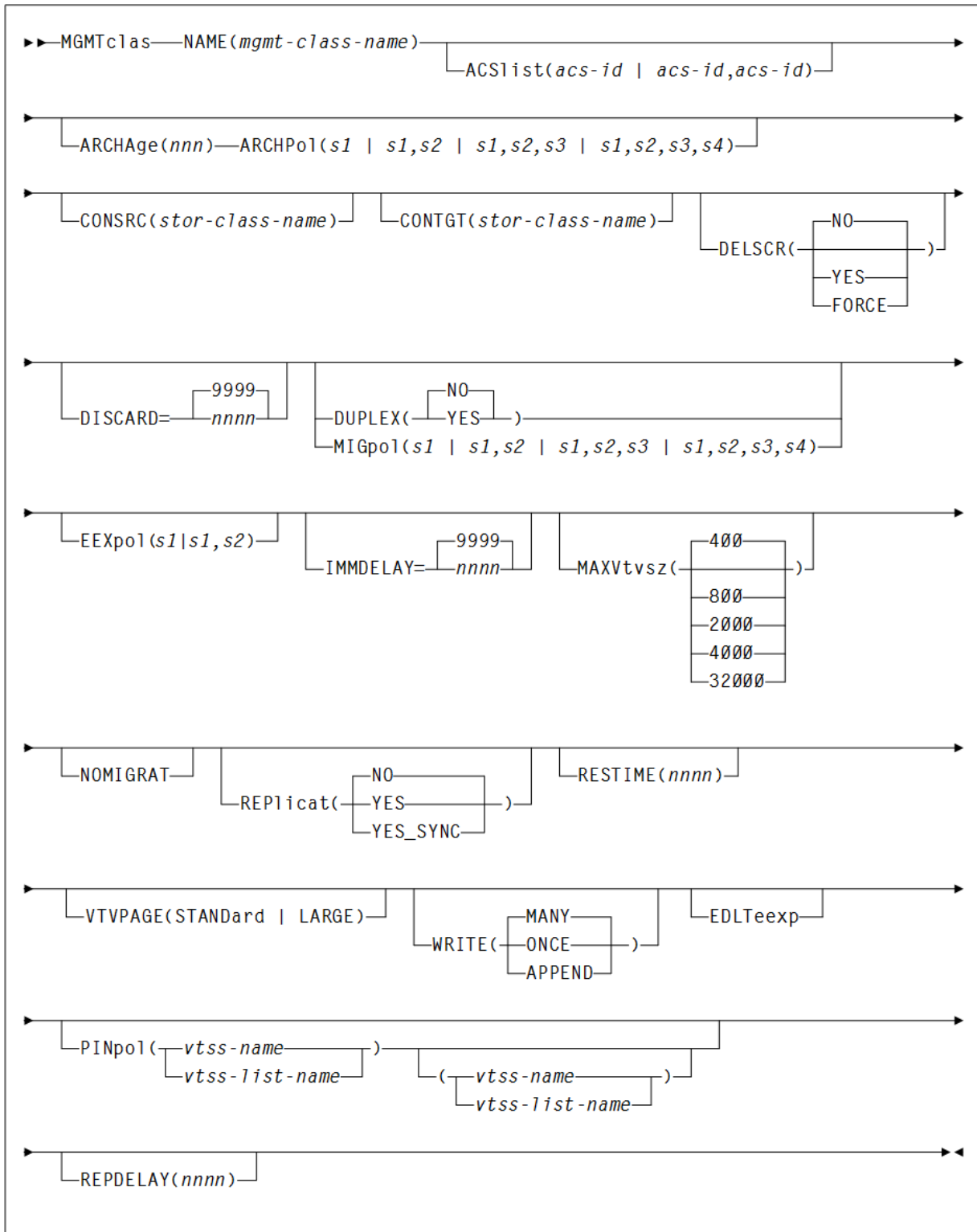
---

---

**Syntax**

Figure 3–108 shows `MGMTclas` control statement syntax.

**Figure 3–108 MGMTclas control statement syntax**



**Parameters**

As shown in [Figure 3–108](#), the MGMTclas command includes the following parameters:

**NAME(mgmt-class-name)**

specifies the name of the Management Class.

*mgmt-class-name* indicates the Management Class name. This name must be one to eight alphanumeric characters beginning with an alpha character and must follow SMS naming conventions.

**ACSlist(acs-id or acs-id,acs-id)**

optionally, specifies the ACSs from which RTDs and MVCs are selected. If not specified, the default is the ACS specified on the CONFIG DEFULTACS parameter.

See DUPlex, below, for information about using the DUPlex and ACSlist parameters.

*acs-id* or *acs-id, acs-id* indicates either one or two ACS IDs. An ACSid is a two digit decimal ACS identifier from 00-99.

**ARCHAge(nnn)**

optionally, 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* indicates the VTV age in days.

**ARCHPol(stor-class-name or stor-class-list)**

optionally, specifies up to four Storage Classes that specify the ACS and media type of the archive MVCs. If you specify:

- If you specify one Storage Class, VTCS archives one copy of a VTV.
- If you specify 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.
- If you specify 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, MVC space reclamation, and VTV consolidation function.

---

This parameter is optional; there is no default value. If you specify ARCHPol, you must specify ARCHAge.

*stor-clas-name1...stor-clas-namen* indicates 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.

**CONSRC(stor-class-name)**

optionally, 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 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 uses 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 uses the MVC associated with the previous Storage Class (in order specified in the MIGpol parameter).

*stor-class-name* indicates the name of a Storage Class that you defined on the `STORclas` control statement.

#### **CONTGT(stor-class-name)**

optionally, specifies the Storage Class that determines the output MVC ACS and media for VTV consolidation (executing `CONSOLID`, `EXPORT VTV` or `EXPORT MGMTclas`). 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* indicates the name of a Storage Class that you defined on the `STORclas` control statement.

#### **DELSCR**

optionally, specifies whether VSM deletes scratched VTVs.

##### **NO**

Do not delete scratched VTVs (the default).

If you are using the pinning feature, enabled in the `MGMGTclas` control statement, resident replica copies are deleted when the VTV is scratched. The primary copy is not deleted until migration time. Once a VTV is scratched, it is no longer pinned.

##### **YES**

Delete scratched VTVs unless this would cause a processing delay due to unavailable resources.

##### **FORCE**

Delete scratched VTVs and wait for access to any resources that are required. This may cause a processing delay.

---



---

**Caution:** When you scratch a VTV with `DELSCR YES` attribute, VSM erases the VTV data at scratch synchronization time, which eliminates the ability to “unscratch” a VTV to recover data!

---



---

Also note that when using 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, Oracle strongly recommends that you do not run any jobs that use scratches during HSC scratch synchronization.

See "[Scratch Conversion Utility \(SLUCONDB\)](#)" on page 5-13 for more information about HSC scratch synchronization with the Scratch Conversion Utility (`SLUCONDB`).

Refer to the *LCM User's Guide* for more information about LCM scratch synchronization with the `SYNCVTV` function.

**DISCARD(nnnn)**

optionally, specifies the discard time in hours. This value represents the time after a VTV is dismounted that the VTV is kept in the buffer. After this time value expires, the VTV is preferred for deletion from the VTSS buffer if all required copies of the VTV exist on MVCs.

*nnnn* indicates the time in hours. Valid values are 0 to 9999. The default is 9999.

When *IMMEDmig* is specified, *DISCARD* is not used for immediate migration processing. It is only applicable for *AUTO* migration requests. If *RESTIME* is also specified for a VTV, it overrides the *DISCARD* value.

**DUplex**

optionally, specifies whether VSM will migrate two copies of the VTV to two MVCs. *DUplex* and *MIGpol* are mutually exclusive.

**NO**

Do not duplex the VTV (the default).

**YES**

Duplex the VTV.

The following table describes possible scenarios using the *DUplex* and *ACSlist* parameters:

**Table 3–7 MGMTclas ACSlist/DUplex Scenarios**

<b>DUplex Setting</b>	<b>ACSlist Setting</b>	<b>Action</b>
YES	two ACSs	VSM migrates the VTVs to two MVCs, one in each ACS. (This scenario is the normal one for duplexing to two ACSs.)
YES	one ACS	VSM migrates the VTVs to two MVCs in the ACS specified.
NO	two ACSs	VSM ignores the <i>DUplex</i> policy and migrates the VTVs to two MVCs, one in each ACS.
NO	one ACS	VSM migrates the VTVs to one MVC in the ACS specified.

**MIGpol(s1 or s1,s2 or s1,s2,s3 or s1,s2,s3,s4)**

optionally, specifies up to four Storage Classes that specify the ACS and media type of migration MVCs. *DUplex* and *MIGpol* are mutually exclusive.

- If you specify one Storage Class, VTCS migrates one copy of a VTV.
- If you specify multiple Storage Classes (with different *ACS* values, different *MEDIA* values, or both), VTCS makes multiple copies the VTV to different MVCs in different ACSs.
- If you specify 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, MVC space reclamation, and VTV consolidation function.

---

This parameter is optional; there is no default value.

*s1* or *s1, s2* or *s1, s2, s3* or *s1, s2, s3, s4* indicates 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 (V61ABOVE)` 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).

---



---

### **EEXpol(s1 or s1,s2)**

optionally, specifies the storage classes for electronic export.

*s1* or *s1, s2* indicates a maximum of two Storage Classes that specify the `TAPEPLEX` parameter. If these storage classes do not specify the `TAPEPLEX` parameter, an error condition results.

- If there are two `TAPELEX` storage classes, then they must specify different destination TapePlex names.
- A warning will be generated if the two TapePlex storage classes are specified with the `SYNC=YES` parameter. It is only possible to synchronously export to one other TapePlex.
- If there is a conflict, electronic export functionality takes precedence over replication with a cluster.
- If either one of the storage classes referenced contains the `THISPLEX` name, then the storage class is silently ignored. This enables common storage class definitions to be applied across TapePlexes.

### **IMMDELAY(nnnn)**

optionally, specifies the immediate migration delay time; the amount of time after VTV dismount that the migration should be queued for action.

This enables VTVs used in multi-step jobs to remain resident for a specified time before being processed for migration.

*nnnn* indicates the immediate migration delay time in minutes. Valid values are 0 to 9999 (the default).

If `IMMDELAY=9999` then immediate migration does not occur. Migration and deletion is handled through auto or command migration.

When this value is specified, `MIGRSEL` and `MIGRVTV` have no influence on migration control.

`RESTIME` and `DISCARD` parameter values represent buffer management priorities:

- If the `IMMDELAY` value is less than the `RESTIME` value, keep the VTD in the VTSS as a priority.
- If the `IMMDELAY` value is greater than the `RESTIME` value but less than the `DISCARD` value, manage the buffer according to LRU (default state).
- If the `IMMDELAY` value is greater than the `DISCARD` value, remove the VTD from the VTSS as a priority.

The `IMMDELAY` parameter is designed to replace the `IMMEDmig` parameter, described in Oracle's ELS publication *ELS Legacy Interfaces Reference*. These parameters are mutually exclusive. The following table describes equivalent values:

**Table 3–8 IMMDELAY and IMMEdmig Equivalent Values**

<b>IMMDELAY or DISCARD Values</b>	<b>Equivalent IMMEdmig Value</b>	<b>Action</b>
IMMDELAY (1-9998)	None	Delay migration for the specified number of minutes.
IMMDELAY (9999)	IMMED (NONE)	VSM does not immediately migrate the VTV, but migrates it according to standard VSM migration criteria. MIGRSEL and MIGRVTV have no influence on migration control.
IMMDELAY (0) DISCARD (9999)	IMMED (KEEP)	VSM immediately migrates a VTV and keeps a copy resident on the VTSS until the VTV becomes eligible for deletion.
IMMDELAY (0) DISCARD (0)	IMMED (DELETE)	VSM immediately migrates the VTV and then deletes it from the VTSS.

**Note:** If you specify `REPLICAT YES` or `YES_SYNC`, VTCS defaults to `IMMDELAY 0`. If you specify `REPLICAT YES` or `YES_SYNC` in combination with `IMMDELAY 9999`, VTCS ignores the `IMMDELAY` parameter without issuing a message. For example:

```
MGMT NAME (REPIMNO) REP (YES) IMMDELAY (9999)
```

causes VTCS to use `IMMDELAY 0` and perform immediate migration. To suppress immediate migration, add a `MIGRVTV MGMTclas(x) IMMDELAY (9999)` statement for the management class. See "[MIGRVTV Control Statement](#)" on page 3-235 for more information.

### **MAXVtvsz**

optionally, specifies the maximum size for VTVs in this Management Class. Valid values for this parameter depend on both the CDS level and the microcode levels of the applicable VTSSs.

#### **400**

400 MB. This is the default.

#### **800**

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

#### **2000**

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

#### **4000**

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

#### **32000**

32 GB. The CDS must be at an I level or above.

Considerations:

- 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 or VSM5 microcode D02.02.00.00 or VSM3 microcode N01.00.77.00. No installed option is required.
- `MAXVTVSZ(32000)` requires VSM6 minimum microcode level 6.2 and VLE minimum microcode level 1.5.1, if VLE is in the configuration.

**NOMIGRAT**

optionally, specifies that VTVs in the Management Class are not candidates for migration, consolidation or export, but are candidates to reside on a tapeless VTSS.

VTSS selection is changed to prefer tapeless VTSSs for VTVs in Management Classes with `NOMIGRAT`, and to disallow VTVs without `NOMIGRAT` from VTSSs with no RTDs.

`NOMIGRAT` is mutually exclusive with `ACSLIST`, `IMMDELAY`, `DUPLEX`, `MIGPOL`, `ARCHAGE`, `ARCHPOL`, `RESTIME`, `CONSRC` and `CONTGT`.

**REPLICAT**

optionally, specifies whether VSM replicates the VTV.

Synchronous replication must be enabled through the `CONFIG GLOBAL SYNCHREP` parameter. For more information, see "[CONFIG GLOBAL Statement](#)" on page 3-42.

If you specify `REPLICAT YES` or `YES_SYNC`, VTCS defaults to `IMMDELAY 0`. If you specify `REPLICAT YES` or `YES_SYNC` in combination with `IMMDELAY 9999`, VTCS ignores the `IMMDELAY` parameter without issuing a message. For example:

```
MGMT NAME(REPIMNO) REP(YES) IMMDELAY(9999)
```

causes VTCS to use `IMMDELAY 0` and perform immediate migration.

To suppress immediate migration, add a `MIGRVTV MGMTclas(x) IMMDELAY(9999)` statement for the management class. See "[MIGRVTV Control Statement](#)" on page 3-235 for more information.

**NO**

Do not replicate the VTV (the default).

**YES**

Asynchronously replicate the VTV.

**YES\_SYNC**

Synchronously replicate the VTV.

**RESTIME(nnnn)**

optionally, 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. Value 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.

*nnnn* indicates the residency time in hours.

`RESTIME` and `IMMEDmig(DELETE)` are mutually exclusive. `RESTIME` takes effect when a VTV is created, and does not apply to a recalled VTV.



**VTVPAGE**

optionally, specifies that the page size used to store VTV data in the VTSS and on the MVCs. This setting only applies to 400 MB and 800 MB VTVs. If `VTVPAGE` is not specified on either the `MGMTCLAS` statement or the `CONFIG GLOBAL` statement, the default is `STANDARD`.

**STANDARD**

standard page size, which is compatible with all VSM3 or 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`" on page 3-37. For 2 GB, 4 GB, or 32 GB VTVs (`MAXVTVSZ` 2000, 4000, or 32000), a `VTVPAGE` setting of `LARGE` is always used.

## Considerations:

- `VTVPAGE` does not apply to VSM2s. `VTVPAGE (LARGE)` requires VSM4 or 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`.
- 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 `VTVPAGE` defaults to `STANDARD`.
- If you specify `STANDARD` for 2 GB 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.
- The `VTVPAGE` valued specified for this Management Class overrides the global value specified on the `CONFIG` utility.

**WRITE**

optionally, specifies the VTSS-resident VTV `VOLSAFE` policy as follows:

**MANY**

specifies no `VOLSAFE` write protection. This is the default.

**ONCE**

specifies partial (write once) `VOLSAFE` protection. After the VTV is non-scratch, it cannot be overwritten or appended.

**APPEND**

specifies full `VOLSAFE` protection. This is only supported for VSM6 systems.

- VTV data can be appended once non-scratch.
- Data cannot be overwritten.

The `Display VTV` command and `VTVRPT` report will indicate that the VTV is write-append protected.

VTVs with full VOLS SAFE protection can only be scratched with RACF ALTER authority. Use the following RACF commands to set RACF authority:

```
RDEFINE TAPEVOL volser UACC(NONE)
PERMIT volser CLASS(TAPEVOL) ID(userid) ACCESS(ALTER)
```

### **EDLTeexp**

optionally, specifies whether pending Electronic Export VTVs are candidates for early deletion. The VTVs will only be deleted if all CLINKs to the remote Tapeplex are not operational. The EDLTeexp option requires that the EEXPOL and MIGPOL policies be set for the management class.

---

---

**Note:** For the early delete VTVs to be electronically exported, you must run the VTCS RECONCIL utility after the remote Tapeplex CLINKs are varied ONLINE.

---

---

### **PINpol(vtss-name or vtss-list-name)**

optionally, specifies one or two VTV pinning locations where a resident replica of a VTV is stored in the VTSS buffer. A location can be either a *vtss-name* or *vtsslst name*.

*vtssname* indicates a VTSS name.

*vtss-list-name* indicates the name of a VTSS list (as specified in a MGMTDEF VTSSLST control statement). See "VTSSLST Control Statement" on page 3-246 for more information. This value can be a maximum of 8 alphanumeric characters, with the first character in the range of A-Z.

If two locations are specified, there must also be a replication (REPLICAT) policy defined on the MGMTCLAS statement.

---

---

**Note:** Only pin VTVs that are required to stay resident. Pinning every VTV or too many VTVs can cause the VTSS to fill to capacity and become unusable.

For pinning to function correctly, PTF L1H18UI (adds VTV pinning support) must be applied to all systems, and the MGMTCLAS statements with a pinning policy must be defined on all hosts.

---

---

### **REPDELAY(nnnn)**

optionally, specifies the asynchronous replication and/or asynchronous electronic export delay time, the minimum amount of time after a VTV is dismounted that the asynchronous replication and/or asynchronous electronic export will be delayed.

*nnnn* indicates the delay time in minutes. Valid values are 0 (the default) to 9999.

- REPDELAY (*nnnn*) requires either REPLICAT or EEXPOL. If both REPLICAT (YES) and EEXPOL are specified on the same MGMTCLAS then the asynchronous replication and asynchronous electronic export will both be delayed by the time specified.
- REPDELAY is ignored for synchronous replication and synchronous electronic export.
- REPDELAY is honored across varying a VTSS offline or online. When the VTSS is varied back online and the delay time has not expired then the VTV(s) will continue to be delayed. If the delay time has expired then the VTV(s) will be replicated once the VTSS is online.

- REPDELAY is not honored across cycling HSC. Once HSC or VTCS is fully initialized the VTV(s) will be replicated.

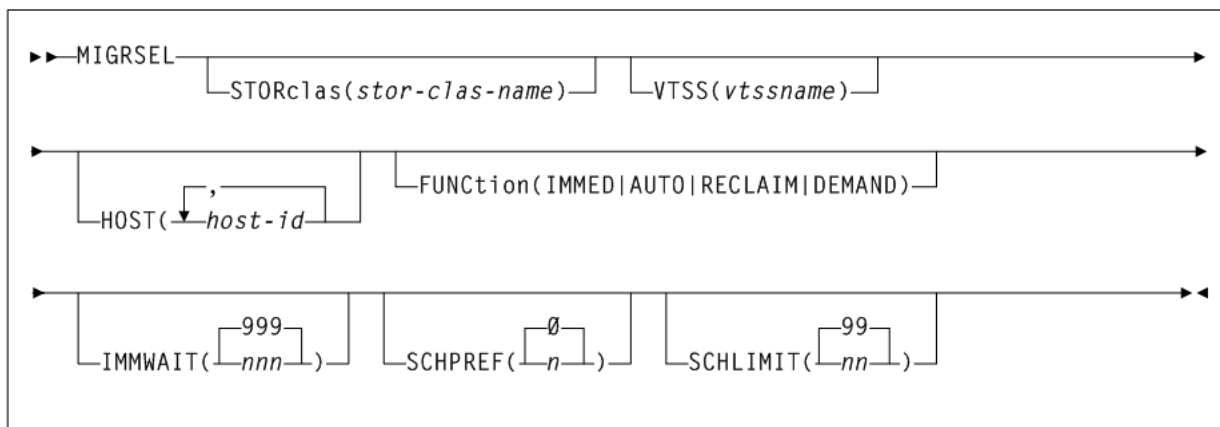
## MIGRSEL Control Statement

The MIGRSEL control statement controls migration request settings for a Storage Class, VTSS, and/or host. It is loaded by the MGMTDEF command.

### Syntax

Figure 3–109 shows MIGRSEL control statement syntax.

**Figure 3–109** MIGRSEL control statement syntax



### Parameters

As shown in Figure 3–109, the MIGRSEL control statement includes the following parameters:

#### STORclas(stor-clas-name)

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* indicates the Storage Class name you defined on a STORclas control statement.

#### VTSS(vtssname)

optionally, specifies a VTSS to which the migration settings apply. If you do not specify a VTSS, the statement applies to all VTSSs.

*vtssname* indicates the VTSS name.

#### HOST(host-id)

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* indicates 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).

**IMMWAIT(nnn)**

optionally, makes the MIGRSEL rule sensitive to the state of the current immediate migration work load. The specified value provides an immediate migration wait time or age, in minutes, that this MIGRSEL migration rule will apply to. This value is compared against the amount of time VTVs have been waiting for immediate migration to a particular storage class. If the amount of time (minutes) that VTVs have been waiting is less than or equal to the IMMWAIT value the MIGRSEL rule will apply.

*nnn* indicates the immediate migration wait time or age, in minutes. Valid values are 0 to 999. The default is 999.

- The default value of 999 makes the rule apply to all VTV wait times.
- A value of zero is used for applying MIGRSEL rules if immediate migration is not active.

**SCHPREF(n)**

optionally, preferences automatic and immediate migration per storage class. MIGRSEL VTSS and HOST allow you to specify the VTSS and host to which the preferencing applies.

*n* indicates the preferencing value. Valid values are 0 to 9. The default is 0.

- Higher values can produce quicker migration times, but may not optimize MVC usage.
- Lower values may produce slower migration times, but may optimize MVC usage.

The MIGRSEL SCHPREF setting may be affected by the number of RTDs available, the SCHLIMIT setting, and the GLOBAL MAXMIG parameter for the VTSS.

**SCHLIMIT(nn)**

optionally, de-prioritizes migration per Storage Class.

*nn* indicates 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-prioritize 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-prioritizes 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.

## MIGRVTV Control Statement

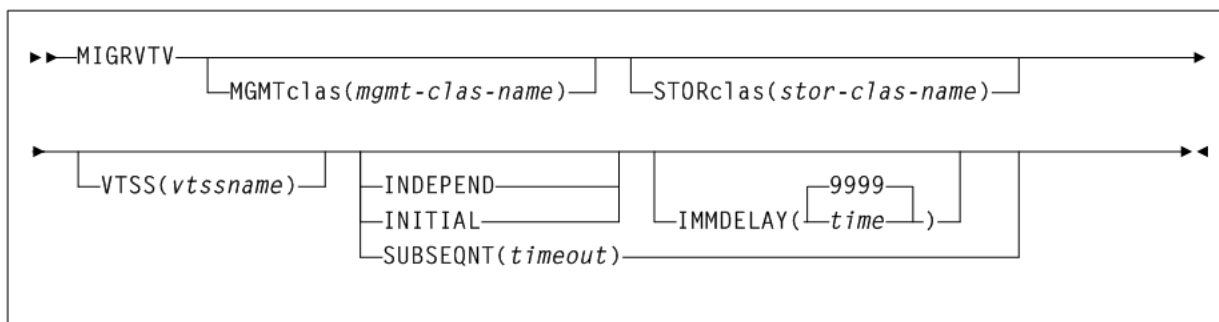
The `MIGRVTV` control statement controls individual VTV copies processed by immediate migration. It is loaded by the `MGMTDEF` command.

This statement defines how the behavior of an individual immediate migrate should be modified depending upon various environmental considerations. The `MIGRVTV` rules are searched for each immediate migrate instance of a VTV. The first match found for each instance is used to modify the immediate migrate.

### Syntax

Figure 3–110 shows `MIGRVTV` control statement syntax.

**Figure 3–110** *MIGRVTV control statement syntax*



### Parameters

As shown in Figure 3–110, the `MIGRVTV` control statement includes the following parameters:

#### **MGMTclas(mgmt-clas-name)**

optionally, specifies a Management Class to which the migration settings apply. If you do not specify a Management Class, the statement applies to all Management Classes.

*mgmt-clas-name* indicates the name of a Management Class you defined on the `MGMTclas` control statement.

#### **VTSS(vtssname)**

optionally, specifies a VTSS to which the migration settings apply. If you do not specify a VTSS, the statement applies to all VTSSs.

*vtssname* indicates the VTSS name.

#### **STORclas(stor-clas-name)**

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* indicates the name of a Storage Class you defined on the `STORclas` control statement.

**INDEPEND**

Affected VTVs are independent of any other VTV migrate instance. This is the default and applies to any migrates that do not match a `MIGRVTV` rule or any rule without an `INITIAL` or `SUBSEQNT` parameter.

**INITIAL**

Affected VTV migrates are scheduled and performed before any migrate instances that match to a rule with the `SUBSEQNT` parameter.

**SUBSEQNT(timeout)**

Affected VTV migrates are scheduled and performed after any migrate instances that match to a rule with the `INITIAL` parameter.

*timeout* indicates the time in minutes after which the rule expires. This is in addition to any `IMMDELAY` value applied to the `INITIAL` migrate instances.

**IMMDELAY(time)**

optionally, specifies the immediate migration delay time; the amount of time after VTV dismount that the migration should be queued for action. This enables VTVs used in multi-step jobs to remain resident for a specified time before being processed for migration.

This `IMMDELAY` parameter may be used to override the `IMMDELAY` value in the management class definition for the VTV as a whole. If the delay exceeds the `DISCARD` value in the management class, then the migrate will cause an implicit deletion of the VTV copy from the source VTSS.

If a delay time of 9999 is specified by this parameter or the `IMMDELAY` parameter in the management class, then the specific migrate instance is deleted and must later be serviced using auto/command migration.

*nnnn* indicates the immediate migration delay time in minutes. Valid values are 0 to 9999 (the default).

Considerations:

- If `IMMDELAY=9999` then immediate migration does not occur. Migration and deletion is handled via auto/command migration.  
When this value is specified, `MIGRSEL` and `MIGRVTV` have no influence on migration control.
- If `IMMDELAY=0` then immediate migration is scheduled immediately.
- If `IMMDELAY` is less than 9999 and `DISCARD` is greater than `IMMDELAY` then immediate migration occurs and deletion is deferred to auto/command migration.
- If `IMMDELAY` is less than 9999 and `DISCARD` is less than or equal to `IMMDELAY` then immediate migration and immediate deletion occur together.
- If auto migration encounters a VTV that has passed its `DISCARD` time, the VTV is moved to the front of the migration queue, and it will be processed first.

---

---

**Note:** See "[MGMTclas Control Statement](#)" on page 3-222 for information about the `DISCARD` parameter.

---

---

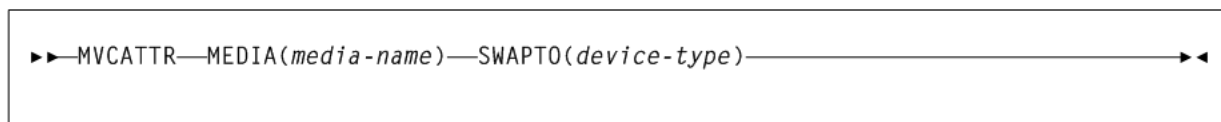
## MVCATTR Control Statement

The MVCATTR control statement assigns a swap-to RTD device type to an MVC media name. When an error occurs while reading an MVC on an RTD, VTCS may swap the MVC to another RTD to retry the operation.

### Syntax

Figure 3–111 shows MVCATTR control statement syntax.

**Figure 3–111 MVCATTR control statement syntax**



### Parameters

As shown in Figure 3–111, the MVCATTR control statement includes the following parameters:

#### MEDIA(media-name)

specifies the MVC media name to which the attribute is assigned. Only one MVCATTR should be coded for each *media-name*.

*media-name* indicates the MVC media name. See Table 3–9.

#### SWAPTO(device-type)

defines the RTD device type the MVC is swapped to (if possible).

*device-type* indicates the RTD device type. See Table 3–9.

**Table 3–9 Valid MVC Media Names and Compatible SWAPTO RTD Device Types**

Valid MEDIA Names	Compatible SWAPTO Device Types
STK1R	STK1RA34, STK1RB34, STK1RD34, STK1RDE4
STK1RC	STK1RC34, STK1RD34, STK1RDE4
STK1RD	STK1RD34, STK1RDE4
STK1RDE	STK1RDE4
STK2P	STK2PA34, STK2PB34
STK2PB	STK2PB34
T1000T1	T1A34, T1AE34, T1B34, T1BE34
T1000TS	T1A34, T1AE34, T1B34, T1BE34
T1000E1	T1AE34, T1BE34
T1000ES	T1AE34, T1BE34
T1B00T1	T1B34, T1BE34
T1B00TS	T1B34, T1BE34
T1B00E1	T1BE34
T1B00ES	T1BE34

## STORclas Control Statement

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

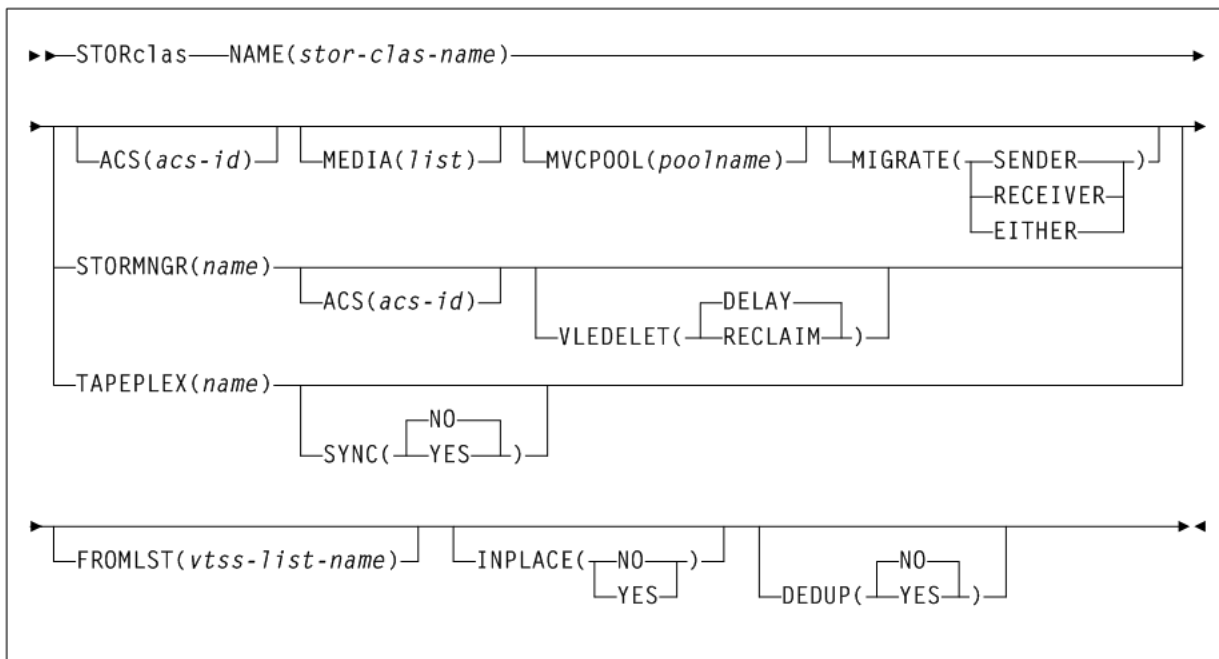
This statement can specify whether a VTV copy is to be written to:

- An MVC (with required attributes)
- The name of a remote TapePlex to which a copy of the VTV is to be exported. It also specifies whether a storage class uses partitioned media.
- The subsystem name of a VLE.

### Syntax

Figure 3–112 shows `STORclas` control statement syntax.

**Figure 3–112** *STORclas control statement syntax*



### Parameters

As shown in Figure 3–112, the `STORclas` control statement includes the following parameters:

#### **NAME(stor-clas-name)**

specifies the name of the Storage Class.

*stor-clas-name* indicates 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(acs-id)**

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

*acs-id* indicates a two digit decimal ACS identifier from 00-99.



**STORMNGR(name)**

optionally, specifies one of the following:

- VLE subsystem name
- remote library

*name* indicates the Storage Manager name:

- For VLEs, this value must match the VLE Subsystem Name.
- For remote libraries, this value must match a TapePlex name defined on the SMC TAPEPLEX NAME parameter.

For more information, refer to Oracle's ELS publication *Configuring the Host Software for VLE*.

You can specify both STORMNGR and ACS parameters to restrict selection of MVCs to the library in the remote TapePlex with the designated ACS number.

**VLEDELET**

optionally, control when deletion of the VTV from the VLE occurs.

**DELAY**

The deletion of the VTV from the VLE will be delayed until the vMVC is mounted for migration. DELAY is the default.

**RECLAIM**

The deletion of the VTV from the VLE will be done when the RECLAIM utility is run against the vMVC.

---

**Caution:** The current CDS backup can no longer be used as a DR baseline for vMVC content if you use the DRCHKPT utility and/or the CONFIG RECLAIM PROTECT parameter to protect CDS backup content for vMVCs, and then RECLAIM a vMVC for a STORCLAS that specifies VLEDELET (RECLAIM).

---

**MEDIA(list)**

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

*list* indicates the preference list of media types.

**MVCPOOL(poolname)**

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* indicates 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 FROMLIST 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(name)**

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.

*name* indicates the TapePlex name.

**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(vtss-list-name)**

optionally, when migrating or exporting to a Storage Class, provides VTCS with a list of VTSSs to source the VTV. FROMLST applies to both Clustered VTSSs and to VLE to VLE connections. For more information, see "[FROMLST Parameter Usage](#)" on page 3-242.

*vtss-list-name* indicates the name of a VTSS list (as specified in a MGMTDEF VTSSLST control statement). See "[VTSSLST Control Statement](#)" on page 3-246 for more information.

**INPLACE**

optionally, specifies whether Dynamic Reclaim is enabled for this Storage Class. See "[Dynamic Reclaim Usage](#)" on page 3-241 for more information.

**NO**

Dynamic Reclaim is not enabled for this Storage Class.

**YES**

Dynamic Reclaim is enabled for this Storage Class.

**DEDUP**

optionally, specifies whether VTV data migrated to VMVCs in a VLE is deduplicated.

**NO**

Do not deduplicate VTVs. This is the default.

**YES**

Deduplicate VTVs when migrated to VMVCs.

The `DEDUP` value is a Storage Class attribute that applies to all VMVC mount requests sent to a VLE. `DEDUP` applies to all migrations that occur for the life of the mount. To change the `DEDUP` value, the VMVC is dismounted, the `DEDUP` value is changed, and the VMVC is remounted.

**Dynamic Reclaim Usage**

Dynamic Reclaim only applies to T10000B and above media (full and full encrypted only) that have been formatted by T10000 drives capable of creating partitions. It requires an H level CDS, VSM5 microcode D02.11.16.00 or later, and T1010000B firmware 1.41a.209 or later.

You can specify Dynamic Reclaim policies at the global, system wide level or on a Storage Class and MVC pool level. Use the `VTCS CONFIG RECLAIM` statement to set the global policies:

- The `INPLACE` parameter controls the usage of Dynamic Reclaim. Its setting can either be explicitly specified on the `STORCLAS` statement or inherited from the `RECLAIM` statement in the VTCS configuration. The default is `INPLACE=NO`. The setting that is derived for a Storage Class does not force usage of media formatted in either mode. It is acceptable to use this parameter with drives or media that do not support partitioning. Also, any media written in a different mode continues to be a migration target. This helps avoid the need to actively manage the transition when Dynamic Reclaim is enabled. Its effect is to cause new media allocated to the Storage Class to be formatted and written in either standard (`INPLACE=NO`) or partitioned (`INPLACE=YES`) mode. The switch can only occur when media is empty.
- If a hard switch-over between standard and dynamic reclaim is required, then a lengthy process involving parameter settings (for example, `MEDIA` on the `STORCLAS` statement) is required. MVCs must be removed from the MVC pool or set into a read-only state, using `MVCMAINT`. Also, MVCs formatted in the wrong mode must be drained.
- The `INPTHRSH` parameter specifies the fragmentation level that makes partitioned MVCs eligible for Dynamic Reclaim. When a partitioned MVC's fragmentation falls between the `INPTHRSH` and `THRESH` values, it first becomes eligible for Dynamic Reclaim. If no space is freed on the initial reclaim attempt, the reclaim is retried when the fragmentation level has risen significantly. Should the fragmentation level reach the standard `THRESHOLD`, then a fall-back method of moving VTVs is performed.

`VTCS CONFIG RECLAIM` statements provide the global defaults for the system. You can override these defaults at the Storage Class and `MVCPool` level:

- The `STORCLAS INPLACE` parameter specifies whether Dynamic Reclaim is enabled for a Storage Class. Initially, you may wish to specify `INPLACE(NO)` in the `VTCS CONFIG RECLAIM` statement and `INPLACE(YES)` for selected Storage Classes, in order to evaluate the effects of Dynamic Reclaim. Subsequently, if you wish to apply Dynamic Reclaim to all eligible media, you can specify `INPLACE(YES)` on the `VTCS CONFIG RECLAIM` statement.
- The `INPTHRSH` parameter on the `POOLPARAM` statement can be used to set a Dynamic Reclaim threshold specifically for this set of MVC media. This is useful when you wish to treat certain MVCs differently from the `VTCS CONFIG RECLAIM` defaults.

Dynamic Reclaim can be disabled either globally at a system wide level or at the Storage Class level:

- To disable Dynamic Reclaim globally, specify `INPLACE(NO)` in the `VTCS CONFIG RECLAIM` statement and ensure that any Storage Class definition does not override this setting.
- To disable Dynamic Reclaim for one or more Storage Classes, change `INPLACE` to `NO` in the Storage Class policy statements.

---

**Note:** Any MVCs that are currently in use for migration will continue in partitioned format. If you wish to stop active MVCs from further migrates, then use the `MVCMaint` utility to mark them as read-only. Once the partitioned volumes are empty, they are automatically re-used in non-partitioned mode. An MVC becomes empty either when all VTVs have expired or the MVC has been `DRAINED`.

---

### STORMNGR and ACS Parameter Interactions

The `STORCLAS` statement now enables you to specify both the ACS and `STORMNGR` parameters. The statement is generally neutral to whether the target is to a local HSC tape library, a VLE or a remote tape library. The interaction of these parameters is as follows:

- If neither the ACS or `STORMNGR` parameters is specified, MVCs are selected from all available.
- If only the ACS parameter is specified, the selection of MVCs is restricted to the library serviced by the local HSC with the designated ACS number.
- If only the `STORMNGR` parameter is specified, then selection of MVCs is restricted to the specified VLE or remote library complex. For a remote library complex, this applies only to ACS 0.
- If both the ACS and `STORMNGR` parameters are specified, selection of MVCs is restricted to the library in the remote tapeplex with the designated ACS number.

The ACS and `STORMNGR` parameters allow the explicit specification of the location of the Storage Class. Whenever possible, use ACS and `STORMNGR` instead of implicit location selection by using the `MVCPool` and `MEDIA` parameters.

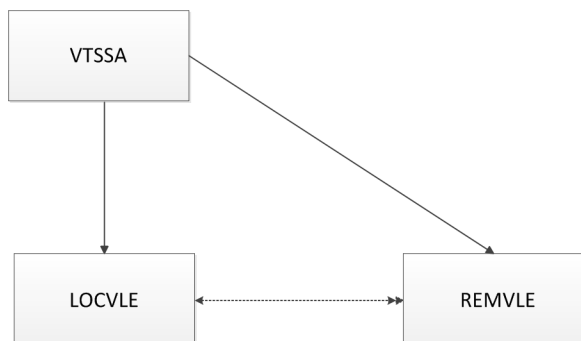
### FROMLST Parameter Usage

If you do not specify `FROMLST`, the default behavior is as follows:

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

Figure 3–113 shows a DR scenario with a local VLE (`LOCVLE`) and remote VLE (`REMVLE`) connected to `VTSSA`. You wish to migrate two VTV copies:

- First, a local copy from `VTSSA` to `LOCVLE`.
- Second, a copy via VLE-to VLE copy from `LOCVLE` to `REMVLE`.

**Figure 3–113 VLE to VLE Replication (Versus VTSS to VLE)**

To make the VTV copies as desired, do the following:

1. Create a `VTSSLST` statement to create a VTSS list that contains only VTSSA.

```
VTSSLST NAME (VSM2VLE) VTSS (VTSSA)
```

2. Create a `STORCLAS` statement that sends a VTV copy to REMVLE.

```
STORCLAS NAME (FORREMOT) STORMNGR (REMVLE)
```

3. Create a `MIGRVTV` statement that delays the migration copy to REMVLE.

```
MIGRVTV STOR (FORREMOT) IMMDELAY (360)
```

The migration delay is 360 minutes to allow the migration to the local site occur first; then the migration to the remote site is by VLE-to-VLE copy. Note that 360 minutes is just an example value, you can specify values of up to 9998 (do not specify 9999, because then the VTV is only migrated via automigration).

4. Create a `STORCLAS` statement that sends a VTV copy to LOCVLE.

```
STORCLAS NAME (FORLOCAL) STORMNGR (LOCVLE) FROMLST (VSM2VLE)
```

Note that the `FROMLST` parameter specifies that the local VTV copy is sourced from VTSSA.

5. Finally, create a `MGMTCLAS` statement that specifies two VTV copies, one to the local site and one to the remote site:

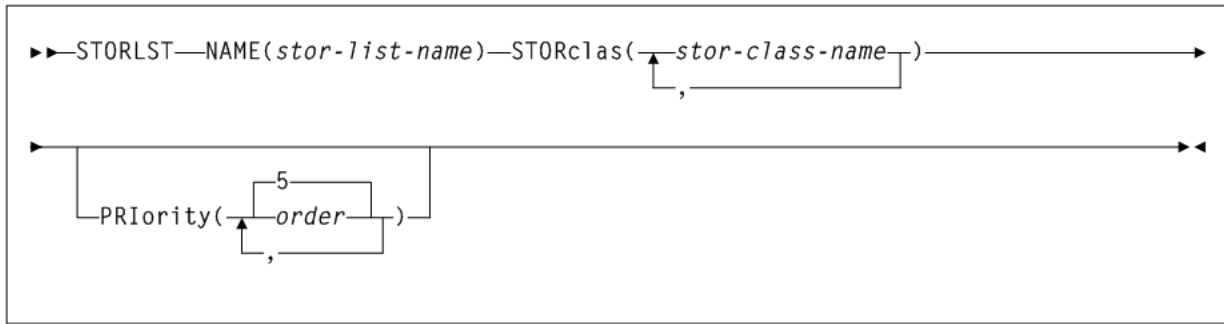
```
MGMTCLAS NAME (DRVLE) MIGPOL (FORLOCAL, FORREMOT)
```

## STORLST Control Statement

The `STORLST` control statement specifies a list of Storage Classes and their corresponding preferencing. It is loaded by the `MGMTDEF` command.

### Syntax

Figure 3–114 shows `STORLST` control statement syntax.

**Figure 3–114 STORLST control statement syntax****Parameters**

As shown in [Figure 3–114](#), the STORLST control statement includes the following parameters:

**NAME(stor-list-name)**

specifies the name of the Storage Class list.

*stor-list-name* indicates the list name (a maximum of 8 alphanumeric characters).

**STORclas(stor-clas-name)**

specifies one to ten Storage Classes on the Storage Class list.

*stor-clas-name* indicates the name of a Storage Class that you defined on the STORclas control statement.

**PRIority(order)**

a list of priorities corresponding to the Storage Classes specified on the STORclas parameter.

*order* indicates the specified priority. Valid values are 0 to 9 (highest priority), and the default is 5. You can assign the same priority to multiple Storage Classes. For example, if two Storage Classes both have a priority of 9, VTCS selects randomly from the two. A 0 (zero) priority specifies that VTCS selects the Storage Class only if all other Storage Classes are unavailable (for example, no free MVCs available for write).

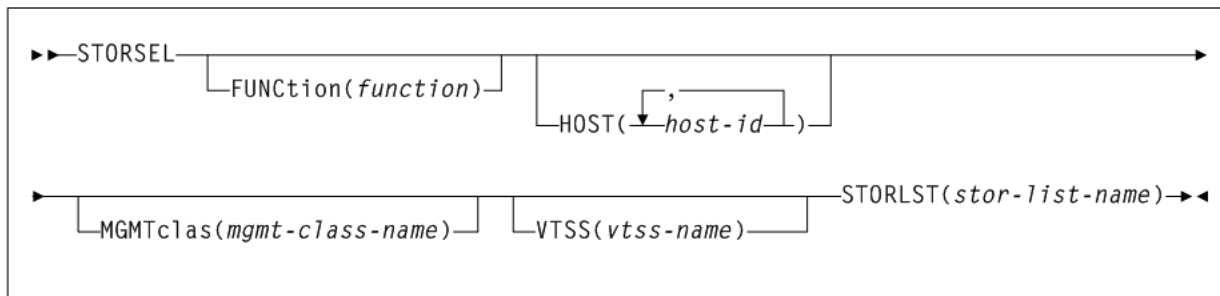
The Storage Class list is further qualified by the criteria specified by the MGMTclas and VTSS parameters of the STORSEL statement.

**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. It is loaded by the MGMTDEF command.

**Syntax**

[Figure 3–115](#) shows STORSEL control statement syntax.

**Figure 3–115** *STORSEL control statement syntax***Parameters**

As shown in [Figure 3–115](#), the `STORSEL` control statement includes the following parameters:

**FUNCTION(function)**

optionally, specifies the VSM function to which the rule applies. Only one function can be specified per statement. If this parameter is omitted, it applies to all functions. If all functions do apply, then it makes economical sense to omit the `FUNCTION` parameter, thus reducing the number of statements to 25% of what would otherwise need to be coded.

*function* indicates the function name, as described in the following table:

**Table 3–10** *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 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 <code>STORLST</code> 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 <code>STORLST</code> 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 <code>STORLST</code> statement. This list influences the list of MVC copies of a VTV to select the optimal MVC for consolidation of the VTV.

**HOST(host-id)**

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 parameter is not used, the statement applies to all hosts.

*host-id* indicates a host identifier (maximum 8 characters).

**MGMTclas(mgmt-class-name)**

optionally, specifies a Management Class.

*mgmt-class-name* indicates the name of a Management Class that you defined on the `MGMTclas` control statement.

**VTSS(vtss-name)**

optionally, specifies a VTSS.

*vtss-name* indicates the VTSS name, as follows:

- For automatic recalls, the VTSS where the recall is performed.
- For all other functions, the VTSS where the VTV previously resided. This may be determined from the *VTSS* value shown in the `Display VTV` output.

### **STORLST(stor-list-name)**

specifies a list of Storage Classes and their corresponding preferencing.

*stor-list-name* indicates the name of a Storage Class list that you defined on the `STORLST` control statement.

---



---

**Note:** The Storage Class list specified on the `STORLST` parameter is further qualified by the criteria specified by the `MGMTclas` and `VTSS` parameters.

---



---

## **VTSSLST Control Statement**

The `VTSSLST` control statement specifies a list of VTSSs and their corresponding preferencing. It is loaded by the `MGMTDEF` command.

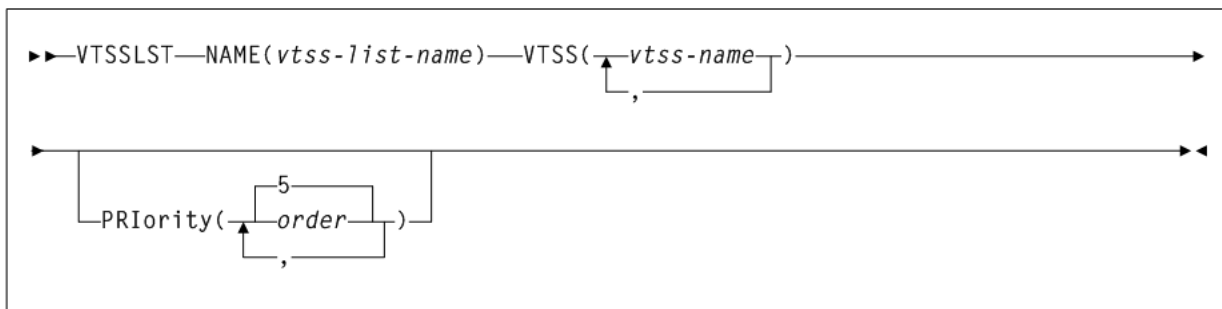
VTCS first determines a system priority for each VTSS, based on various factors. For example, whether the VTSS can service the request, whether the required resources are online/available or whether the VTSS is in a compromised state (high DBU).

When more than one VTSS has the highest system priority, `VTSSLST` priorities can be used to influence which VTSS is used. However, `VTSSLST` is only considered when there is an obvious choice of VTSSs with equal abilities to service a request.

### **Syntax**

Figure 3–116 shows `VTSSLST` control statement syntax.

**Figure 3–116** *VTSSLST control statement syntax*



### **Parameters**

As shown in Figure 3–116, the `VTSSLST` control statement includes the following parameters:

#### **NAME(vtss-list-name)**

specifies the name of the VTSS list.

*vtss-list-name* indicates the list name (a maximum of 8 alphanumeric characters).

#### **VTSS(vtss-name)**

specifies one to sixty four VTSSs on the VTSS list.



*vtss-name* indicates a VTSS name.

### **PRiority(order)**

optionally, lists priorities corresponding to the VTSS names specified on the *VTSS* parameter.

*order* indicates 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 a 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, all RTDs offline, or all VTDs are busy).

Specifying priority zero does not ensure that the VTSS will never be selected. To ensure that the VTSS is never selected, do not specify the VTSS in the *VTSS (. .)* list.

---

**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. It is loaded by the *MGMTDEF* command.

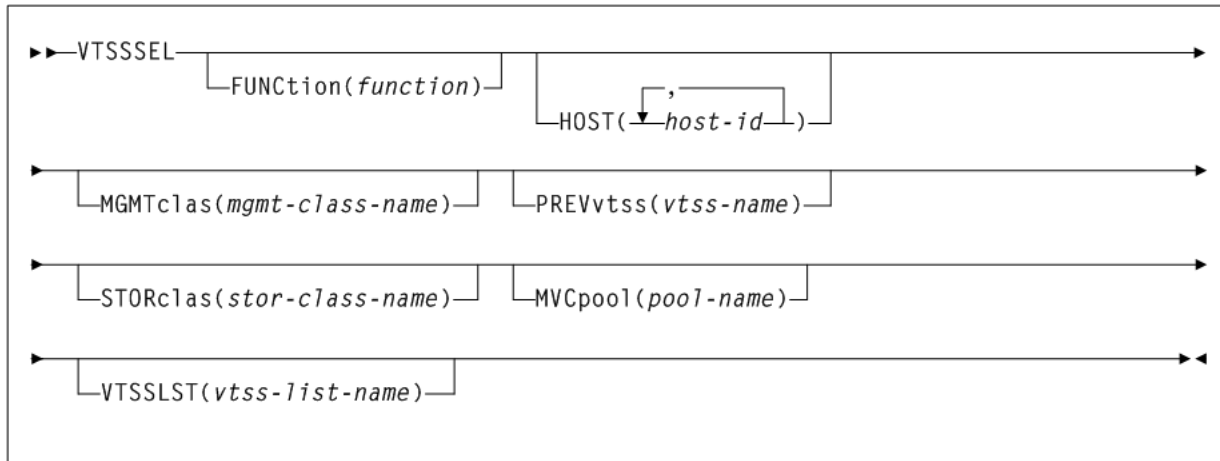
---

**Note:** *VTSSSEL* statements are honored only if the VTVs on the MVCs being processed are not resident in a VTSS. If the VTVs are resident, then VTCS ignores the *VTSSSEL* statement and migrates the VTV from the VTSS where it is resident.

---

### **Syntax**

Figure 3–117 shows *VTSSSEL* control statement syntax.

**Figure 3–117 VTSSSEL control statement syntax**

### Parameters

As shown in [Figure 3–117](#), the VTSSSEL control statement includes the following parameters:

#### **FUNCTION(function)**

optionally, specifies the VSM function to which the rule applies. Only one function can be specified per statement. If this parameter is omitted, it applies to all functions. If all functions do apply, then it makes economical sense to omit the FUNCTION parameter, thus reducing the number of statements to 25% of what would otherwise need to be coded.

*function* indicates the function name, as described in the following table:

**Table 3–11 VTSSSEL Functions**

Function	Description
SCRATCH	Applies to non-specific (scratch) VTV allocation. The list of eligible VTDs is determined by the specified VTSSLST statement. The PREVTSS, 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.
MOVEVTVS	Applies when an ARCHIVE or RECONCIL command has the MOVEVTV 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.

**Table 3–11 (Cont.) VTSSSEL Functions**

Function	Description
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(host-id)**

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 parameter is not used, the statement applies to all hosts.

*host-id* indicates 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(mgmt-class-name)**

optionally, specifies a Management Class.

*mgmt-class-name* indicates the name of a Management Class that you defined on the MGMTclas control statement.

**PREVvtss(vtss-name)**

optionally, specifies a VTSS where a VTV:

- Is resident
- Was migrated from

*vtss-name* indicates the VTSS name.

**STORclas(stor-clas-name)**

optionally, specifies a Storage Class and applies only when MVCs are used to select VTSSs.

*stor-clas-name* indicates the name of a Storage Class that you defined on the STORclas control statement.

**MVCpool(poolname)**

optionally, specifies a Named MVC Pool and applies only when MVCs are used to select VTSSs.

*poolname* indicates the name of an MVC Pool that you defined on the MVCpool control statement.

**VTSSLST(vtss-list-name)**

optionally, specifies a list of VTSSs and their corresponding preferencing.

*vtss-list-name* indicates the name of a VTSS list that you defined on the VTSSLST control statement.

# MIGrate

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

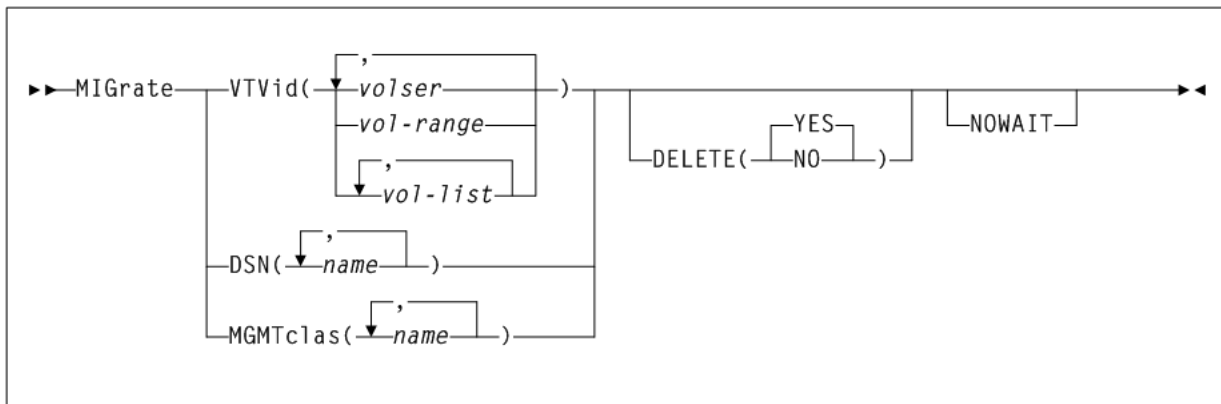
## Description

The MIGrate command migrates VTVs to MVCs.

## Syntax (Format 1)

Figure 3–13 shows MIGrate command syntax.

**Figure 3–118 MIGrate command syntax (Format 1)**



## Parameters (Format 1)

As shown in Figure 3–118, the MIGrate command (format 1) includes the following parameters:

**VTVid(volser, vol-range, or vol-list)**

specifies the VTVs to migrate.

*volser*, *vol-range*, or *vol-list* indicates the volsers of one or more VTVs. You can also specify one or more ranges.

**DSN(name)**

specifies data sets used to select VTVs to migrate.

*name* indicates the data set name. Table 3–12 describes the valid wild cards for data set names. You cannot address a member of a GDG using a wildcard.

**Table 3–12 Valid Wildcards**

Wildcard	Description
*	A qualifier or one or more characters within a qualifier. An asterisk can precede or follow a set of characters.

**Table 3–12 (Cont.) Valid Wildcards**

Wildcard	Description
**	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.

---

**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.

---

**MGMTclas(mgmt-class-name)**

specifies one or more Management Classes that determine one or more VTVs to migrate. MGMTclas, VTVID, and DSN are mutually exclusive.

*mgmt-class-name* indicates 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 3-222.

**DELETE**

optionally, 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). VTVs that are pinned in a VTSS by MGMTclas PINPOL will not be deleted.

**FORCE**

Delete VTSS resident copies. VTVs that are pinned in a VTSS by MGMTclas PINPOL will not be deleted.

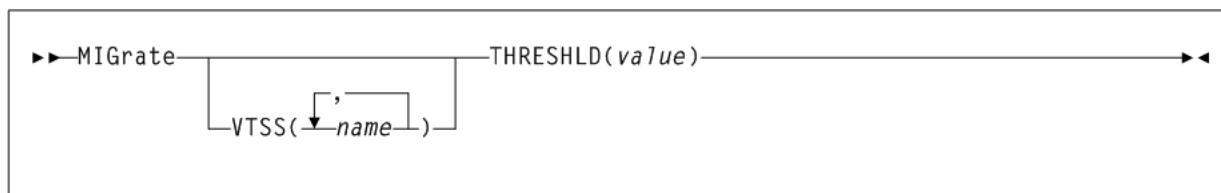
**NOWAIT**

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

**Syntax (Format 2)**

Figure 3–119 shows MIGrate command syntax.

**Figure 3–119 MIGrate command syntax (Format 2)**



## Parameters (Format 2)

As shown in [Figure 3–119](#), the MIGrate command (format 2) includes the following parameters:

**VTSS(name)**

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

*name* indicates the names of one or more VTSSs.

**THRESHLD(value)**

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

## MNTD

**Interfaces:**

- Console or PARMLIB only
- UUI Support: No

**Subsystem Requirements:**

Active HSC at FULL service level

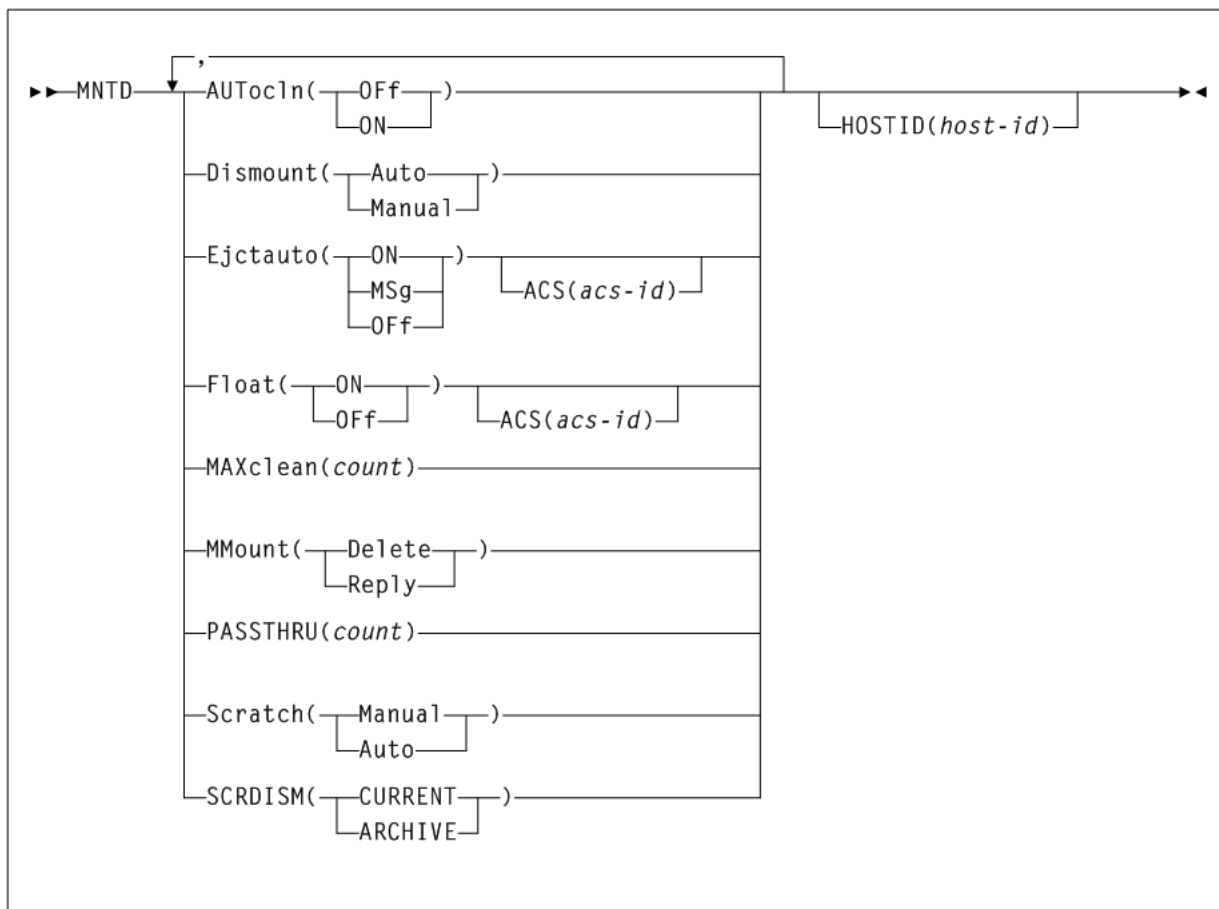
## Description

The MNTD command specifies how HSC processes mounts and dismounts of library volumes.

## Syntax

[Figure 3–120](#) shows MNTD command syntax.

Figure 3–120 MNTD command syntax



## Parameters

As shown in Figure 3–120, the MNTD command includes the following parameters:

### AUTocln

controls automated transport cleaning.

#### OFF

disables the automated cleaning function. OFF is the initial value for the HSC.

#### ON

enables the following HSC cleaning functions:

- the automated cleaning function, which detects when a transport requires cleaning and schedules the mount of a cleaning cartridge. For an LMU at level 22 or above, the cleaning occurs after the next dismount. For all other levels, the cleaning occurs prior to the next volume mount.
- the CLean command which is used to schedule the mount of a cleaning cartridge for a specified transport. See "Clean" on page 3-31 for more information.

### Dismount

specifies whether volumes are to be deleted automatically from the control data set when a dismount is requested in a manual mode LSM for a volume that was mounted by the robot before the LSM was modified offline.

**Auto**

indicates that volumes are to be automatically deleted from the control data set when a manual dismount is requested for a volume that was mounted by the robot. `Auto` is the initial value for the HSC.

**Manual**

directs the HSC to issue a message prompting the operator to decide whether the volume is to be deleted from the control data set when a manual dismount is requested for a volume that was mounted by the robot.

Setting `Dismount` to `Manual` is useful when an LSM is modified offline for only a short time. In this case, the operator can choose to not respond to the dismount message and leave the volume mounted on the transport. When the LSM is modified online the HSC re-drives the outstanding dismount request, causing the robot to dismount the volume and place it in a storage cell.

**Ejctauto**

controls automatic ejection of cleaning cartridges that have exceeded their maximum use.

In a multi-host environment, the `EJctauto` setting for a given ACS should be the same on all hosts. `EJctauto` status is not shared among the HSCs on different hosts.

**ON**

directs the HSC to automatically eject cleaning cartridges that have exceeded the number of times they can be used (as specified by the `MNTD MAXclean` or `VOLATTR MAXclean` parameters). `ON` is the initial value for the HSC.

**MSg**

directs the HSC to prompt the operator to either eject an over use-limit cartridge from the ACS or to keep a cleaning cartridge in the ACS.

**OFF**

directs the HSC to keep an over use-limit cleaning cartridge in the ACS. A message is issued displaying the cleaning cartridge's volser and describing this action.

**ACS(acs-id)**

optionally, specifies that the `EJctauto` settings apply only to the specified ACS. If this parameter is omitted, `EJctauto` settings affect all ACSs.

*acs-id* indicates a decimal value from 00 to 99 that identifies the ACS. A single digit *acs-id* can be entered.

**Float**

specifies whether the HSC is to select a new home cell location when it dismounts a volume that required a pass-thru when it was mounted.

In multi-host environments, the `Float` setting for a specific ACS should be the same on all hosts. `Float` status is not shared among the HSCs on different hosts.

**ON**

directs the HSC to select a new home cell location for the volume in the LSM where the dismount occurs (provided a cell is available). If no cells are available in the new LSM, a location is chosen in the nearest LSM with free cells or the volume can be forced to its original home cell. Setting `Float` to `ON` reduces the number of pass-thru operations. `ON` is the initial value for the HSC.



**Off**

directs the HSC to return the volume to its original home cell location when it is dismounted.

---

**Note:** The MNTD Float Off command is useful for remote ACS/CDS link down situations to avoid control data set integrity issues by making sure cartridges are returned to their original home cell locations. Refer to Oracle's ELS publication *ELS Legacy Interfaces Reference* for information about remote libraries.

---

**ACS(acs-id)**

optionally, specifies that the Float setting applies only to this ACS. If the ACS parameter is omitted, the Float setting affects all ACSs.

*acs-id* indicates a decimal value from 00 to 99 that identifies the ACS. A single digit *acs-id* can be entered.

**MAXclean(count)**

specifies the maximum number of times a cleaning cartridge is to be used.

- The Ejectauto setting in effect for the ACS controls how cleaning cartridges are handled when they exceed their maximum use.
- In a multi-host environment, the MAXclean setting should be the same on all hosts. The MAXclean value is not shared among the HSCs on different hosts.
- Follow the cartridge vendor's recommendations for the number of times a cleaning cartridge should be used.

*count* indicates a decimal value, in the range from 1 through 32767. The initial value for the HSC is 100.

The *count* value applies to each cleaning cartridge in the library. When a cleaning cartridge is used count number of times, it is not selected if there are cleaning cartridges compatible with the transports in the ACS that have been used less than count number of times. Over use-limit cleaning cartridges may be automatically ejected, depending on the MNTD Ejectauto setting.

**MMount**

specifies whether or not a mount message is issued during manual mode. This enables the operator to retain a manually mounted volume in the control data set.

**Delete**

generates a manual mode mount message which prompts the operator to respond D to delete the volume from the control data set, or I to ignore the mount request. Delete is the initial value of the HSC.

**Reply**

generates a manual mode mount message which prompts the operator to reply M to DOM the message and retain the volume in the control data set, or I to ignore the mount request.

---

**Note:** When MNTD MMount(Reply) is specified, the HSC action for a manual mode dismount is determined by the MNTD Dismount setting.

---

**PASSTHRU(count)**

specifies the maximum number of pass-thrus that can occur to allow cartridge archival if `SCRDISM(ARCHIVE)` is specified.

*count* indicates the maximum number of pass-thrus allowed for archival of cartridges. Allowable values are decimal in the range from 1 through 99. The initial value of the HSC is 1.

**Scratch**

determines how a scratch volume is selected to satisfy a scratch mount request for a manual mode LSM.

**Manual**

specifies that the operator must select a scratch volume when a scratch mount is requested for a manual mode LSM. `Manual` is the initial value for the HSC.

**Auto**

directs the HSC to select a scratch volume when a scratch mount is requested for a manual mode LSM. If `Scratch` is set to `Auto`, the HSC manual mount message indicates the cartridge `VOLSER` and cell location as if it were a request for a specific volume.

**SCRDISM**

specifies whether or not scratch volumes mounted in a 9310 or 9360 LSM are to be automatically archived to a larger or slower LSM upon dismount.

---

---

**Note:** 9740 LSMs cannot attach to any other type of LSM. Thus, `SCRDISM` does not affect mounts in ACSs containing 9740s.

---

---

**CURRENT**

indicates that scratch volumes mounted in a 9310 or 9360 LSM are to be dismounted according to the `MNTD Float` parameter setting. `CURRENT` is the initial value for the HSC.

**ARCHIVE**

indicates that scratch volumes mounted in a 9310 or 9360 LSM are to be archived into a larger or slower storage device. Archival occurs only if the number of pass-thrus does not exceed `PASSTHRU`. Archival of a cartridge can occur:

- from a 9360 to either a 9310 or a 4410
- from a 9310 to a 4410.

`SCRDISM(ARCHIVE)` overrides the `MNTD Float` parameter setting.

The `ARCHIVE` parameter does not affect dismounts in ACSs containing 9740 LSMs.

**HOSTID(host-id)**

optionally, identifies the host associated with the `MNTD` command. This enables you to restrict certain startup options to a specific host.

If this parameter is not specified, the command is executed by each host that accesses `PARMLIB`.

*host-id* indicates the host ID (the SMF system identifier for both JES2 and JES3).

---



---

**Note:** If the *host-id* does not match the host executing the command, a message is issued and the command is not processed.

---



---

## MODify

### Interfaces:

- Console or PARMLIB only
- UUI Support: No

### Subsystem Requirements:

Active HSC at FULL service level

## Description

The MODi fy command performs the following functions:

- modifies a CAP or an LSM online or offline to all hosts, independent of the issuing host
- starts dynamic hardware reconfiguration for a SL3000 or a SL8500 library
- adds, deletes, or updates an SL3000 or SL8500 ACS definition in the CDS that was preallocated using the SLLIBRY FUTRACS parameter

Dynamic hardware reconfiguration for the SL3000 and SL8500 libraries represents the portion of Near Continuous Operation (NCO) that enables you to dynamically add or delete drives and expansion panels.

Enter the MODi fy CONFIG command to activate dynamic hardware reconfiguration.

Refer to Oracle's ELS publication *Configuring HSC and VTCS* for more information about Near Continuous Operation and dynamic hardware reconfiguration.

---



---

**Caution:** Before you enter MODi fy CONFIG, perform a backup of the CDS.

---



---

It is possible to remove LSMs from the SL8500 configuration without performing a LIBGEN, MERGEcds, and recycle of the HSC. However, this operation requires assistance from Oracle StorageTek Support.

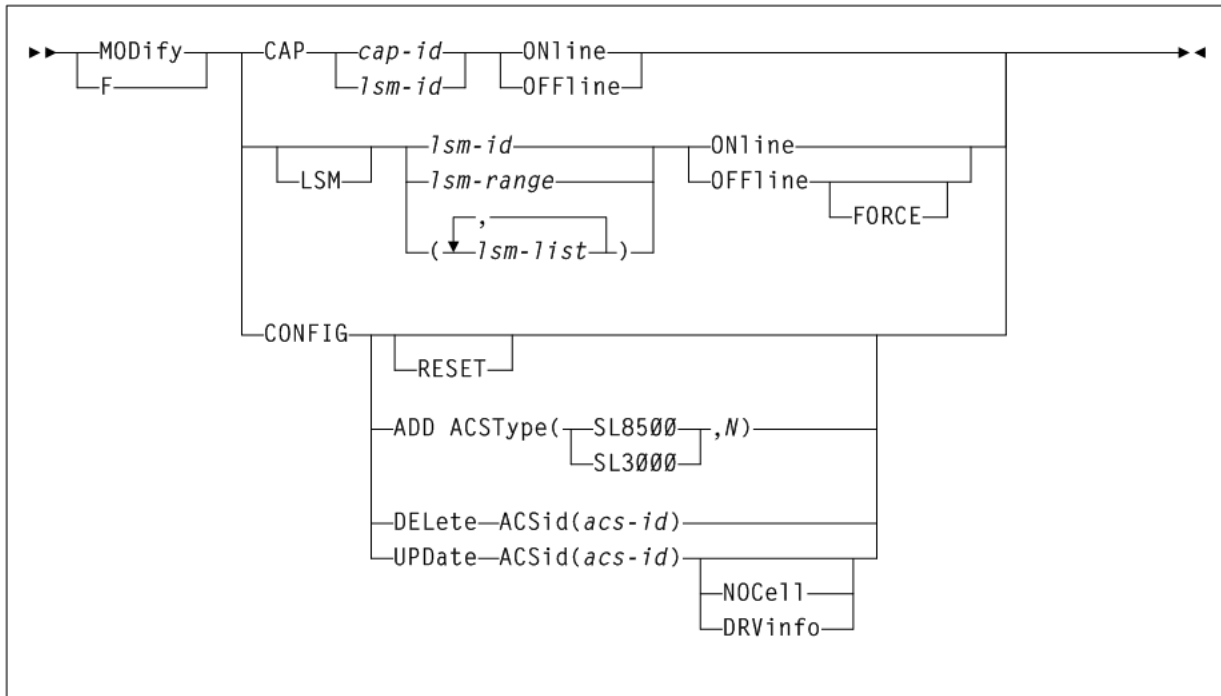
The MODi fy command differs from the Vary command in the following ways:

- The MODi fy command places a specified CAP or LSM online/offline globally to all hosts.
  - A CAP that is modified offline cannot be used for eject/enter processing.
  - An LSM that is modified offline must be operated in manual mode.
  - A manual mode LSM is still available for diagnostic requests from a host.
- The Vary command places a host station online/offline to an LMU.
  - A host that has all of its LMU stations varied offline is disconnected from all LSMs attached to the LMU.
  - An LSM can still be used to semi-automate cartridge handling for a disconnected host by issuing HSC commands from a connected host.

## Syntax

Figure 3–16 shows MODify command syntax.

Figure 3–121 MODify command syntax



## Parameters

As shown in Figure 3–121, the MODify command includes the following parameters:

### CAP

specifies that a CAP is to be modified online/offline to all hosts.

#### cap-id

a specific CAP to be modified online/offline. The format of a *cap-id* is *AA:LL:CC*, where *AA:LL* indicates the LSMid, and *CC* indicates the CAP. See "CAPid" on page B-1 for a list of valid values.

#### lsm-id

a standard CAP to be modified online/offline. The format of an *lsm-id* is *AA:LL*, where *AA* indicates the ACSid (decimal 00-99) and *LL* indicates the LSM number (decimal 00-99).

#### ONline

specifies that the CAP is to be modified online to all hosts. When a CAP is modified online, the HSC attempts to restore the CAP mode (automatic or manual) that was in effect when the CAP was modified offline.

#### OFFline

specifies that the CAP is to be modified offline to all hosts. Modifying a CAP offline places it in an unavailable state, preventing it from being allocated. The CAP mode (automatic or manual) is retained in the control data set.

- Be sure to specify the `CAP` parameter to modify a CAP offline. If the LSM is online, and you specify an `lsm-id` to modify the CAP offline but do not specify the `CAP` parameter, the LSM will be modified offline.
- Use this command only in an emergency. Make sure the CAP is not being used by another active process. Issuing the `MODify` command on an active CAP may cause the process using the CAP to receive errors.

**LSM**

optionally, indicates that one or more LSMs are to be modified online/offline to all hosts.

**lsm-id, lsm-range, or lsm-list**

one or more LSMs to be modified online/offline to all hosts. Each `lsm-list` element may be either a single `lsm-id` or an `lsm-range`. The elements in a list must be separated by commas or blanks, and the entire list must be enclosed in parentheses.

The format of an `lsm-id` is `AA:LL`, where `AA` indicates the ACSid (decimal 00-99) and `LL` indicates the LSM number (decimal 00-99).

**ONline**

specifies that the LSMs are to be modified online to all hosts. Modifying an LSM online places it in automatic mode. When an LSM is modified online, CAPs that were in auto mode before the LSM was modified offline are again placed in auto mode (unlocked state).

**OFFline**

specifies that the LSMs are to be modified offline to all hosts. Modifying an LSM offline places it in manual mode. CAPs in a manual mode LSM cannot be used, but the auto/manual state of each CAP is retained.

**FORCE**

specifies that the LSM(s) is to be modified offline immediately. `FORCE` is only used with the `OFFline` parameter to modify an LSM offline.

**CONFIG**

optionally, initiates the Near Continuous Operation (NCO) for hardware reconfiguration process for SL3000 and SL8500 libraries. This is a system wide change which propagates to all hosts connected to the CDS.

Refer to Oracle's ELS publication *Configuring HSC and VTCS* for more information about Near Continuous Operation and dynamic hardware reconfiguration. Before you enter `MODify CONFIG`, perform a backup of the CDS.

**no parms**

optionally, initiates the Near Continuous Operation (NCO) for hardware reconfiguration process for SL3000 and SL8500 libraries. This is a system wide change which propagates to all hosts connected to the CDS.

- Refer to Oracle's ELS publication *Configuring HSC and VTCS* for more information about Near Continuous Operation and dynamic hardware reconfiguration.
- Before you enter `MODify CONFIG`, perform a backup of the CDS.

**RESET**

resets internal flags for any host that may be failing as a hardware configuration is being performed. The `RESET` also terminates any active dynamic hardware

reconfiguration in the HSC complex during the hardware and software comparison phase. When the failing host comes back up, the new hardware configuration is automatically recognized and implemented.

A reset is intended only to reset a failing host and does not initiate the actual dynamic hardware reconfiguration operation. The `MODify CONFIG` command, without the `RESET` parameter, must be run separately to invoke dynamic hardware reconfiguration.

**ADD**

initiates an NCO configuration change to add one or more ACSs to the HSC configuration. The ACS added takes on the library characteristics of the library type specified by the `ACSType` parameter. The number of ACSs that can be added is limited by the number specified on the `SLLIBRY FUTRACS` parameter.

**ACSType**

specifies the library type to use for the configuration change. Options are `SL3000` or `SL8500`.

Additionally, you can specify `N` (separated by a comma) to indicate the number of ACSs to add, from 1 to 9. If this keyword is not specified, 1 is the default.

**DELeTe ACSid(acs-id)**

initiates an NCO configuration change to delete the disconnected ACS specified in the `ACSid(acs-id)` parameter.

After the `ACSid` is deleted, the ACS cannot be re-added if the ACS was defined by a `SLIACS` statement. If the ACS was added by a prior `MODify CONFIG ADD` command, the ACS can be re-added.

- If the `ACSid` is the not highest numbered ACS, the ACS is placed in "unallocated" status.
- If the `ACSid` is the highest numbered ACS, the ACS is entirely deleted and can be re-added as a different `ACSType`.
- If the `ACSid` is not the highest numbered ACS, the ACS becomes a place holder with a status of unallocated and can be re-added only as the same `ACSType`.

**UPDate ACSid(acs-id)**

initiates an NCO configuration change to update the ACS configuration specified in the `ACSid(acs-id)` parameter. This is a system wide change, and other hosts connected to the CDS are notified of the change.

The `ADD` and `DELeTe` parameters are only valid if the `FUTRACS` parameter was specified in the `SLLIBRY` macro during the `LIBGEN` process.

**NOCell**

Used in conjunction with ACS `acs-id` to bypass cell discovery within the ACS. This option is used only for drive and/or CAP changes and will reduce the reconfiguration time.

**DRVinfo**

Used in conjunction with ACS `acs-id` to update drive information (for example, serial number, worldwide name) for all drives of an ACS. This option is used only when replacing defective drives with drives of the same type.

---

---

**Note:** When this command option is invoked, it executes immediately and does not disrupt library activity.

---

---

## Mount

### Interfaces:

- Console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC at FULL service level

## Description

The `Mount` command mounts a scratch or specific volume on a transport, or mounts a VTV on a VTD and optionally assigns a Management Class to that VTV.

Do not use this command to manage a mount that appears to have been missed by a batch job. Instead, use the `SMC RESYNChronize` command to automate pending mounts. See "RESYNChronize" on page 2-64 for more information.

---

---

**Note:** The HSC `Mount` command does not support VTDs that are not defined in the `VTCS CONFIG`. The `SMC Mount` command must be used in place of the HSC `Mount` command for any VTDs that are not specified in the `VTCS CONFIG`.

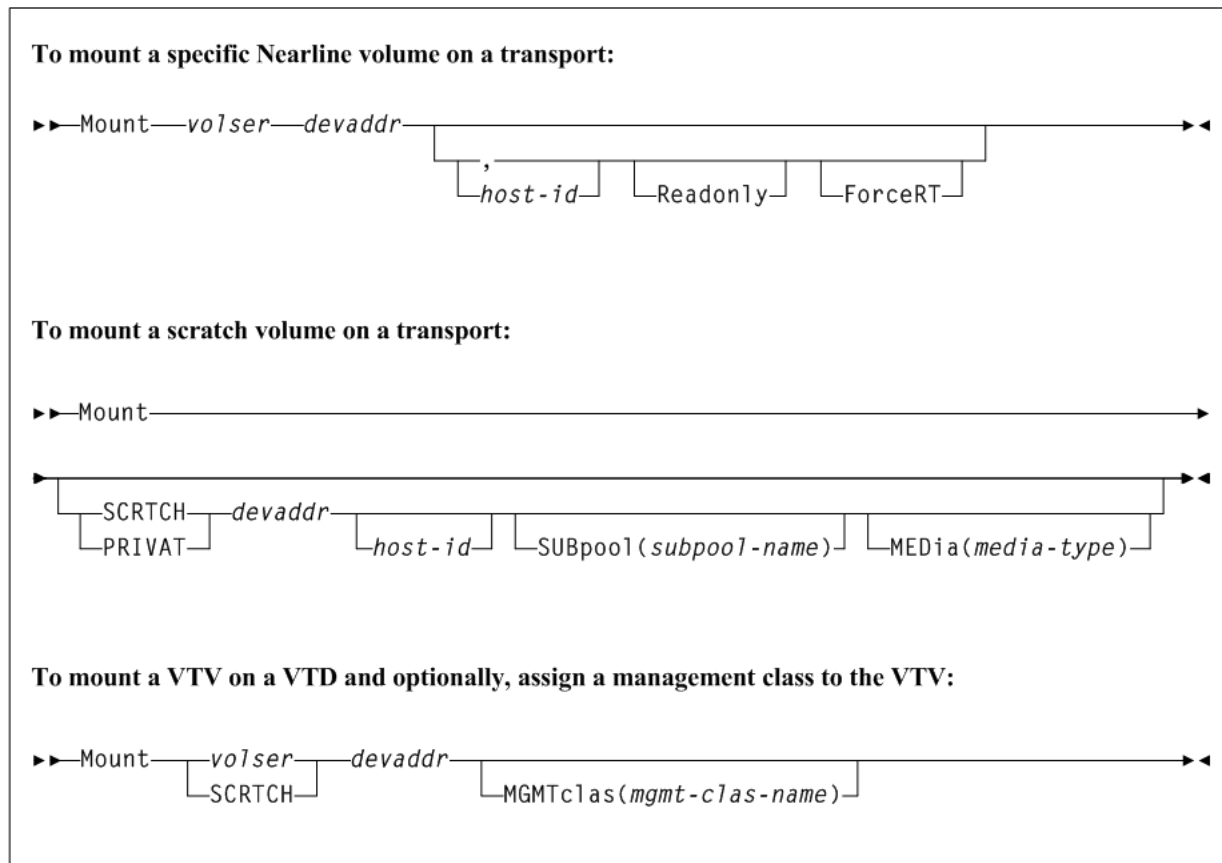
---

---

## Syntax

Figure 3–122 shows `Mount` command syntax.

Figure 3–122 Mount command syntax



## Parameters

As shown in Figure 3–120, the Mount command includes the following parameters:

### To Mount a Specific Volume on a Transport

#### volser

specifies the volume to be mounted.

#### devaddr

specifies the device address of the transport on which to mount the volume.

#### host-id

optionally, indicates that the volume is to be mounted on the device address defined to the specified host (the SMF system identifier for both JES2 and JES3).

---

**Note:** The *host-id* specified is only used for device address resolution. The *host-id* for subpool validation is obtained from the command origin.

---

#### Readonly

indicates that the volume is to be mounted for read-only access.



If you do not supply a *host-id* and do specify `Readonly`, a comma must be entered immediately before `Readonly` to indicate the missing operand. For example:

```
MOUNT 123456 0B0,,READONLY
```

### **ForceRT**

optionally, enables a volume to be mounted on a device with a different recording technique. The media type for the volume must be compatible with the device.

Caution should be used because it is possible to mount a volume written at a high density on a drive that is not capable of reading the high density.

## **To Mount a Scratch Volume on a Transport**

### **SCRATCH**

specifies a scratch volume is to be mounted.

### **PRIVAT**

specifies a scratch volume is to be mounted.

### **devaddr**

specifies the device address of the transport on which to mount the volume.

### **host-id**

optionally, indicates that the volume is to be mounted on the device address defined to the specified host (the SMF system identifier for both JES2 and JES3).

---



---

**Note:** The *host-id* specified is only used for device address resolution, The *host-id* for subpool validation is obtained from the command origin.

---



---

### **SUBpool(subpool-name)**

optionally, indicates the scratch volume is to be taken from a scratch subpool. If this parameter is not specified, then the behavior is dependent upon how scratch pools were defined:

- If `VOLPARM` specified, then the volume will be selected from the `DEFAULTPOOL` which contains all scratch tapes in the ACS that were specified in `VOLPARM` but were not assigned to a named subpool.
- If `VOLPARM` not specified (that is, `VOLATTR`, or `UX03` scratch pool definitions), then the volume will be selected from `subpool-0` which contains all scratch tapes in the ACS including both non-subpool and subpool volumes.

### **MEDIA(media-type)**

optionally, specifies the type of media for the scratch volume. The specified media must be compatible with the requested *devaddr*.

*media-type* indicates the media type.

See [Appendix A, "MEDIA, RECTech, and MODEL Values"](#) for a list of valid *media-type* values.

---



---

**Note:** If `MEDIA` is not specified, the next scratch cartridge is selected without regard to media type.

---



---

## To Mount a VTV on a VTD and Optionally, Assign a Management Class

**volser**

specifies a specific VTV volser. volser indicates the volser of a specific VTV.

**SCRATCH**

specifies the scratch VTV attribute (SCRATCH).

**devaddr**

specifies the MVS device address of the VTD to use to mount the VTV.

**MGMTclas(mgmt-class-name)**

optionally, specifies a Management Class you defined on the MGMTclas control statement.

*mgmt-class-name* indicates the Management Class name.

## MOVE

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at FULL service level

## Description

The MOVE command moves volumes to specified locations within the same LSM or to any other LSM within an ACS.

---

---

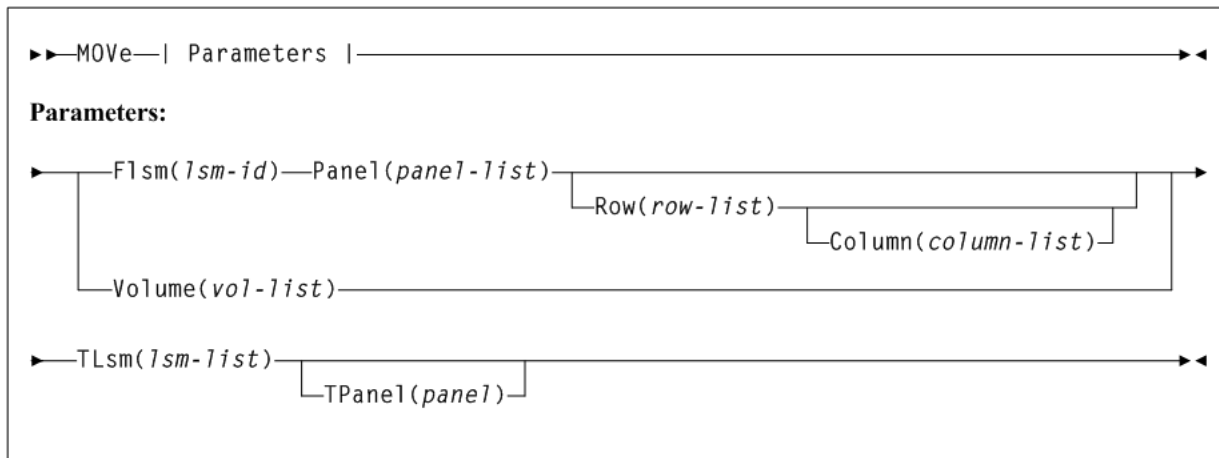
**Note:**

- Movement of volumes to all LSMs is done on a first-come first-served basis. As volumes are moved, should an LSM become fully populated before the move request is satisfied for that LSM, the move request continues with the movement of volumes designated for the next LSM specified in the request. An LSM is fully populated when all available cells contain tape cartridges. This process continues until the entire move request is completed or all destination LSMs are full.
  - Moves are performed one at a time to allow for other LSM activity.
- 
- 

## Syntax

Figure 3–18 shows MOVE command syntax.

Figure 3–123 MOVE command syntax



## Parameters

As shown in [Figure 3–123](#), the MOVE command includes the following parameters:

### Flsm(lsm-id)

specifies the “From” LSMid in the format *AA:LL* that the cartridges are to be moved from. If you specify the `Flsm` parameter, you may not specify the `Volume` parameter.

*lsm-id* indicates the LSM identifier name. An LSMid (*lsm-id*) is made up of the ACSid (decimal 00-99) and the LSM number (decimal 00-99) separated by a colon (:).

### Panel(panel-list)

specifies a panel number to be moved from. This parameter has a corequisite of the `Flsm` parameter and is required.

An entire panel can be emptied if the `Row` and `Column` parameters are not specified.

*panel-list* indicates a one or two digit panel number. Ranges are not valid. This parameter cannot contain a list if a list is specified for the `Row` or `Column` parameter.

Panels specified by *panel-list* are excluded as destination panels if cartridges are being moved within the same LSM.

A *panel-list* element is a one or two digit decimal number, specific to the LSM type. See "[Panels](#)" on page B-2 for a list of valid values.

### Row(row-list)

specifies a list of rows to be moved from. This parameter has a corequisite of the `Panel` parameter.

*row-list* indicates a one or two digit row number or list of row numbers. The maximum list allowed is four rows. However, this parameter cannot contain a list if a list was specified for the `Column` parameter. Ranges are not valid.

A *row-list* element is a one or two digit decimal number, specific to the LSM type. See "[Rows](#)" on page B-3 for a list of valid values.

### Column(column-list)

specifies a list of columns to be moved from. This parameter has a corequisite of the `Row` parameter and is optional. If this parameter is not specified, all columns will be moved for the rows specified.

*column-list* indicates a one or two digit column number or list of column numbers. This parameter cannot contain a list if a list was specified for the Row parameter. Ranges are not valid.

A *column-list* element is a one or two digit decimal number, specific to the LSM type. See "Columns" on page B-4 for a list of valid values.

**Volume(vol-list)**

optionally, specifies volumes to be moved.

*vol-list* indicates a list of volumes (a maximum of 300 can be specified) or a range of volumes. If you specify the Volume parameter, you may not specify the Flsm parameter.

**TLsm(lsm-list)**

specifies the target LSM(s). This is a required parameter. The LSMs are specified as *AA:LL*, where *AA* indicates the ACSid and *LL* indicates the LSMid. The ACSid:

- must be identical to the Flsm parameter "aa" (ACSid), or
- must be the same ACS in which the volume resides if the Volume parameter is specified.

*lsm-list* indicates a list of LSMs (a maximum of 24 can be specified). Ranges are invalid. An LSMid (*lsm-id*) is made up of the ACSid (decimal 00-99) and the LSM number (decimal 00-99) separated by a colon (:).

**TPanel(panel)**

optionally, specifies the panel in the TLsm to move the cartridge(s) to.

*panel* indicates a one or two digit panel number. This parameter cannot contain a list or range. See "Panels" on page B-2 for a list of valid values.

## MVCDRain

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

## Description

The MVCDRain command recalls all current and scratched VTVs from an MVC and, optionally, "virtually" ejects the MVC, making it unavailable for VSM use without physically ejecting it from the library. You can use this command to override the CONFIG RECLAIM CONMVC setting.

---

---

**WARNING:** If an error occurs on a VOLSAFE (WORM) tape, do not drain the MVC without the recommendation of Oracle/StorageTek Support. Once a WORM tape is drained, it can no longer be used for migration and must be replaced.

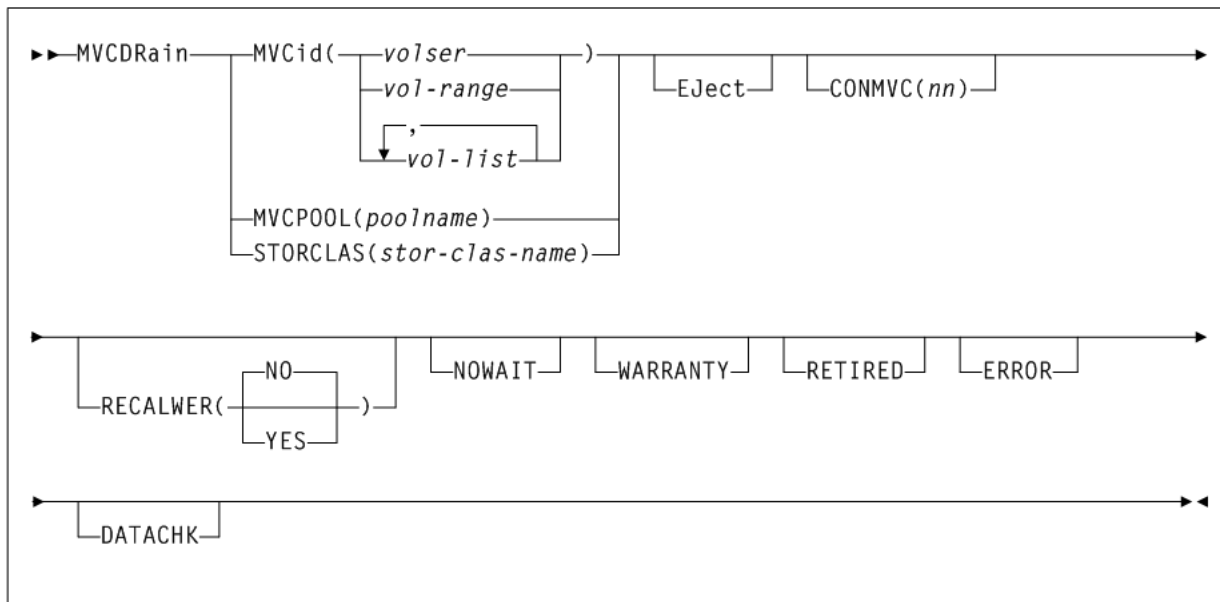
---

---

## Syntax

Figure 3–124 shows MVCDRain command syntax.

Figure 3–124 MVCDRAIN command syntax



## Parameters

As shown in [Figure 3–124](#), the MVCDRAIN command includes the following parameters:

### MVCid(volser, vol-range, or vol-list)

drain one or more MVCs by volser.

*volser*, *vol-range*, or *vol-list* indicates the volsers of one or more MVCs up to a maximum of 50.

### Eject

optionally, specifies that VTCS “virtually” ejects the MVC (the MVC will not be used for output).

For virtual MVCs (VMVCs):

- MVCDRAIN with the EJECT parameter physically deletes the VTVs.  
The current CDS backup can no longer be used as a DR baseline for VMVC content if you use the DRCHKPT utility or the CONFIG RECLAIM PROTECT parameter to protect CDS backup content for VMVCs, and then issue the MVCDRAIN EJECT command.
- MVCDRAIN without the EJECT parameter does not delete the VTVs, but updates the CDS record to show no VTVs on the VMVC.

### CONMVC(nn)

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

*nn* indicates the number of MVCs. Valid values are 1 to 99. If not specified, the default is the CONMVC value specified on the CONFIG RECLAIM statement.

### MVCPOOL(poolname)

drain the MVCs in the specified Named MVC Pool.

*pool-name* indicates the name of an MVC Pool that you defined on the `MVCPool` control statement. Refer to Oracle's ELS publication *ELS Legacy Interfaces Reference* for more information.

**STORCLAS(storclas-name)**

drain the MVCs in the specified Storage Class.

*stor-class-name* indicates the name of a Storage Class that you defined on the `STORCLAS` control statement; for more information, see "[STORCLAS Control Statement](#)" on page 3-238.

**RECALWER**

optionally, specifies whether VTCS recalls VTVs with read data checks.

**NO**

Do not recall VTVs with read data checks. This is the default.

**YES**

Recall VTVs with read data checks.

**NOWAIT**

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

**WARRANTY**

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

**RETIRED**

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

**ERROR**

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

**DATACHK**

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

## MVCMAINT

**Interfaces:**

- Utility only
- UUI Support: Yes

**Subsystem Requirements:**

- Active HSC/VTCS required if `RENVTSS` is specified
- Can run in batch-only mode when there are no hosts active (on any LPAR) using the CDS that is to be updated

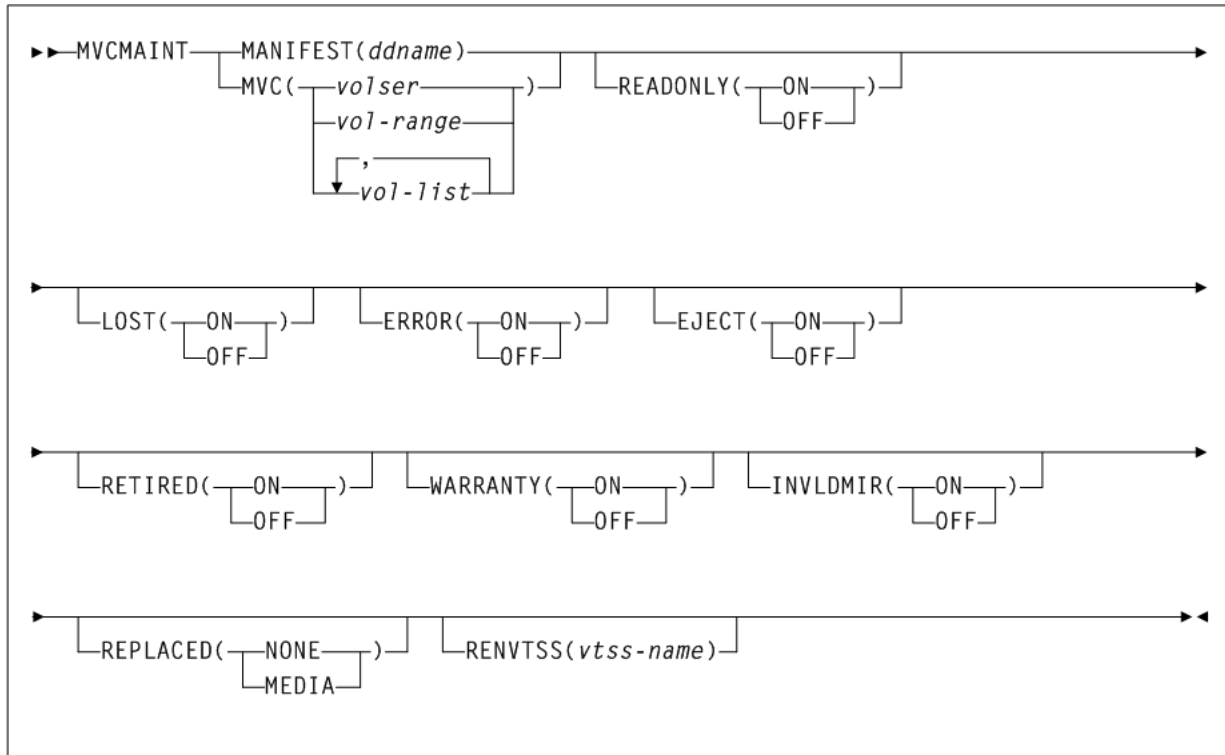
## Description

The `MVCMAINT` command specifies MVC attributes.

## Syntax

Figure 3–125 shows MVCMAINT command syntax.

Figure 3–125 MVCMAINT command syntax



## Parameters

As shown in Figure 3–125, the MVCMAINT command includes the following parameters:

### MANIFEST(ddname)

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

*ddname* indicates the DD name of the manifest file. The default is MANIFEST.

When specified, VTCS reads the MANIFEST file to discover which MVCs are to be updated (all MVCs in the manifest file). VTCS then updates the MVCs in the CDS with the actions specified by the remaining parameters (LOST, READONLY, etc)

### MVC(volser, vol-range, or vol-list)

optionally, specifies the MVCs whose read/write attribute is changed.

*volser*, *vol-range*, or *vol-list* indicates the volsers of one or more MVCs. You can specify a maximum of 50 MVCs.

### READONLY

optionally, sets the read/write status of the MVC.

#### ON

MVC is readonly.

**OFF**

MVCs is writable.

**LOST**

optionally, sets the “lost” status of the MVC.

**ON**

MVC is lost.

**OFF**

MVCs is not lost.

**ERROR**

optionally, sets the error status of the MVC.

**ON**

MVC is in error.

**OFF**

MVCs is not in error.

**EJECT**

optionally, sets the “logical eject” status of the MVC.

**ON**

MVC is “logically ejected”.

**OFF**

MVC is not “logically ejected”.

**RETIRED**

optionally, sets the “retired” status of the MVC.

**ON**

MVC is retired.

**OFF**

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

**WARRANTY**

optionally, sets the “expired warranty” status of the MVC.

**ON**

MVC’s warranty has expired.

**OFF**

MVC’s warranty has not expired.

---

---

**Note:** WARRANTY and RETIRED are mutually exclusive.

---

---

**INVLDMIR**

optionally, sets the invalid MIR status of the MVC.

**ON**

MIR is invalid.



**OFF**

MIR is not invalid.

**REPLACED**

optionally, updates various MVC fields in the MVC record after an MVC is physically replaced.

**NONE**

The MVC record is not updated.

**MEDIA**

The following MVC fields are updated:

- **MVC media type** 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
- **Date/Time the MVC was last mounted** is cleared

---



---

**Note:** This parameter requires the MVC to be empty (contain zero) and unmounted. REPLACED is mutually exclusive with the LOST, ERROR, EJECT, RETIRED, WARRANTY, and INVLDMIR parameters.

---



---

**RENTSS(*vtss-name*)**

optionally, sets the name of the owning VTSS to *vtss-name*.

*vtss-name* indicates the new VTSS name. This must be the name of an existing VTSS in your configuration.

---



---

**Note:** To use RENTSS, the host on which MVCMAINT is run must have access to the owning VTSS.

---



---

## Additional JCL Requirements

In addition to the required JCL definition statements described in "Invoking [SLUADMIN](#)" on page 5-1, the following definition statements apply to the MVCMAINT JCL:

**manifest file DD**

DD statement for the manifest file.

**Note:**

- MVCMAINT JCL 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 data sets specified in MVCMAINT JCL must match the CDS data sets used by HSC/VTCS.
- MVCMAINT JCL must specify the CDS to be updated if MVCMAINT is being executed in batch-only mode, i.e. when there are no hosts active (on any LPAR) using the CDS.

**Return Codes**

The MVCMAINT command includes the following return codes:

- 0 indicates that all requested updates for the MVC completed successfully.
- 4 indicates that one requested update for the MVC failed and at least one other requested update completed successfully.
- 8 indicates that 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 MVC 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.

**MVCMAINT Report**

The following is an example of an MVCMAINT report produced by the following command:

```
MVCMAINT MVC(022577-022579) READONLY=OFF
```

**Example 3-53 MVCMAINT Report**

```
MVC          RC
022577       04
022578       04
022579       04
```

**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
```

MVC	Number		Size	Mount	<----Status-->	<-----Last Mounted----->
Volser	VTVs	%Used %Avail	%Frag (GB)	Count	I B L D R U T M	Date Time VTSS
ACSID	Owner/					
Consolidate Time						
022577	0	0.00 99.96	0.04 .4	1	I - - - - C - -	2012Feb04 09:14:23
00	S1					
022578	0	0.00 99.96	0.04 .4	4	I - - - - U - -	2012Mar08 10:11:04 VTSS16
00	S1					

```

022579 0      0.00 99.96  0.04  .4      4      I - - - - U - -   2012Mar08 10:11:04
00      S1

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

As shown in [Example 3–53](#), 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

### Interfaces:

- Utility only
- UUI Support: Yes

### Subsystem Requirements:

Active HSC/VTCS

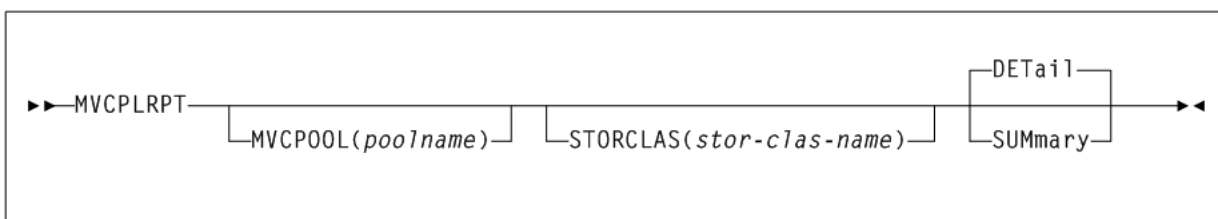
## Description

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

## Syntax

[Figure 3–126](#) shows MVCPLRPT command syntax.

**Figure 3–126 MVCPLRPT command syntax**



## Parameters

As shown in [Figure 3–126](#), the MVCPLRPT command includes the following parameters:

### MVCPOOL(poolname)

optionally, report on MVCs in the specified Named MVC Pool.

*poolname* indicates the name of an MVC Pool that you defined on the MVCPOOL control statement. Refer to Oracle’s ELS publication *ELS Legacy Interfaces Reference* for more information.

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

**STORCLAS(storclas-name)**

optionally, report on MVCs in the specified Storage Class.

*stor-clas-name* indicates the name of a Storage Class that you defined on the STORCLAS control statement. See Appendix A, "MEDia, RECtech, and MODel Values" for more information. You can also specify Storage Classes which used to exist to which media is still assigned.

To report on all Storage Classes, specify ALL or omit the STORCLAS parameter value.

The STORCLAS parameter enables you to filter the MVCs reported from the MVCPOOL, i.e. only those MVCs with a matching Storage Class are reported. If the STORCLAS parameter is specified then the summary counts section will contain Storage Class subtotals, either for every Storage Class encountered if the parameter value ALL or no value is specified, or for those media types where there are MVCs matching the specified value.

**DETail**

Report MVC detail lines (the default).

**SUMmary**

Report summary counts only.

**Named MVC Pool Report**

The following is an example of a report for Named MVC Pool CUST1POOL.

**Example 3-54 Named MVC Pool report**

```

SLUADMIN (7.3.0) TAPEPLEX=PRIMARY StorageTek Enterprise Library Software Utility SSYS=SLS0
PAGE 00002
TIME 08:23:02 Details MVCPOOL CUST1POOL
DATE 2016-05-10
MVC Avail Used Frag Unaval Size <----Status-->
Volser VTVs (GB) (GB) (GB) (GB) (GB) I B L D R U T M
711807 0 0 0 0 40 40 - - - - - J T - Not in
summary
740260 1880 3,925 506 67 0 4,498 P - - - - U - -
740261 133 4,075 423 0 0 4,498 P - - - - U - -
740262 1507 3,901 536 61 0 4,498 P - - - - U - -
740263 201 3,927 571 0 0 4,498 P - - - - U - -
740264 1867 3,995 450 53 0 4,498 P - - - - U - -
741031 0 4,498 0 0 0 4,498 P - - - - U - -
741044 0 0 0 0 5,000 5,000 P - - - R U - - Not in
summary
    
```

```

SLUADMIN (7.3.0) TAPEPLEX=PRIMARY StorageTek Enterprise Library Software Utility SSYS=SLS0
PAGE 00003
TIME 08:23:02 Details MVCPOOL CUST1POOL
DATE 2016-05-10

INITMVC MVCFree Threshold INPTHRSH Max MVC Start RECLaim
Yes 1 95 90 10 95 AUTO

ACS/ Media Total MVC Free MVC Available Used Fragmented Unavailable
Location Count Count Gb Gb Gb Gb
00 T1C000T2 6 1 24.321 2,486 181 0
00 ***Total 6 1 24,321 2,486 181 0
    
```

## Summary of MVCs by usage:

```

7 Total MVCs processed
0 Standard MVCs processed
7 Partitioned MVCs processed
0 Un-initialized MVCs processed
1 Free MVCs available
0 MVCs with status AUDIT
0 MVCs with status DRAIN
0 MVCs with status EXPORT
1 MVCs marked EJECTed
0 MVCs marked FULL
0 MVCs with maximum VTVs
1 MVCs marked Read-Only
0 MVCs with status Broken
0 MVCs with status Lost
0 MVCs with Expired Warranty
0 MVCs marked Retired
0 MVCs have Invalid MIRs
0 MVCs have Data-Checks
0 MVCs with status CONSOLIDATE
6 MVCs are USABLE

```

**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.

**Available (GB)**

the available space, specified in Gb, on the MVC.

**Used (GB)**

the used space, specified in Gb, occupied by current VTVs on the MVC.

**Fragmented (GB)**

the fragmented space, specified in Gb, occupied by non-current VTVs on the MVC. This space is not usable until it is reclaimed or the MVC is drained.

**Unavailable (GB)**

the unavailable space, specified in Gb, on the MVC. MVC with the following states are unavailable:

- FULL
- BROKE
- RETIRED
- DATACHECK
- READONLY

- Warranty Expired

**Size (GB)**

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

MVCs in a VLE have a nominal capacity of 250Gb. When VTVs are migrated with deduplication to a vMVC, the VTVs are divided into VTV unique data and data shared with other VTVs, which can reside on different vMVCs. As a result:

The sum of the VTVs can exceed the 250Gb vMVC media size, because the VTVs reference shared data that exists on other vMVCs.

Conversely, the sum of the VTV sizes on a full vMVC can be less than the 250Gb media size because the vMVC contains shared data for VTVs that reside on other vMVCs. For example, in an extreme case, a vMVC can have zero VTV copies but have zero available or fragmented space (the vMVC is full of shared VTV data).

**STATUS**

one or more of the following statuses:

**I**

The MVC has been initialized. A partitioned MVC is indicated by a P.

**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
- DDR swap failure
- 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 an `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 or 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.

**Not in summary**

MVC with the following states are not included in the MVCPLRPT summary:

- EJECTED
- READONLY
- DATACHECK
- Logically Deleted

**Summary for Storage Class or Named MVC Pool**

This section shows number of MVCs and total storage (Gb) by ACS and media type:

**Total MVC count**

Total number of MVCs.

**Free MVC count**

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.

**Available (GB)**

Available space, specified in Gb, on the MVC.

**Used (GB)**

Used space, specified in Gb, occupied by current VTVs on the MVC.

**Fragmented (GB)**

Fragmented space, specified in Gb, occupied by non-current VTVs on the MVC. This space is not usable until it is reclaimed or the MVC is drained.

**Unavailable (GB)**

Unavailable space, specified in Gb, on the MVC.

**Summary of MVCs by usage**

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

## MVCRPt

**Interfaces:**

- Utility only
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC not required

## Description

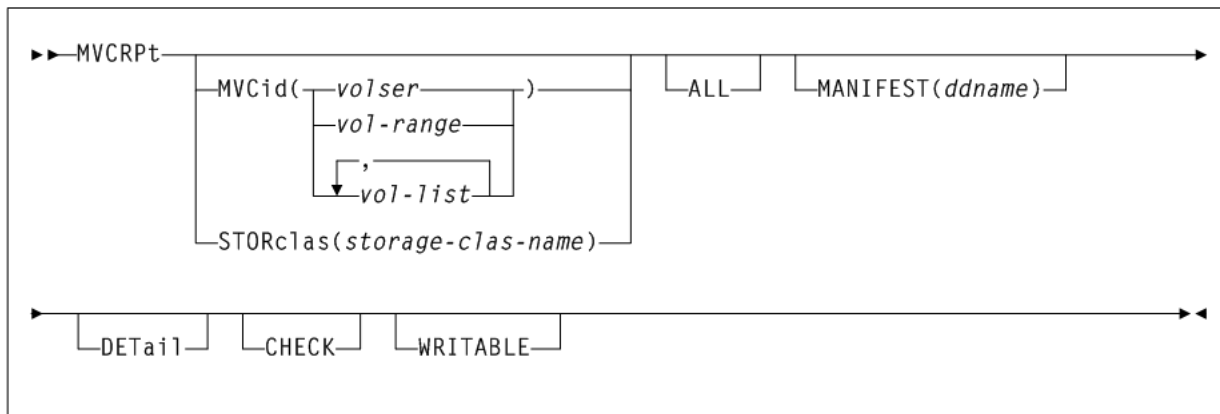
The `MVCRPt` command reports the status of your VSM system's MVCs.

## Syntax

[Figure 3–127](#) shows `MVCRPt` command syntax.



Figure 3–127 MVCRPt command syntax



## Parameters

As shown in [Figure 3–120](#), the MVCRPt command includes the following parameters:

### **MVCid(volser, vol-range, or vol-list)**

optionally, 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* indicates the volsers of one or more MVCs.

### **STORclas(storclas-name)**

optionally, specifies a storage class with the MVCs to be reported.

*stor-clas-name* indicates a 1-8 character storage class name.

---

**Note:** STORclas is mutually exclusive with MVCid, ALL, and MANIFEST.

---

### **ALL**

optionally, generate a report for all MVCs according to the selection criteria, even those that would otherwise be omitted.

### **MANIFEST(ddname)**

optionally, specifies the input DD name of the manifest file used to generate the report.

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

### **DETail**

optionally, produce a detailed MVC report. See [Example 3–56](#) for an example. If you do not specify this option, the default is to produce a summary MVC report. See [Example 3–55](#) for an example.

### **CHECK**

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

### **WRITABLE**

optionally, MVCRPt ignores read only MVCs on MVC detailed reports and when you specify the CHECK parameter.

## MVC Reports

The following sections describe the MVC summary and detailed reports produced by the MVCRPt command.

### MVC Summary Report

The following figure shows an example of an MVC summary report:

#### Example 3-55 MVCRPt MVC summary report

```

MVC      Number          Size Mount <----Status----> <-----Last Mounted----->
Location Owner/Comment/
Volser  VTVs %Used %Avail %Frag (GB) Count I B L D R U T M Date Time VTSS / ACS
ID Consolidate Time
DMV600  40 11.11 70.60 18.29 930.0 257 P - - - - U - - 2009Sep30 13:42:27 DVTSS16 00
DMV600
DMV601  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV602  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV603  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV604  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV605  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV606  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV607  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV608  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV609  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV610  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV611  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV612  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV613  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV614  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV615  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV616  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV617  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV618  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
DMV619  0 0.00 100.00 0.00 500.0 256 I - - - - U - - 2009Sep29 16:59:59 00
20 Initialized MVCs processed
  19 Initialized MVCs are standard format
   1 Initialized MVCs are partitioned
  0 Non-Initialized MVCs processed
  0 Non-Library MVCs processed
SLS0155I Condition code for utility function is 0

```

#### Fields

The MVCRPt MVC summary report includes the following 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.

**Size (GB)**

the size of the MVC (GB). This will only be determined after VTCS has used an MVC. N/A appears in this field until VTCS migrates a VTV to the MVC.

Note that MVCs in a VLE have a nominal capacity of 250 GB. When VTVs are migrated with deduplication to a VMVC, the VTVs are divided into VTV unique data and data shared with other VTVs, which can reside on different VMVCs. As a result:

The sum of the VTVs can exceed the 250 GB VMVC media size, because the VTVs reference shared data that exists on other VMVCs.

Conversely, the sum of the VTV sizes on a full VMVC can be less than the 250 GB media size because the VMVC contains shared data for VTVs that reside on other MVCs. For example, in an extreme case, a VMVC can have zero VTV copies but have zero available or fragmented space (the MVC is full of shared VTV data).

**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. MVCs written in partitioned format are indicated by the letter 'P' under this column.

**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.

---

**WARNING: If the error occurred on a VOLS SAFE (WORM) tape, do not drain the MVC without the recommendation of Oracle/StorageTek Support. Once a WORM tape is drained, it can no longer be used for migration and must be replaced.**

---

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, refer to Oracle's ELS publication *ELS 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 `MVCMMAINT` 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.

---

---

**WARNING:** If the error occurred on a VOLSAFE (WORM) tape, do not drain the MVC without the recommendation of Oracle/StorageTek Support. Once a WORM tape is drained, it can no longer be used for migration and must be replaced.

---

---

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.

---

---

**WARNING:** If the error occurred on a VOLSAFE (WORM) tape, do not drain the MVC without the recommendation of Oracle/StorageTek Support. Once a WORM tape is drained, it can no longer be used for migration and must be replaced.

---

---

#### **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 `MVCMMAINT` utility to set `READONLY (OFF)`.

The MVC does not have the appropriate SAF rules set to enable VTCS to update the MVC. Correct the error and use the `MVCMMAINT` utility to set `READONLY (OFF)`. Refer to Oracle's ELS publication *Installing ELS* for information about defining a security subsystem user ID for HSC, SMC, and VTCS.

#### **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, or 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.

---



---

**WARNING:** If the error occurred on a VOLSAFE (WORM) tape, do not drain the MVC without the recommendation of Oracle/StorageTek Support. Once a WORM tape is drained, it can no longer be used for migration and must be replaced.

---



---

#### Last Mounted

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

#### Location/ACS ID

the ACS containing the MVC, or the VLE containing remote MVC volumes or VMVCs in VLEs.

**\*\*MANY\*\*** indicates that an MVC with a specific volser has been discovered in more than one location. MVCs in this state are marked unusable.

#### 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.

#### Initialized MVCs processed

the total number of initialized MVCs. Also includes a breakdown of the number of MVCs that are standard format or partitioned.

#### Non-Initialized MVCs processed

the number of non-initialized MVCs.

#### Non-library MVCs processed

the number of non-library MVCs processed.

#### MVC Detailed Report

The MVC detailed report provides all the fields from the MVC summary report and a separate section that lists additional fields. The following figure shows an example of these additional fields from an MVC detailed report.

#### Example 3–56 MVCRPt MVC detailed report (additional fields)

Vtv Volser	Size (MB)	Block Id	Management Class	Migration Date	Block Count	Message
---------------	--------------	-------------	---------------------	-------------------	----------------	---------

DX0000	1802.24	0000/00000000	MF	2010Aug16	29078
DX0001	282.62	0000/00007196	MF	2010Aug16	4567
DX0002	3850.24	0000/0000836D-0001	MF	2010Aug16	62109
DX0003	1802.24	0001/0001760A	MF	2010Aug16	29077
DX0004	2764.80	0001/0001E79F-0002	MF	2010Aug16	44602
DX0005	778.24	0002/000295D9	MF	2010Aug16	12561
DX0006	1802.24	0002/0002C6EA	MF	2010Aug16	29077
DX0007	180.22	0002/0003387F	MF	2010Aug16	2915
DX0008	2826.24	0002/000343E2-0003	MF	2010Aug16	45593
DX0009	3850.24	0003/0003F5FB-0004	MF	2010Aug16	62109
DX0010	266.24	0004/0004E898	MF	2010Aug16	4303
DX0011	3440.64	0004/0004F967-0005	MF	2010Aug16	55503
DX0012	1290.24	0005/0005D236	MF	2010Aug16	20819
DX0013	282.62	0005/00062389	MF	2010Aug16	4567
DX0014	3850.24	0005/00063560-0006	MF	2010Aug16	62109
DX0015	1802.24	0006/000727FD	MF	2010Aug16	29077
DX0016	282.62	0006/00079992	MF	2010Aug16	4567
DX0017	1802.24	0006/0007AB69	MF	2010Aug16	29077
DX0018	2826.24	0006/00081CFE-0007	MF	2010Aug16	45593
DX0019	778.24	0007/0008CF17	MF	2010Aug16	12561
DX0020	1802.24	0007/00090028-0008	MF	2010Aug16	29077
DX0021	2826.24	0008/000971BD	MF	2010Aug16	45593
DX0022	3850.24	0008/000A23D6-0009	MF	2010Aug16	62109
DX0023	282.62	0009/000B1673	MF	2010Aug16	4567
DX0024	3850.24	0009/000B284A-000A	MF	2010Aug16	62109

25 VTVs found for MVC:DMV600 Last migrate:2010Aug16 13:00:04

181 unused / 11 used partitions found for MVC:DMV600

Unused partitions :

```

--- --- --- --- --- --- --- --- --- --- --- 00B 00C 00D 00E 00F 010 011 012 013 014 015 016 017
018 019 01A 01B 01C 01D
01E 01F 020 021 022 023 024 025 026 027 028 029 02A 02B 02C 02D 02E 02F 030 031 032 033 034 035
036 037 038 039 03A 03B
03C 03D 03E 03F 040 041 042 043 044 045 046 047 048 049 04A 04B 04C 04D 04E 04F 050 051 052 053
054 055 056 057 058 059
05A 05B 05C 05D 05E 05F 060 061 062 063 064 065 066 067 068 069 06A 06B 06C 06D 06E 06F 070 071
072 073 074 075 076 077
078 079 07A 07B 07C 07D 07E 07F 080 081 082 083 084 085 086 087 088 089 08A 08B 08C 08D 08E 08F
090 091 092 093 094 095
096 097 098 099 09A 09B 09C 09D 09E 09F 0A0 0A1 0A2 0A3 0A4 0A5 0A6 0A7 0A8 0A9 0AA 0AB 0AC 0AD
0AE 0AF 0B0 0B1 0B2 0B3
0B4 0B5 0B6 0B7 0B8 0B9 0BA 0BB 0BC 0BD 0BE 0BF

```

### Fields

The MVCRPt MVC detailed report includes the following additional fields:

#### VTV Volser

the volsers of the VTVs on the MVC.

#### Size (MB)

the uncompressed size of the VTV (MB).

#### Block ID

the partition id and logical block ID of the beginning of the VTV on the MVC. If there are VTVs spanning partitions, the spanned to partition is shown to the right of the blockid location.

#### 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 Oracle, 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 Oracle StorageTek software support.

**Used/Unused Partitions**

Summary of used and unused partitions for the MVC.

## OFFload

**Interfaces:**

- SLUADMIN utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

## Description

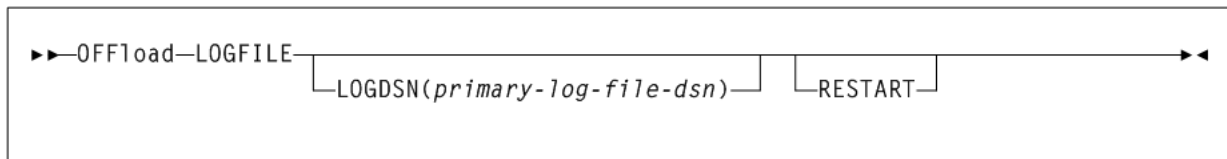
The OFFload command offloads transaction records from active log file(s) used by the HSC Transaction Logging Service and saves them in a sequential data set so that space in the log files can be reused.

If logging is currently active, you must run the OFFload LOGFILE utility to change log file names.

## Syntax

Figure 3–128 shows OFFload command syntax.



**Figure 3–128 OFFload command syntax**

## Parameters

As shown in [Figure 3–128](#), the OFFload command includes the following parameters:

### LOGFILE

This keyword is required. Generally, OFFload LOGFILE is issued without parameters. The utility selects the input log data set from the active log data sets as recorded in the CDS.

### LOGDSN(primary-log-file-dsn)

optionally, offloads the log file without using or updating the CDS. Use this parameter in any case where the CDS is restored, or is otherwise considered to be unreliable.

*primary-log-file-dsn* indicates the data set name for the primary log file.

For example:

```
OFFLOAD LOGFILE LOGDSN(hsc.log1)
```

### RESTART

optionally, restarts the offload, selecting the input log file from among the group of log data sets that were active when the failed offload started. It also uses the block number range used by the failed offload.

Use this parameter to recover the records from the log data set(s) that become inactive when a failed offload causes a log data set switch.

## Additional JCL Requirements

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statements apply to the OFFload LOGFILE JCL:

### SLSOFFLD

specifies the output data set

It is recommended that no DCB be specified for this data set. If a DCB is specified, the RECFM and LRECL are ignored.

## OPTION TITLE Control Statement

The OPTION TITLE control statement specifies an identifying string for a definition data set. This identifying string can be any information that helps the user describe the contents of the definition data set. You can specify this statement in a definition data set along with the following:

- LMUPDEF
- MGMTDEF
- TREQDEF (SMC)

You can also specify this control statement using the following legacy HSC commands, as described in the *ELS Legacy Interfaces Reference*:

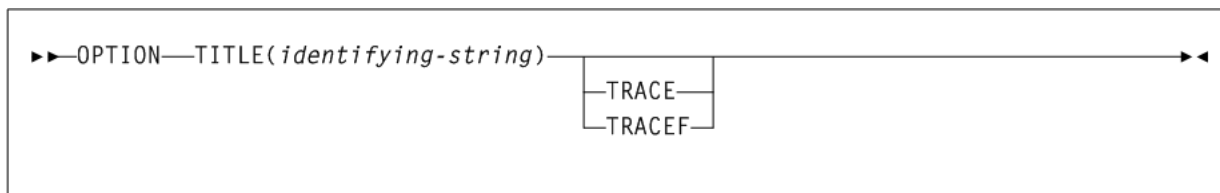
- MVCDEF
- SCRDEF
- VOLDEF

The `OPTION TITLE` statement must be placed in a definition data set; it cannot be issued as an operator command. If more than one `OPTION` statement is specified in the definition data set, only the identifying string of the last `OPTION` statement encountered is retained.

## Syntax

Figure 3–129 shows `OPTION TITLE` control statement syntax.

**Figure 3–129** *OPTION TITLE control statement syntax*



## Parameters

As shown in Figure 3–129, the `OPTION TITLE` control statement includes the following parameters:

### identifying-string

indicates an identifying string for the definition data set. The maximum length of the identifying string is fifty characters. If the identifying string includes one or more spaces, or any characters other than alphabetic, numeric, or national (for example, \$, @, #), it must be enclosed in quotes.

### TRACE or TRACEF

optionally, used to control internal tracing of HSC table lookups. During problem resolution, Software Support may request that you specify one of these parameters.

## OPTion

### Interfaces:

- Console or PARMLIB only
- UI Support: No

### Subsystem Requirements:

Active HSC at `BASE` or `FULL` service level

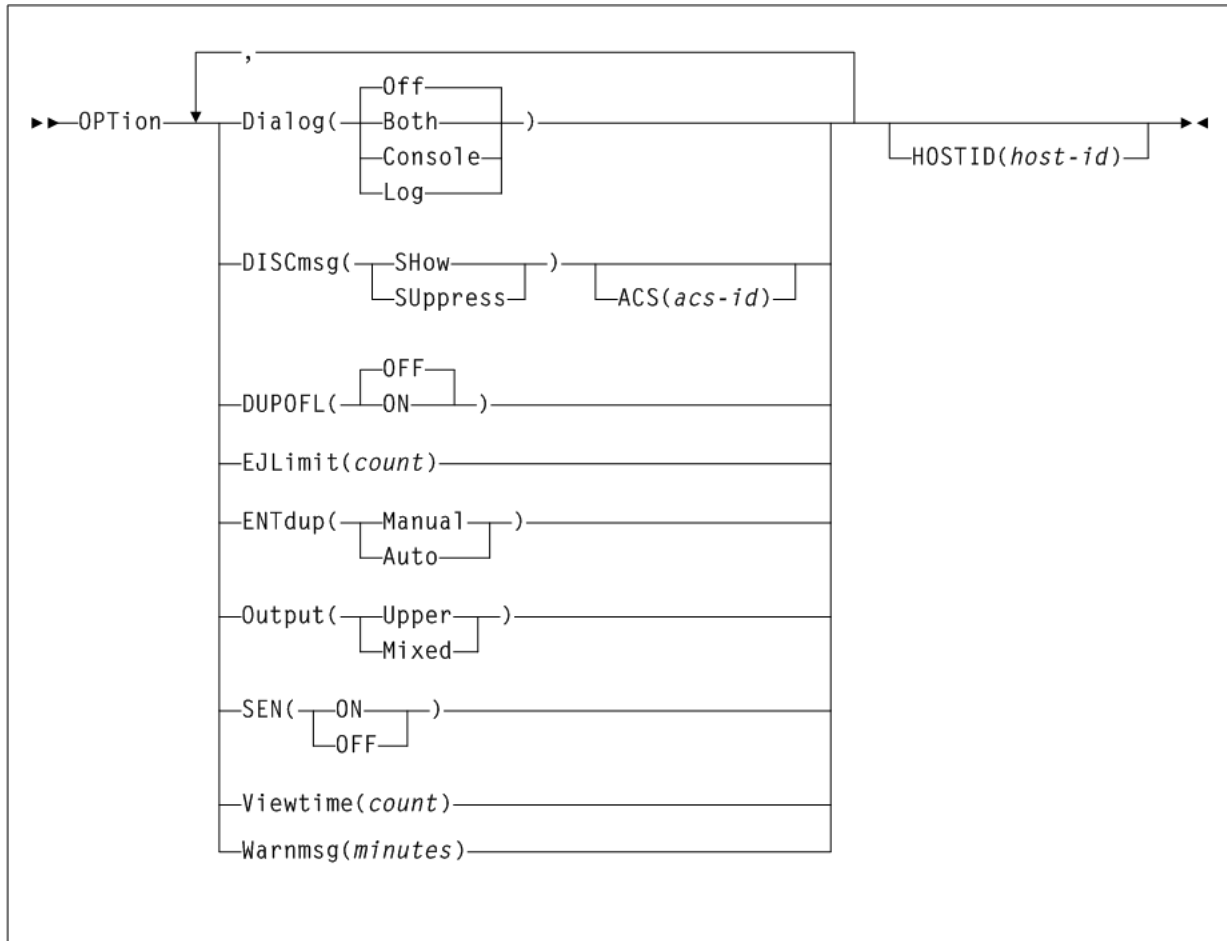
## Description

The `OPTion` command sets or changes general HSC options.

## Syntax

Figure 3–130 shows OPTion command syntax.

**Figure 3–130** OPTion command syntax



## Parameters

As shown in Figure 3–130, the OPTion command includes the following parameters:

### Dialog

enables HSC messages to be written to the operator console and/or the system log. These messages indicate that the HSC is waiting for an active task to complete before the HSC terminates. The initial Dialog value is set by the EXEC statement during HSC startup.

#### Off

specifies that messages are not to be written. Off is the default value for the HSC if the Dialog parameter is not specified on the EXEC statement.

#### Both

specifies that messages are written to the system log and the operator console.

#### Console

specifies that messages are written to the operator console only.

**Log**

specifies that messages are written to the system log only.

**DISCmsg**

specifies whether or not the ACS *acs-id* is disconnected message (SLS1664A) is displayed for the specified ACS.

This option is useful when an ACS has been added to the LIBGEN in advance to provide for future expansion. Entering DISCmsg enables the user to turn off the display for those ACSs that will not be connected until later.

**SHow**

Displays the ACS *acs-id* is disconnected message (SLS1664A) for an ACS. This is the initial value.

**SUppress**

Suppresses the ACS *acs-id* is disconnected message (SLS1664A) for an ACS.

**ACS(acs-id)**

optionally, specifies the ACS for which the command applies. If ACS is not specified, the DISCmsg setting entered (Show or Suppress) applies to all ACSs.

*acs-id* indicates the decimal ACSid value (00-99). If ACS is not specified, all ACSs are affected.

**DUPOFL**

optionally, enables the duplicate VOLSER process to continue when the VOLSER being entered into the CAP shows in the CDS that it exists in an ACS that is disconnected, or in an LSM that is offline.

**OFF**

disables the duplicate VOLSER process. This is the default.

**ON**

enables the duplicate VOLSER process.

**EJLimit(count)**

indicates the maximum number of cartridges that can be specified on one EJECT command.

*count* indicates the maximum number of cartridges. Allowable values are 1 through 9999. The HSC initial value is 300.

**ENTdup**

specifies whether the HSC prompts the operator when an enter operation finds a duplicate VOLSER in the control data set, but cannot locate the cartridge in the ACS.

When someone attempts to enter a cartridge with a VOLSER that duplicates an entry in the control data set, the HSC attempts to locate the original cartridge. The ENTdup option determines how the HSC responds when the cartridge is not in its home cell, is not selected, and is not errant.

**Auto**

instructs the HSC to delete the cartridge in the control data set and allow the enter to continue.

**Manual**

instructs the HSC to issue a console message when a duplicate VOLSER is entered. The message prompts the operator to decide whether the HSC should delete the

cartridge in the control data set and allow the enter to continue, or eject the duplicate cartridge. `Manual` is the initial value for the HSC.

### **Output**

specifies whether the output messages are displayed on the console in uppercase or mixed case.

#### **Upper**

specifies uppercase. `Upper` is the initial value for the HSC.

#### **Mixed**

specifies mixed case.

### **SEN**

enables or disables the Significant Event Notification (SEN) facility. Refer to Oracle's ELS publication *ELS Programming Reference* for more information about the SEN.

#### **ON**

enables the SEN facility.

#### **OFF**

disables the SEN facility. In this mode, the SEN facility accepts registration requests. However, invocation of listener programs is not performed. This is the initial value for HSC.

### **Viewtime(count)**

specifies the length of time in seconds the system is to hold an LSM camera in one location when the `VIEW` command is issued. See "[VIEW](#)" on page 3-370 for more information.

*count* indicates the number of seconds. Allowable values for *count* are decimal in the range from 5 through 120. The initial value for the HSC is 10 seconds.

The `VIEW` command enables the user to override the `OPTion Viewtime` setting for a single viewing operation.

An SMF record (subtype 8) is written when the `VIEW` command is issued to document robot activity.

### **Warnmsg(minutes)**

sets the number of minutes between scratch depletion messages.

*minutes* indicates the number of minutes. Allowable values for *minutes* are 1 through 65535. The initial value is 5 minutes.

### **HOSTID(host-id)**

optionally, identifies the host associated with the `OPTion` command. This enables you to restrict certain startup options to a specific host.

If this parameter is not specified, the command options are applied to each host that accesses `PARMLIB`.

*host-id* indicates the host identifier (the SMF system identifier for both JES2 and JES3).

---

**Note:** If the *host-id* does not match the host executing the command, a message is issued and the command is not processed.

---

## PITCOPY

### Interfaces:

- SLUADMIN utility only
- UUI Support: No

### Subsystem Requirements:

Active HSC not required

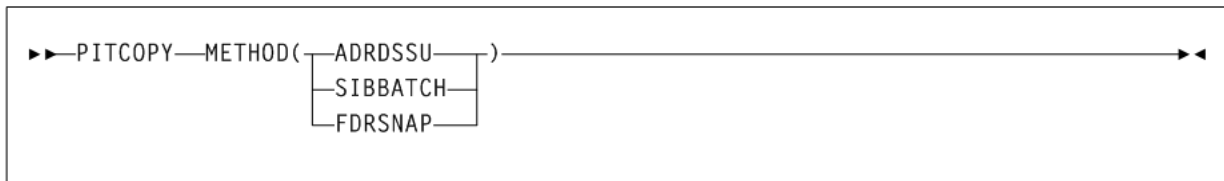
## Description

The PITCOPY command uses a utility snapshot feature to produce a point-in-time copy of the CDS. PITCOPY ensures CDS backup integrity by performing a CDS RESERVE to ensure that any ongoing updates are serialized. The PITCOPY function backs up only the primary CDS. Refer to Oracle's ELS publication *Managing HSC and VTCS* for more information about how to use the PITCOPY utility.

## Syntax

Figure 3–131 shows PITCOPY command syntax.

Figure 3–131 PITCOPY command syntax



## Parameters

As shown in Figure 3–131, the PITCOPY command includes the following parameters:

### METHOD

specifies the snapshot copy utility to use for the PIT copy.

#### ADRDSSU

DFSMSdss SnapShot/DFSMSdss FlashCopy (ADRDSSU)

#### SIBBATCH

StorageTek SVAA SnapShot (SIBBATCH) utility

#### FDRSNAP

specifies that messages are written to the operator console only.

## Additional JCL Requirements

In addition to the required JCL definition statements described in "Invoking SLUADMIN" on page 5-1, the following definition statements apply to the PITCOPY JCL:

### SLSPARMP

specifies a parameter file containing control statements to back up the CDS data set defined as SLSCNTL. Note that this may not be the primary CDS data set at the time the backup is taken.

**SLSPARMS**

specifies a parameter file containing control statements to back up the CDS data set defined as `SLSCNTL2`.

**SLSPARMB**

specifies a parameter file containing control statements to back up the CDS data set defined as `SLSSTBY`.

**SYSIN**

specifies a file to be output by the `PITCOPY` function and used as input to the snapshot copy function. When the `PITCOPY` function determines the primary CDS, it copies the appropriate `SLSPARMS` file into `SYSIN`.

---

---

**Note:**

- Other definition statements are required by the snapshot utility you are requesting.
  - If you use Oracle's StorageTek SVAA SnapShot function, it is recommended that you use the `SOURCE` option specifying the data set name, rather than the `INDDNAME` option specifying the `DDNAME`; or that you always include the `DD` statements for the CDS data sets. If the dynamic CDS allocation feature is used, the primary CDS is automatically allocated to `SLSCNTL`. Therefore, a control statement in `SLSPARMS`, for example, to back up the `SLSCNTL2 DD`, would result in backing up a CDS copy that is not the primary.
- 
- 

## RECall

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

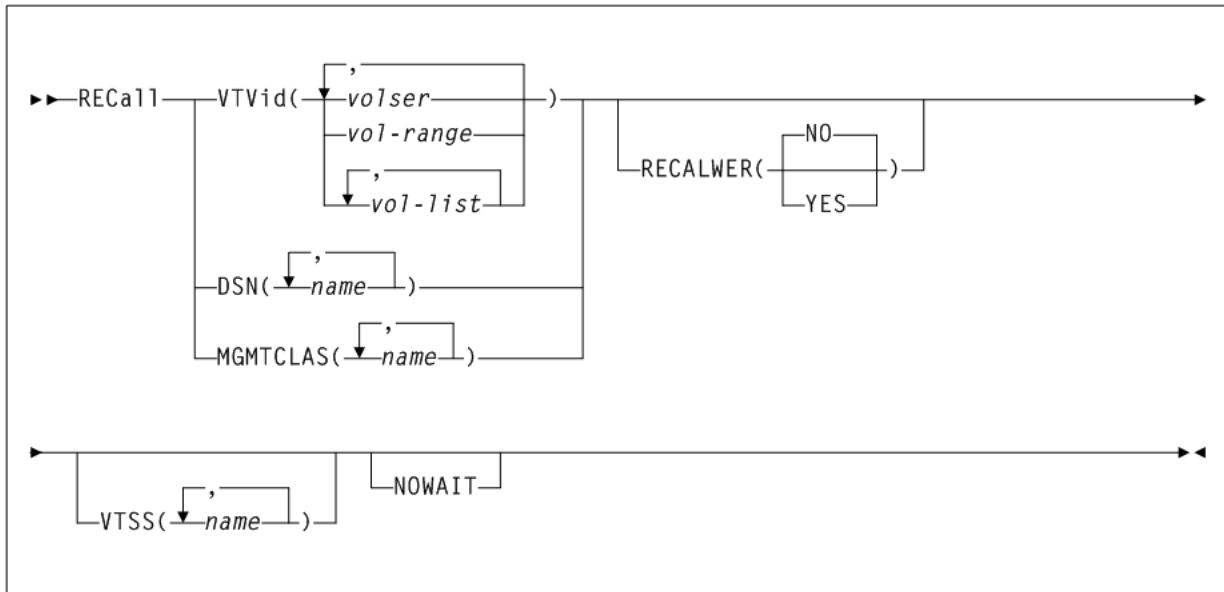
## Description

The `RECall` command performs demand recalls of VTVs to a VTSS.

## Syntax

Figure 3–132 shows `RECall` command syntax.

Figure 3–132 RECall command syntax



## Parameters

As shown in [Figure 3–132](#), the RECall command includes the following parameters:

### VTVID(volser, vol-range, or vol-list)

specifies the VTVs to recall.

*volser*, *vol-range*, or *vol-list* indicates the volsers of one or more VTVs. You can also specify one or more ranges.

### DSN(name)

specifies data sets used to select VTVs to recall.

*name* indicates the data set name. [Table 3–12](#) describes the valid wildcards for data set names.

### MGMTCLAS(mgmt-class-name or mgmt-class-list)

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

*mgmt-class-name* or *mgmt-class-list* indicates 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 3-222.

MGMTCLAS, VTVID, and DSN are mutually exclusive.

### RECALWER

optionally, specifies whether VTCS recalls VTVs with read data checks.

#### NO

Do not recall VTVs with read data checks. This is the default.

#### YES

Recall VTVs with read data checks.

### VTSS(name)

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 VTSSs, 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.

*name* indicates the names of one or more VTSSs.

**NOWAIT**

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

## RECLaim

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

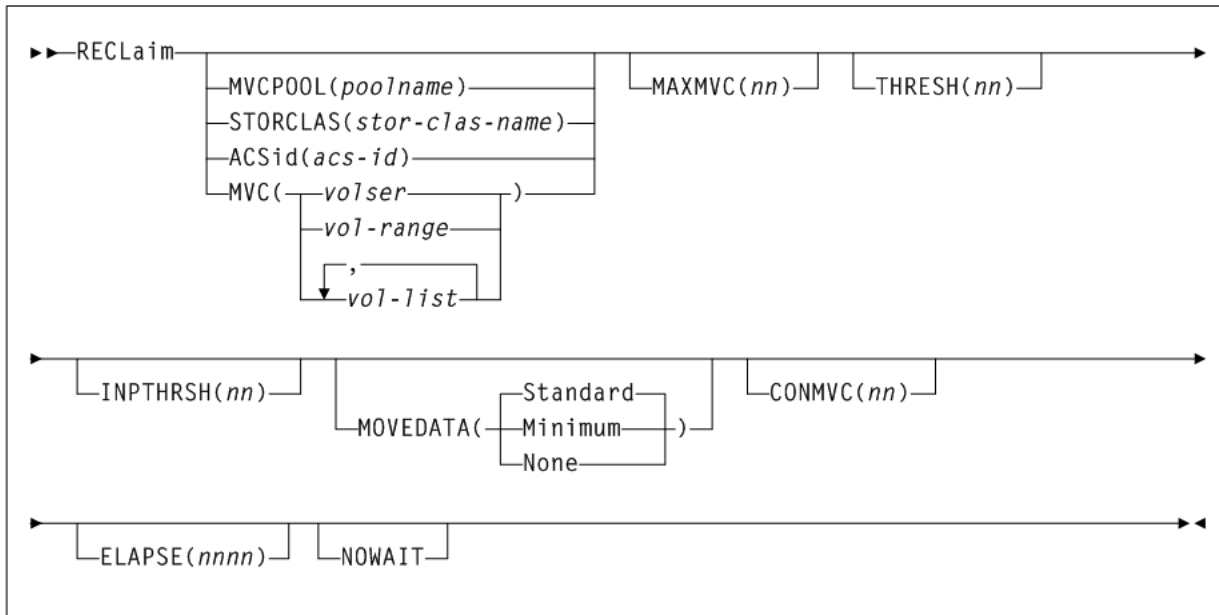
## Description

The RECLaim command performs demand MVC space reclamation. This command can also override the CONFIG RECLaim settings for the THRESHLD, INPTHRSH, MAXMVC, and CONMVC parameters.

## Syntax

Figure 3–133 shows RECLaim command syntax.

Figure 3–133 RECLaim command syntax



## Parameters

As shown in [Figure 3–133](#), the RECLaim command includes the following parameters:

### MVCPOOL(poolname)

reclaim the MVCs in the specified Named MVC Pool.

*poolname* indicates the name of a Named MVC Pool that you defined on the MVCPOOL control statement. Refer to Oracle's ELS publication *ELS Legacy Interfaces Reference* for more information.

### STORCLAS(stor-clas-name)

reclaim the MVCs in the specified Storage Class.

*stor-clas-name* indicates the name of a Storage Class that you defined on the STORCLAS control statement; for more information, see "[STORCLAS Control Statement](#)" on page 3-238.

### ACSid(acs-id)

reclaim the eligible MVCs in the specified ACS.

*acs-id* indicates a two digit decimal ACS identifier from 00-99.

### MVC(volser, vol-range, or vol-list)

reclaim the specified MVC(s).

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

optionally, specifies the maximum number of MVCs that will be processed by a single space reclamation task.

*nn* indicates the maximum number of MVCs. 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)**

optionally, specifies the percentage of fragmented space that makes an MVC eligible for demand or automatic reclamation.

*nn* indicates the percentage of fragmented space. Valid values are 4 to 98. If not specified, the `CONFIG RECLAIM` value (or default) is used.

### **INPTHRSH(nn)**

optionally, specifies the percentage of fragmented space that makes an MVC in partitioned format eligible for dynamic reclaim processing. This value overrides the global `INPTHRSH` value specified in the `CONFIG RECLAIM` statement. If this parameter is not specified, the current active global `INPTHRSH` value is used.

*nn* indicates a percentage between 3 and 97. This value must be less than the `THRESHLD` value.

---



---

**Note:** If `INPTHRSH` is specified, `THRESHLD` must also be specified. Both `INPTHRSH` and `THRESHLD` play roles when space reclamation processes MVCs in partitioned format. Refer to Oracle's ELS publication *Configuring HSC and VTCS* for more information.

---



---

### **MOVEDATA**

optionally, specifies the extent of VTV data movement to occur as a result of an MVC being reclaimed by the `RECLaim` command.

#### **Standard**

Move all VTV copies from the MVC being reclaimed to the currently applicable Storage Classes as specified in the `MGMTclas` or `STORclas` definitions.

A new `STORclas` definition may have been explicitly implemented by `MGMTDEF` changes, or it could be as a result of an `ARCHIVE` policy specifying a new `STORclas` taking effect.

This is the default.

#### **Minimum**

For an ALP MVC being reclaimed `INPLACE`, if the fragmentation percentage space in a partition exceeds the `THRESH` value specified on the `RECLaim` command, move all VTV copies from the MVC being reclaimed to the currently applicable Storage Classes as specified in the `MGMTclas` or `STORclas` definitions. This option supplies the Defrag function for an ALP MVC.

If a VTV spans partitions, it may have a segment in a low-use partition. For a partition to be eligible for defrag, the total size of all VTVs that reside partially or completely within the partition is used to calculate the used space associated with the partition.

For defrag to occur, the total associated used space must be less than 100 minus the `THRESH` value percentage of the partition size. This prevents the `RECLaim` from potentially recalling a large amount of data residing in multiple partitions to free up a small amount of space in a single partition.

Do not move VTV copies from a VLE VMVC to a new Storage Class.

**None**

Do not move VTV copies from any MVC. This turns off all data movement.

Empty partitions on an ALP MVC that is being reclaimed `INPLACE` are freed. Additionally, free space on VLE VMVCs becomes available.

**CONMVC(nn)**

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

*nn* indicates the number of MVCs. Valid values are 1 to 99. If not specified, the default is the `CONMVC` value specified on the `CONFIG RECLAIM` statement.

**ELAPSE(nnnn)**

optionally, 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.

*nnnn* indicates the reclaim time in minutes. Valid values are 1 to 1440. If not specified, there is no time limit on the reclaim process.

**NOWAIT**

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

## Output

The following is an example of the output produced by the `RECLaim` command:

**Example 3-57 RECLaim output**

```
SLS5011I Reclaim - -----MVC Reclaim Request 1,085-----
SLS5011I Reclaim - Targeting MVC pool:PREAL ACS:00 Media:STK1R
SLS5011I Reclaim - MVCs unused=1,990 MVCs empty=0 MVCs failed=0 MVCs
readonly=0
SLS5011I Reclaim - MVCs draining=0 MVCs full=10 MVCs in-use=0
SLS5011I Reclaim - MVCs deleted space=8 MVCs needing reclaim=2
SLS5011I Reclaim - 0 MVCs have up to 7% wasted space
SLS5011I Reclaim - 1 MVCs have up to 15% wasted space
SLS5011I Reclaim - 1 MVCs have up to 22% wasted space
SLS5011I Reclaim - 0 MVCs have up to 30% wasted space
SLS5011I Reclaim - 2 MVCs have up to 37% wasted space
SLS5011I Reclaim - 1 MVCs have up to 45% wasted space
SLS5011I Reclaim - 1 MVCs have up to 52% wasted space
SLS5011I Reclaim - 0 MVCs have up to 60% wasted space
SLS5011I Reclaim - 0 MVCs have up to 67% wasted space
SLS5011I Reclaim - 0 MVCs have up to 75% wasted space
SLS5011I Reclaim - MVC DMV408 selected and contains 5 VTVs
SLS5011I Reclaim - MVC DMV401 selected and contains 10 VTVs
SLS5011I Starting processing
```

**Fields**

`RECLaim` command output includes the following fields:

**MVCs unused=n**

the number of non-initialized MVCs.

**MVCs empty=n**

the number of empty MVCs.

**MVCs failed=n**

the number of MVCs that have experienced a Data Check, or are being Audited, or are Lost, or are Broken.

**MVCs readonly=n**

the number of read only MVCs.

**MVCs draining=n**

the number of MVCs that are being Drained or Reclaimed.

**MVCs full=n**

the number of MVCs that are marked Full, or contain MAXVTV VTVs, or are reclaim candidates.

**MVCs in-use=n**

the number of MVCs that contain data and are not Full.

**MVCs deleted space=n**

the number of fragmented MVCs.

**MVCs needing reclaim=n**

the number of MVCs selected for Reclaim.

## RECONcil

**Interfaces:**

- Utility only
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

## Description

The `RECONcil` command reconciles VTV media and location (moves VTVs from one Storage Class to another).

`RECONcil` performs a series of cross-checks between the MVCs, VTVs, `STORCLAS` and `MGMTCLAS` definitions.

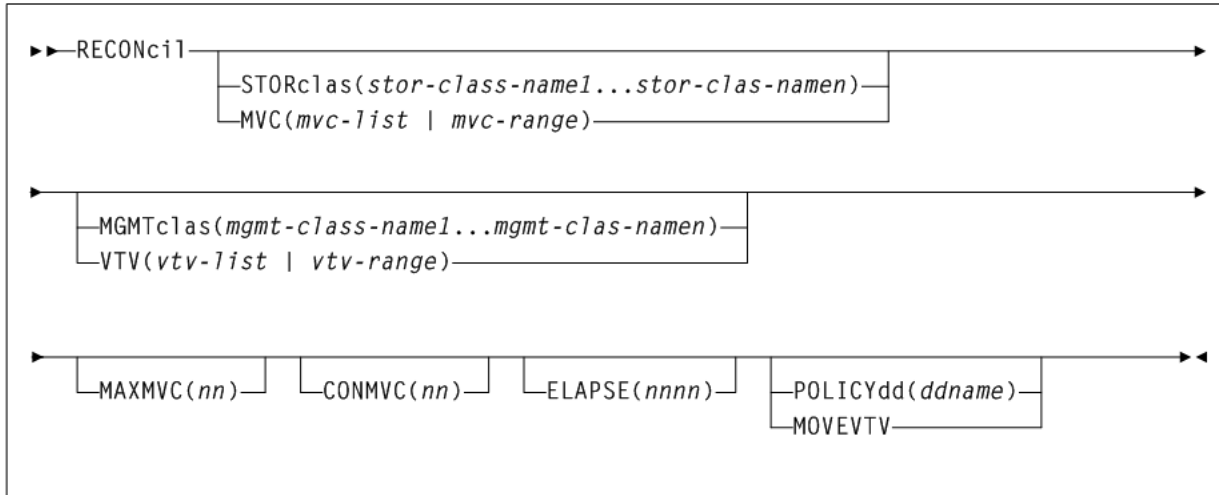
`RECONCIL` has two phases:

- In the first phase, VTCS selects MVC and Storage Classes to be checked. MVCs are checked to ensure they meet the Storage Class definitions and ACS definitions. If inconsistencies are found, then the VTVs on those MVCs are added to the list of VTVs to be recalled and re-migrated to give correct MVC, ACS and Storage Class copies for that VTV. The `RECONcil` parameters `STORCLAS` and `MVC` limit the checking in this phase.
- In the second phase, VTCS selects the VTVs and Management Classes to establish the correct number and location of VTV copies. Any VTVs with inconsistencies are added to the list of VTVs to be recalled and re-migrated. The `RECONcil` parameters `MGMTCLAS` and `VTV` limit the checking in this phase.

## Syntax

Figure 3–134 shows RECONcil command syntax.

Figure 3–134 RECONcil command syntax



## Parameters

The **STORclas** and **MVC** parameters limit the first phase of checking to the specified Storage Classes or MVCs. If **STORclas** or **MVC** are not specified, VTCS checks all MVCs by default.

As shown in Figure 3–134, the **RECONcil** command includes the following parameters:

### **STORclas(stor-class-name1...stor-class-namen)**

optionally, specifies one or more Storage Classes.

*stor-class-name1...stor-class-namen* indicates one or more Storage Classes.

### **MVC(mvc-list or mvc-range)**

a list or range of MVCs. This parameter is mutually exclusive with **STORclas**.

*mvc-list* or *mvc-range* indicate a list or range of MVCs.

### **MGMTclas(mgmt-class-name1...mgmt-class-namen)**

optionally, specifies one or more Management Classes. This parameter is mutually exclusive with **VTVID**.

*mgmt-class-name1...mgmt-class-namen* indicates one or more Management Classes.

### **VTV(vtv-list or vtv-range)**

optionally, specifies a list or range of VTVs. This parameter is mutually exclusive with **MGMTclas**.

*vtv-list* or *vtv-range* indicates a list or range of VTVs.

### **MAXMVC(nn)**

optionally, specifies the maximum number of MVCs that will be processed by a single reconciliation task.

*nn* indicates the number of MVCs. 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)**

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

*nn* indicates the number of MVCs. 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)**

optionally, 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.

*nnnn* indicates the time in minutes. 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.

**PLICYdd(ddname)**

optionally, specifies the DD name of a file containing an alternate MGMTclas statement. This parameter is mutually exclusive with MOVEVTV.

*ddname* indicates the DD name.

**MOVEVTV**

optionally, 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. (no VTVs are moved). This parameter is mutually exclusive with POLICYdd.

**RECONcil Reports**

The following shows examples of RECONcil reports.

**RECONcil Report (MOVEVTV Not Specified)**

The following is an example of a RECONcil report:

**Example 3–58 RECONcil report (MOVEVTV not specified)**

VTV	Size	Comp	<----Creation---->	MGMT	MVC1	MVC2	MVC3	MVC4	Tapeplex
REASON									
Volser	(MB)	%	Date	Time	CLASS				
DX1000	358.4	0	2009Jan18	16:00:00	M5				SECNDARY
EEXPORT									
DX1001	358.4	0	2009Jan18	15:59:58	M5				SECNDARY
EEXPORT									
DX1008	358.4	0	2009Jan18	15:59:43	M5				SECNDARY
EEXPORT									
DX1009	358.4	0	2009Jan18	15:59:41	M5				SECNDARY
EEXPORT									
DX1010	358.4	0	2009Jan18	15:59:40	M5				SECNDARY
EEXPORT									
DX1011	25.6	0	2009Jan18	15:59:38	M5				SECNDARY
EEXPORT									
DX1020	358.4	0	2009Jan18	15:59:19	M5				SECNDARY
EEXPORT									
DX1021	358.4	0	2009Jan18	15:59:18	M5				SECNDARY
EEXPORT									

RECONcil

DX1024	358.4	0	2009Jan18	15:59:12	M5		SECNDARY
EEXPORT							
DX1025	358.4	0	2009Jan18	15:59:10	M5		SECNDARY
EEXPORT							
DX1028	358.4	0	2009Jan18	15:59:04	M6		SECNDARY
EEXPORT							
DX1029	358.4	0	2009Jan18	15:59:01	M6		SECNDARY
EEXPORT							
DX1032	358.4	0	2009Jan18	15:58:56	M6		SECNDARY
EEXPORT							
DX1033	358.4	0	2009Jan18	15:58:54	M6		SECNDARY
EEXPORT							
DX1036	358.4	0	2009Jan18	15:58:48	M5		SECNDARY
EEXPORT							
DX1037	358.4	0	2009Jan18	15:58:46	M6		SECNDARY
EEXPORT							
DX1050	51.2	0	2009Jan18	15:58:20	M5		SECNDARY
EEXPORT							
DX1051	51.2	0	2009Jan18	15:58:17	M5		SECNDARY
EEXPORT							
DX1052	51.2	0	2009Jan18	15:58:15	M5		SECNDARY
EEXPORT							
DX1053	51.2	0	2009Jan18	15:58:14	M5		SECNDARY
EEXPORT							
DX1054	51.2	0	2009Jan18	15:58:12	M5		SECNDARY
EEXPORT							
DX1055	51.2	0	2009Jan18	15:58:10	M6		SECNDARY
EEXPORT							
DX1056	51.2	0	2009Jan18	15:58:08	M6		SECNDARY
EEXPORT							
DX1057	51.2	0	2009Jan18	15:58:06	M6		SECNDARY
EEXPORT							
DX1058	51.2	0	2009Jan18	15:58:04	M6		SECNDARY
EEXPORT							
DX1059	51.2	0	2009Jan18	15:58:02	M6		SECNDARY
EEXPORT							
Total VTVs =		26	Total Size =		5913MB		

SLUADMIN (7.3.0) StorageTek Enterprise Library Software Utility  
 PAGE 00003  
 TIME 08:06:35 Reconcile Tapeplex Report DATE  
 2015-01-19

Tapeplex	Candidate	VTVs	Total VTV size (MB)								
SECNDARY		26	5913.60								
	DX1000	DX1001	DX1008	DX1009	DX1010	DX1011	DX1020	DX1021	DX1024	DX1025	DX1028
DX1029	DX1032	DX1033									
	DX1036	DX1037	DX1050	DX1051	DX1052	DX1053	DX1054	DX1055	DX1056	DX1057	DX1058
DX1059											
Total tapeplexes =		1	Total Size =		5913MB						
SLS0155I Condition code for utility function is 0											

**Fields**

The RECONcil report includes the following 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 Oracle 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 100 MB VTV compresses to 40 MB 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) affected as a result of the Reconcil.

**Tapeplex**

The TapePlex name.

**Reason**

The reason for the VTV selection: media type, media location, migration (space release), immediate migration, or Storage Class.

The RECONcil Tapeplex Report section displays the number of candidate VTVs and Total VTV size (MB) for each TapePlex.

**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.

**RECONcil MOVEVTV Report**

The following is an example of a RECONcil report produced by the RECONcil command issued with the MOVEVTV parameter:

**Example 3–59 RECONcil MOVEVTV report**

```
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
```

**RECOVer****Interfaces:**

- Console or PARMLIB
- UUI Support: No

**Subsystem Requirements:**

Active HSC at FULL service level

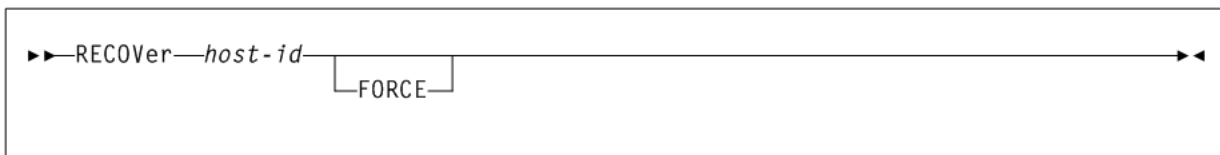
## Description

The `RECOV` command recovers the resources owned by a host that becomes inoperable. Cross-host recovery frees library resources that are owned by the failing host, and transfers that ownership to the recovering host. These resources include CAPs, which can be activated by only one host at a time, and volumes, which are assigned to a particular host while moving or mounting.

## Syntax

Figure 3–135 shows `RECOV` command syntax.

Figure 3–135 *RECOV* command syntax



## Parameters

As shown in Figure 3–135, the `RECOV` command includes the following parameters:

### **host-id**

identifies the host for which to perform recovery (the SMF system identifier for JES2 and JES3).

### **FORCE**

optionally, indicates that recovery is to be performed for the specified host even if the HSC has not detected that the host is inactive. This enables the recovery of a host when the active flag for that host is still on.

---

**Caution:** Use the `FORCE` operand with great caution. Make sure the specified host is inactive before specifying this operand. Forced recovery of an active host requires that the HSC on that host be recycled. All database activity is prohibited on the recovered host, which can cause unexpected abends when tape activity occurs, or when the HSC is recycled on that host.

---

## RELease

### Interfaces:

- Console or `PARMLIB` only
- UUI Support: No

### Subsystem Requirements:

Active HSC at `FULL` service level

## Description

The `RELease` command frees an allocated CAP, making it available for other processes, without requiring a recycle of all HSCs sharing the control data set.

A CAP can be left allocated to a host if the HSC on that host terminated without performing recovery while the CAP was active.

This command can be issued from any connected host. The operator is required to confirm the release operation by responding to a console message.

---



---

**Caution:** Use this command as a last resort. Make sure the CAP is not being used by another active process. Issuing the `RELease` command on an active CAP may cause the process using the CAP to receive errors.

---



---

## Syntax

Figure 3–136 shows `RELease` command syntax.

**Figure 3–136** *RELease command syntax*

```

▶▶RELease cap-id▶▶
  
```

## Parameters

As shown in Figure 3–136, the `RELease` command includes the following parameters:

### **cap-id**

identifies the CAP to be released. The format of a *cap-id* is *AA:LL:CC*, where *AA:LL* indicates the LSMid, and *CC* indicates the CAP. See "CAPid" on page B-1 for a list of valid values.

---



---

**Note:** The format of an LSMid is *AA:LL*, where *AA* indicates the ACSid (decimal 00-99) and *LL* indicates the LSM number (decimal 00-99).

---



---

## REPLaceall

### Interfaces:

- Console or utility
- UI Support: Yes

### Subsystem Requirements:

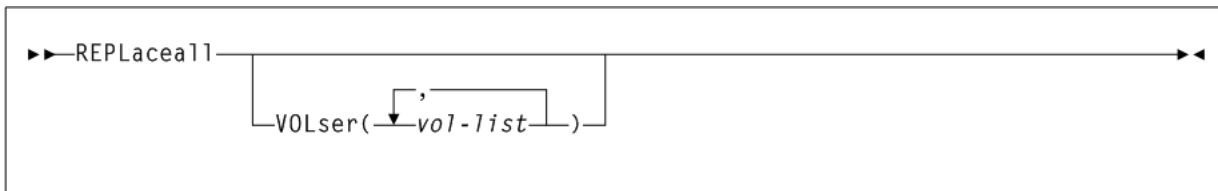
Active HSC at `FULL` service level

## Description

The `REPLaceall` command deletes the scratch list in the CDS and optionally replaces it with a new list.

## Syntax

Figure 3–137 shows `REPLaceall` command syntax.

**Figure 3–137 REPLaceall command syntax**

## Parameters

As shown in [Figure 3–137](#), the REPLaceall command includes the following parameters:

### **VOLser(vol-list)**

optionally, specifies the list of volume serial numbers to be added, deleted, or replaced in the scratch list(s).

*vol-list* indicates the volume serial numbers; this can be a single volume, a list of volume serial numbers, ranges of volume serial numbers, or combinations of lists with ranges delimited by commas. The entire list must be enclosed in parentheses.

## REStore

### Interfaces:

- SLUADMIN utility only
- UUI Support: No

### Subsystem Requirements:

HSC must be down (inactive)

## Description

The REStore command restores, or re-creates the library Control Data Set (CDS) from a previous CDS backup copy. If multiple CDS DD statements are specified, then multiple CDS copies are created in the REStore process.

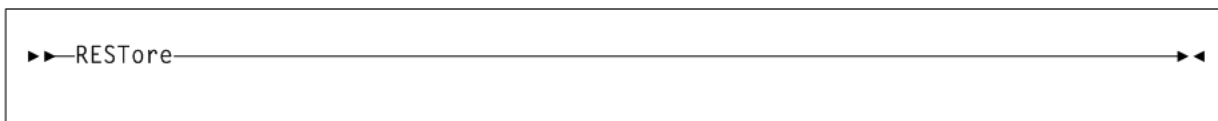
---

**Caution:** It is critical that you restore all data sets referenced by the HSC (primary, secondary, standby). Failing to restore all data sets can cause CDS mismatches. Refer to Oracle’s ELS publication *Managing HSC and VTCS* for detailed information about when to use this utility and how it functions.

---

## Syntax

[Figure 3–138](#) shows REStore command syntax.

**Figure 3–138 REStore command syntax**

## Parameters

None.

## Additional JCL Requirements

In addition to the required JCL definition statements described in "Invoking [SLUADMIN](#)" on page 5-1, the following definition statements apply to the RESTore JCL:

### **SLSCNTL**

specifies the primary CDS.

### **SLSCNTL2**

optionally, specifies the secondary CDS.

If a secondary CDS exists, include this statement so that if a switch occurs and the secondary CDS becomes active, the CDSs can be reordered to maintain database integrity.

### **SLSSTBY**

optionally, specifies the standby CDS.

If a standby CDS exists, include this statement so that if a switch occurs and the standby CDS becomes active, the CDSs can be reordered to maintain database integrity.

### **SLSBKUP**

the backup data set.

## SCRAtch

### Interfaces:

- Console or utility
- UUI Support: Yes

### Subsystem Requirements:

Active HSC at FULL service level

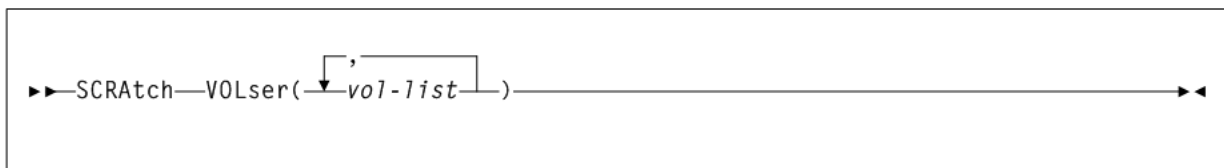
## Description

The SCRAtch command scratches one or more volumes.

## Syntax

Figure 3–139 shows SCRAtch command syntax.

**Figure 3–139** SCRAtch command syntax



## Parameters

As shown in [Figure 3–139](#), the `SCRATCH` command includes the following parameters:

### **VOLser(vol-list)**

specifies the list of volume serial numbers to be added, deleted, or replaced in the scratch list(s).

*vol-list* indicates the volume serial numbers; this can be a single volume, a list of volume serial numbers, ranges of volume serial numbers, or combinations of lists with ranges delimited by commas. You can specify a maximum of 100 volume serial numbers. The entire list must be enclosed in parentheses.

## SCREdist

### **Interfaces:**

- SLUADMIN utility only
- UUI Support: No

### **Subsystem Requirements:**

Active HSC at FULL service level

## Description

The `SCREdist` command redistributes, or balances, the number of scratch volumes across the library.

`SCREdist` permits the redistribution of scratch volumes either:

- among LSMs implied by the specification of an ACS, or
- among the LSMs explicitly identified through an optional parameter.

`SCREdist` enables you to select one media type and recording technique. If specified, scratch volume redistribution will be based on the `MEDIA` and `RECTech` settings.

If neither `MEDIA` nor `RECTech` is specified, volumes are redistributed without regard to media type or recording technique.

When only the ACS parameter is specified, the redistribution method transfers cartridges from LSMs with a greater number of scratch volumes to LSMs with a lesser number of scratch volumes. The process is repeated until the number of scratch volumes in each LSM is within the defined scratch tolerance level.

Distribution in an ACS containing a mix of LSMs (4410s, 9310s, 9360s, 9740s) is performed based on a percentage of scratch cartridges in the ACS. That is, the utility causes each LSM to have the same percentage of scratch tapes within the ACS as the LSM has cells within the ACS. For example, in an ACS where a 9310 LSM contains 80 percent of the cells within the ACS, 80 percent of the scratch tapes will reside in the 9310.

---

**Note:** The SL3000 and SL8500 are standalone libraries and cannot be combined with any other LSM type.

---

After Scratch Redistribution is completed, only the specified LSMs contain scratch cartridges, if the LSM parameter was specified.

Concurrent redistributions among different ACSs are accomplished by multiple executions of this utility program.

Redistribution moves non-scratch cartridges to make space for scratch cartridges. The actual number of scratch cartridges moved may vary depending on the number of free cells and the number of scratch cartridges available.

---



---

**Note:** Cartridges cannot be transferred to a frozen panel.

---

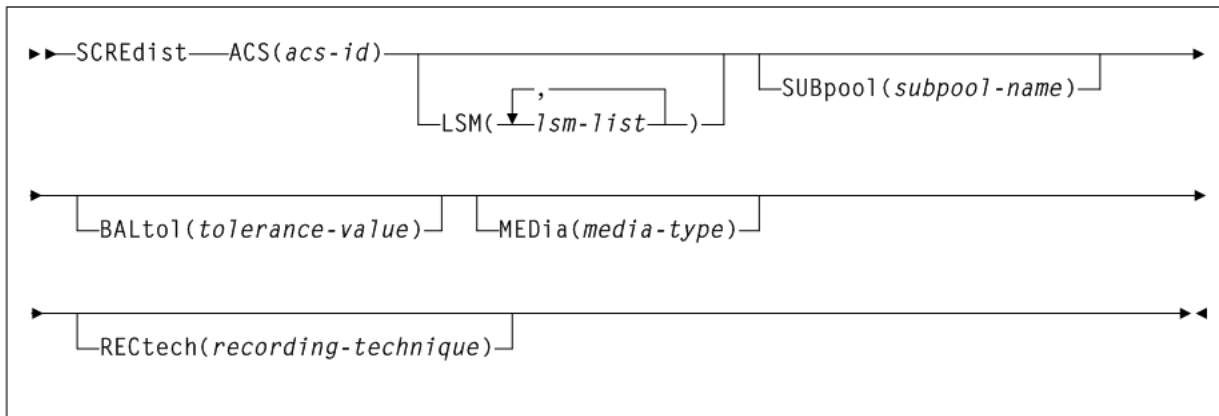


---

## Syntax

Figure 3–140 shows SCREdist command syntax.

**Figure 3–140 SCREdist command syntax**



## Parameters

As shown in Figure 3–140, the SCREdist command includes the following parameters:

### **ACS(acs-id)**

specifies one ACS in the library in which the scratch volumes are to be redistributed.

*acs-id* indicates the ACS. The one or two digit decimal ACS identifier.

### **LSM(lsm-list)**

optionally, specifies that only certain LSMs within an ACS are to contain scratch cartridges (if not specified, scratch volumes are redistributed across all LSMs in the ACS).

*lsm-list* indicates a single LSM or a list of LSMids. An LSMid (*lsm-id*) is made up of the ACSid (decimal 00-99) and the LSM number (decimal 00-99) separated by a colon (:). An LSM range is not allowed.

The element(s) contained in an *lsm-list* must be enclosed in parentheses; multiple elements must be separated by blanks or commas.

### **SUBpool(subpool-name)**

optionally, specifies the subpool name from which scratch volumes are to be redistributed.

*subpool-name* indicates the subpool.



**BALtol(tolerance-value)**

optionally, specifies a balance tolerance value. Scratch volumes are distributed based on this specified value.

*tolerance-value* indicates a percent value that identifies when cartridge redistribution ends. Valid values are 1 through 999, where the values indicate a percentage from .1 to 99.9 (i.e., 1 signifies .1 percent; 999 equals 99.9 percent). The initial value for the HSC is 1. If *tolerance-value* is not specified in this utility, HSC uses the initial value for scratch redistribution.

The utility finishes redistributing scratch cartridges when the percentage of scratch tapes in all specified LSMs is within *tolerance-value* / 2 percentage points of each LSM's percentage of cells in the ACS.

For example, assume an ACS has one 4410 LSM with 5,000 cells, one 9360 (WolfCreek) LSM with 1,000 cells, and 600 total scratch tapes in the ACS. Scratch Redistribution attempts to put 500 scratches in the 4410 and 100 scratches in the 9360.

Entering `BALtol(100)` specifies a setting of 10 percent, which means that the utility ends when all LSMs are within  $\pm 5$  percent of the expected number of scratches for each LSM (500 for the 4410, 100 for the 9360).

The expected number can be determined as follows:

$$\text{expnumlsm} = \text{totscracs} \times (\text{totcellslsm} / \text{totcellsacs})$$

where:

- *expnumlsm* indicates the expected number of scratches for each LSM.
- *totscracs* indicates the total number of scratches in the ACS.
- *totcellslsm* indicates the total number of cells in the LSM.
- *totcellsacs* indicates the total number of cells in the ACS.

In the previous example, the 4410 should have a range 470-530 scratches and the 9360 a range of 70-130. To determine the `BALtol` range:

$$\text{BALtol range} = \text{expnumlsm} + (\text{baltol}/1000 \times \text{totscracs}) \setminus 2$$

where:

- `BALtol range` indicates the range of balance tolerance value.
- *expnumlsm* indicates the expected number of scratches for each LSM (see above).
- *baltol* indicates the user-supplied percent value (*tolerance-value*) that specifies when scratch cartridge redistribution ends.
- *totscracs* indicates the total number of scratches in the ACS.

**MEDia(media-type)**

optionally, specifies the type of cartridge to redistribute across the ACS.

*media-type* indicates the media type

See [Appendix A, "MEDia, RECtech, and MODEL Values"](#) for a list of valid *media-type* values.

**RECtech(recording-technique)**

optionally, specifies the method used to record data tracks on the tape surface.

---



---

**Note:** If RECtech is not specified, the next scratch cartridge is selected depending on the MEDIA type that has been specified.

---



---

*recording-technique* indicates the recording technique. See [Appendix A, "MEDIA, RECtech, and MODEL Values"](#) for valid *recording-technique* values.

## SCRPT

### Interfaces:

- SLUADMIN utility only
- UUI Support: No

### Subsystem Requirements:

- Active HSC not required
- SMC must be active and communicating with at least one VLE with the deduplication feature enabled to generate data in the report output. The output report must run from an authorized library.

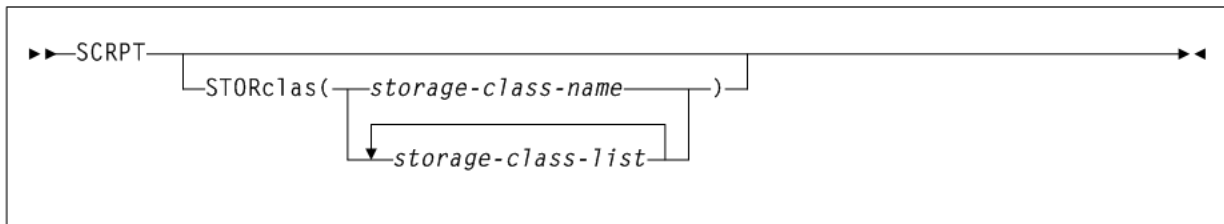
## Description

The SCRPT command reports on VLE usage by Storage Class.

## Syntax

Figure 3–141 shows SCRPT command syntax.

**Figure 3–141 SCRPT command syntax**



## Parameters

As shown in [Figure 3–141](#), the SCRPT command includes the following parameters:

### **STORclas(storage-class-name or storage-class-list)**

specifies the names of one or more Storage Classes. If not specified, the report applies to all Storage Classes.

*storage-class-name* or *storage-class-list* indicate the Storage Class name or list of Storage Classes. Storage Class names must be 1 to 8 alphanumeric characters beginning with an alpha character and must follow SMS naming conventions.

## SCRPT Report

The following is an example of the SCRPT report:

**Example 3–60 SCRPT report**

Storage Class	STORMNGR	Node	Total MVCs	Capacity (GB)	Used (GB)	Compressed (GB)	Uncompressed (GB)	Reduction Ratio
PROD1	VLELIB1	0	4	1000	200	800	3200	16.0:1
		1	3	750	200	400	1600	8.0:1
		2	5	1250	200	400	1600	8.0:1
		3	4	1000	0	0	0	1.0:1
			16	4000	600	1600	6400	10.7:1
Total=			16	4000	600	1600	6400	10.7:1
{All}	VLELIB1	0	4	1000	200	800	3200	16.0:1
		1	3	750	200	400	1600	8.0:1
		2	5	1250	200	400	1600	8.0:1
		3	4	1000	0	0	0	1.0:1
			16	4000	600	1600	6400	10.7:1
Total=			16	4000	600	1600	6400	10.7:1

**Fields**

The SCRPT report includes the following fields:

**Storage Class**

The Storage Class.

**STORMNGR**

The VLE subsystem name.

**NODE**

The server node.

**Total MVCs**

Total count of VMVCs in this grouping.

**Capacity (Gb)**

Total capacity in Gbs of all VMVCs in this grouping.

**Used (Gb)**

Total Gbs of used VMVC space in this grouping.

**Compressed (Gb)**

Total Gbs of compressed data in this grouping, which is equal to the size of the data as the VTSS compresses it (at a 4:1 ratio) and writes it to a VTV.

**Uncompressed (Gb)**

Total Gbs of uncompressed data in this grouping, which equals the original size of the data before the VTSS compresses it and writes it to a VTV.

**Reduction Ratio**

The approximate reduction ratio for the data, which is Uncompressed Gb divided by Used Gb. The Reduction Ratio, therefore, includes both VTSS compression and VLE deduplication. A larger reduction ratio indicates more effective compression and deduplication. For example, the VTSS receives 16 Mb of data, compresses it to 4Mb,

and writes the compressed data to a VTV. VLE subsequently deduplicates the VTV to 2Mb and writes it to a VMVC. Thus, the reduction ratio is 16Mb/2Mb or 8.0:1.

Because the calculation is done using Mb, it is possible to see 0Gb in the Used or Uncompressed fields, yet see a reduction ratio other than 1.0:1.

## SENter

### Interfaces:

- Console or PARMLIB only
- UUI Support: No

### Subsystem Requirements:

Active HSC at FULL service level

## Description

The `SENter` command schedules an enter on a CAP that is currently allocated to an eject operation. The `SENter` command must be issued from the host that has the CAP allocated for ejecting cartridges.

## Syntax

Figure 3–142 shows `SENter` command syntax.

Figure 3–142 *SENter* command syntax



## Parameters

As shown in Figure 3–142, the `SENter` command includes the following parameters:

### **cap-id**

*cap-id* indicates the CAP to use to enter the cartridge. One CAP must be specified and it must be allocated to the host issuing the `SENter` command.

The format of a *cap-id* is *AA:LL:CC*, where *AA:LL* indicates the LSMid, and *CC* indicates the CAP. See "CAPid" on page B-1 for a list of valid values.

---

**Note:** The format of an LSMid is *AA:LL*, where *AA* indicates the ACSid (decimal 00-99) and *LL* indicates the LSM number (decimal 00-99).

---

## SET

The `SET` command enables you to change selected library configuration settings without performing a library reconfiguration. It performs operations directly on the control data set and does not require the HSC to be active.

Issue this command with any of the options listed in the following table. Each option is described individually, and in more detail, on the pages to follow.

**Table 3–13 SET Options**

Option	Description
CLNPRFX	cleaning cartridge prefix
COMPRFX	HSC/VTCS command prefix
DRVHOST	whether to allow all SMC client systems to use device addresses defined in the SMC <code>DRIVEMAP</code> command to influence allocation and request mounts and dismounts.
EJCTPAS	eject password
EJCTSKP	SL8500 Bulk CAP skip global setting
FREEZE	whether to disallow (freeze) or allow (unfreeze) additional cartridges to be stored on a panel or LSM
HOSTID	changes an old host id to a new host id
HSCLEVEL	specifies that the HSC active and release level indicators are to be cleared for the designated host
LOGFILE	defines primary and secondary log files to include logical transactions necessary to recover the CDS.
MAJNAME	specifies that the <code>ENQ/DEQ/RESERVE QNAME</code> is to be set
MIGOPT	changes the following VTCS migration parameters: <ul style="list-style-type: none"> <li>■ maximum and minimum concurrent automatic migration, immediate migration, and migrate-to-threshold tasks</li> <li>■ high and low AMTs</li> </ul>
NEWHOST	specifies that a new host is to be added
RMM	enables or disables the VTCS to <code>DFSMSrmm</code> interface
SCRLABL	specifies that the LMU station addresses are to be set
SLIDRIVS	specifies that the device numbers for the drives in a particular panel are being changed or added
SLISTATN	specifies that the LMU station addresses are to be set.
SMF	specifies that the SMF record type used by the HSC is to be set
TAPEPLEX	specifies that the TapePlex is to be used
TCHNIQE	specifies that the control data set recovery technique is to be set
VAULT	The <code>SET Vault</code> utility is used to define the <code>VAULT</code> information of a server complex.
VAULTVOL	The <code>SET VAULTVOL</code> utility is used to define the <code>VAULT Volume</code> area
VOLPARM	The <code>SET VOLPARM</code> utility defines the volume and pool information of a server complex.
VOLPARM UPDATE	The <code>SET VOLPARM UPDATE</code> utility defines the changes to the volume and pool information for a server complex.
VOLPARM JOIN	The <code>SET VOLPARM JOIN</code> utility merges contiguous volume ranges defined on multiple <code>VOLPARM</code> statements into one new <code>VOLPARM</code> statement.

## SET CLNPRFX

### Interfaces:

- SLUADMIN utility only

- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

**Description**

The `SET CLNPRFX` command sets a cleaning prefix.

---



---

**Note:** HSC must be shut down on all systems before changing the cleaning prefix.

---

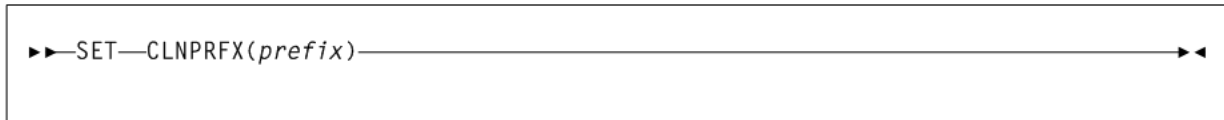


---

**Syntax**

Figure 3–143 shows `SET CLNPRFX` command syntax.

**Figure 3–143** *SET CLNPRFX command syntax*



**Parameters**

As shown in Figure 3–143, the `SET CLNPRFX` command includes the following parameters:

**prefix**

indicates a 3-character prefix for the cleaning cartridges. Valid characters are A-Z, 0-9, \$, #, and @.

---



---

**Note:** HSC must be shut down on all systems before changing the cleaning prefix.

---



---

**SET CLNPRFX Procedure**

To change a cleaning cartridge prefix:

1. Eject all cleaning cartridges from all ACSs. The HSC records information about cleaning cartridges in the CDS.
2. Terminate the HSC on all hosts.
3. Change the cleaning prefix using `SET CLNPRFX`.
4. Initialize the HSC on any desired hosts.
5. Enter new cleaning cartridges identified with the new prefix into all ACSs.

---



---

**Note:** It is preferable to enter new cleaning cartridges because a cartridge's select count is set to zero when it is ejected and reentered. The select count tracks the number of times a cleaning cartridge has been used.

---



---

## SET COMPRFX

### Interfaces:

- SLUADMIN utility only
- UI Support: No

### Subsystem Requirements:

Active HSC not required

### Description

The SET COMPRFX command sets a HSC/VTCS command prefix.

### Syntax

Figure 3–144 shows SET COMPRFX command syntax.

**Figure 3–144 SET COMPRFX command syntax**

```

▶▶—SET—COMPRFX(cmdhex)—▶▶
  
```

### Parameters

As shown in Figure 3–144, the SET COMPRFX command includes the following parameters:

#### cmdhex

indicates the 2-character hexadecimal code of the command prefix. Do not choose a value that conflicts with values assigned to other subsystems. The characters associated with each code are shown in Table 3–14.

The command prefix is not valid until the HSC or VTCS system is recycled.

Make sure that the prefix character used does not conflict with any of the following:

- another subsystem's command prefix character (such as \$ for JES2 or \* for JES3, or ; the delimiter for TSO).
- the command delimiter for MVS, specified as the value for the CMDDELIM keyword in SYS1.PARMLIB(CONSOLxx) and described in the *IBM z/OS Installation and Tuning Guide*.
- a JES line editing character as specified in the JES installation statements or default. For JES2 the initialization statement is CONDEF.
- For JES3 the installation statement is CONSTD. Descriptions of these statements and default values can be found in the appropriate *IBM z/OS JES3 Installation and Tuning Guide*.

If you are running the HSC on a VM host that shares the CDS with a HSC running on MVS, make sure that your command prefix character does not conflict with any of the VM facilities, such as the CP line editing symbol. The following table maps command prefix codes to characters:

**Table 3–14 Mapping of Command Prefix Codes to Characters**

Hex	Character	Description
40	null	blank
4A		cent
4B	.	period
4C	<	less than
4D	(	left parenthesis
4E	+	plus
4F		vertical bar
50	&	ampersand
5A	!	exclamation point
5B	\$	dollar sign
5C	*	asterisk
5D	)	right parenthesis
5E	;	semicolon
5F		not symbol
60	-	minus
61	/	slash
6B	,	comma
6C	%	percent
6D	_	underscore
6E	>	greater than
6F	?	question mark
7A	:	colon
7B	#	crosshatch
7C	@	at sign
7E	=	equals sign
7F	"	double quote

If you specify a null command prefix (hex 40), you must use the MVS `MODIFY` command to perform any HSC operator command.

For example: `F SLS0 MOUNT EDU050,B30`

where:

- F is an abbreviation for the MVS `MODIFY` command.
- SLS0 is an HSC subsystem.
- MOUNT is an HSC operator command.
- EDU050 is a VOLSER ID.
- B30 is a designated tape drive.



## SET DRVHOST

### Interfaces:

- SLUADMIN utility only
- UUI Support: No

### Subsystem Requirements:

Active HSC not required

### Description

The SET DRVHOST command directs SMC client systems to use drive addresses defined in the SMC DRIVemap command to influence allocation and request mounts and dismounts.

---

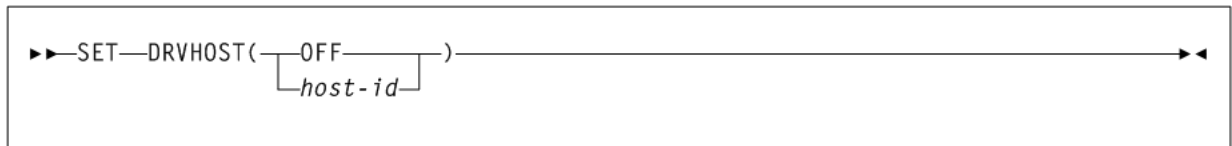
**Note:** See "DRIVemap" on page 2-24 for more information about the SMC DRIVemap command.

---

### Syntax

Figure 3–145 shows SET DRVHOST command syntax.

**Figure 3–145 SET DRVHOST command syntax**



### Parameters

As shown in Figure 3–145, the SET DRVHOST command includes the following parameters:

#### OFF

indicates that each HSC reports device addresses to SMC clients based on drives defined for that host in the HSC SLIBRARY LIBGEN macro.

#### host-id

a host name defined in the CDS.

## SET EJCTPAS

### Interfaces:

- SLUADMIN utility only
- UUI Support: No

### Subsystem Requirements:

Active HSC not required

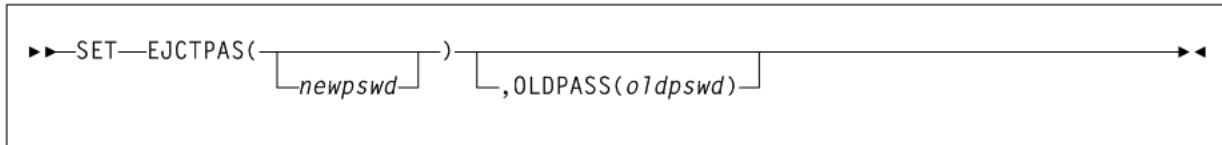
### Description

The SET EJCTPAS command sets a HSC eject password. This password is not valid until the HSC system is recycled.

## Syntax

Figure 3–146 shows SET EJCTPAS command syntax.

**Figure 3–146 SET EJCTPAS command syntax**



## Parameters

As shown in Figure 3–146, the SET EJCTPAS command includes the following parameters:

### newpswd

a 1- to 8-character eject password.

The new password must be alphanumeric: A-Z (capital letters only) and 0-9. If *newpswd* is omitted, eject password checking is disabled.

---

**Note:** *newpswd* is not be displayed on SLSPRINT.

---

### ,OLDPASS(oldpswd)

specifies that the old eject password is being specified. The old password must be specified to change or delete a password. To delete an old password, omit it from the syntax, that is, OLDPASS( ). If there is not currently an eject password, this parameter can be omitted.

*oldpswd* indicates the 1-to 8-character old eject password.

---

**Note:** *oldpswd* is not displayed on SLSPRINT.

---

## SET EJCTSKP

### Interfaces:

- SLUADMIN utility only
- UUI Support: No

### Subsystem Requirements:

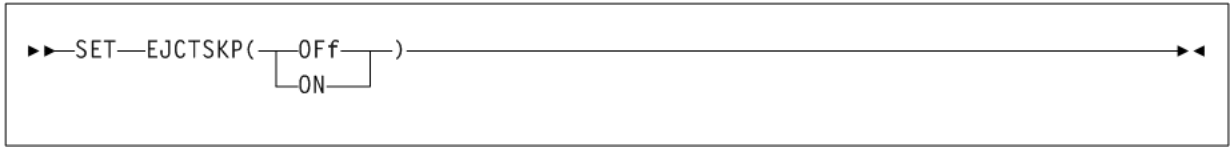
Active HSC not required

### Description

The SET EJCTSKP command enables or disables the SL8500 Bulk CAP global setting that requests the EJECT process skip the last slot of the magazines on rails 1-3 and the first slot of rail 4. These empty cells provide the ability to more easily remove the magazine from the CAP.

## Syntax

Figure 3–147 shows SET EJCTSKP command syntax.

**Figure 3–147 SET EJCTSKP command syntax****Parameters**

As shown in [Figure 3–147](#), the SET EJCTSKP command includes the following parameters:

**OFF**

Disables this setting for all Bulk CAPs.

**ON**

Enables this setting for all Bulk CAPs.

---

**Note:** Refer to Oracle’s ELS publication *Configuring HSC and VTCS* for information about how to change CAP types from SL8500 standard rotational CAPs to Bulk CAPs, or vice versa.

---

**SET FREEZE****Interfaces:**

- SLUADMIN utility only
- UUI Support: No

**Subsystem Requirements:**

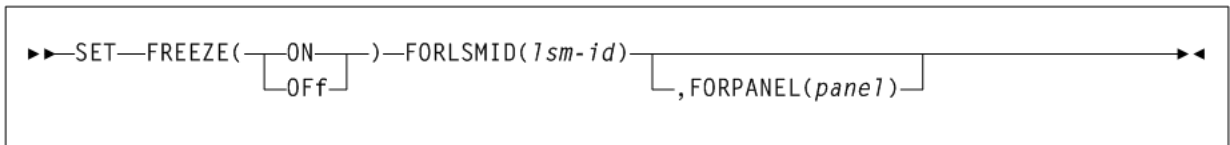
Active HSC not required

**Description**

The SET FREEZE command specifies whether to disallow (freeze) or allow (unfreeze) additional cartridges to be stored on a panel or LSM.

**Syntax**

[Figure 3–148](#) shows SET FREEZE command syntax.

**Figure 3–148 SET FREEZE command syntax****Parameters**

As shown in [Figure 3–148](#), the SET FREEZE command includes the following parameters:

**ON**

specifies to freeze a panel or LSM, which prevents additional cartridges from being moved to it. This restriction includes allocating new cartridge locations on a panel or LSM as a result of the following:

- a MOVE command, utility, or PGMI request
- cartridge entry into the ACS
- float, scratch dismount, or scratch redistribution processing.

If a cartridge on a frozen panel is selected (e.g., through a mount request), it may be returned to its home cell on the frozen panel after fulfilling the request.

Cartridges already located on a frozen panel or LSM must be deliberately moved off using the MOVE command, utility, or PGMI request, or cartridges can be ejected by running the EJECT command or EJECT utility.

**OFF**

specifies to unfreeze a panel or LSM, which enables additional cartridges to be moved to it.

**FORLSMID(*lsm-id*)**

specifies that the operation is being restricted to a specific panel or LSM.

*lsm-id* indicates the LSMid (*AA:LL*) of the LSM being frozen or unfrozen. An LSMid (*lsm-id*) is made up of the ACSid (decimal 00-99) and the LSM number (decimal 00-99) separated by a colon (:).

**FORPANEL(*panel*)**

optionally, specifies that the operation is being restricted to a specific panel.

*panel* indicates the 2-digit ID of the panel to be frozen or unfrozen. The specified panel must be an existing drive panel in the LSM. See "Panels" on page B-2 for a list of valid values.

---

---

**Note:**

- If frozen panels already exist in an LSM, and then you freeze the entire LSM, all panels will be unfrozen when you unfreeze the LSM.

If you want some panels to remain frozen after unfreezing the entire LSM, you can add SET FREEZE ON FORLSMID FORPANEL statements to the same SLUADMIN JOBSTEP.

- When an LSM is frozen, no new cell allocations occur. However, any cartridge from the frozen LSM that is currently mounted on a drive will return to its home cell.
- Frozen or unfrozen panels and LSMs are recognized immediately by all active HSCs. It is not necessary to stop and reinitialize active HSCs to detect changes made by SET FREEZE.

On a frozen panel, if a panel type is changed by running the MERGEcds or Reconfiguration utility, the new panel is not frozen. If the panel type did not change, the panel remains frozen after a merge or reconfiguration.

---

---

## SET HOSTID

### Interfaces:

- SLUADMIN utility only
- UI Support: No

### Subsystem Requirements:

Active HSC not required

### Description

The SET HOSTID command renames a host ID.

---

**Note:** If other hosts are active at the time of the host name change, some messages on those active hosts may display the old host name. After the HSC is restarted on those hosts, the new host name appears in the messages.

---

### Syntax

Figure 3–149 shows SET HOSTID command syntax.

**Figure 3–149 SET HOSTID command syntax**

```
▶▶—SET—HOSTID(newhost),FORHOST(oldhost)————▶▶
```

### Parameters

As shown in Figure 3–149, the SET HOSTID command includes the following parameters:

#### HOSTID(*newhost*)

specifies that the old host ID is being changed to a new name.

*newhost* indicates a 1-8 character host ID, which represents the SMF system ID for both JES2 and JES3. *newhost* cannot already be defined in the CDS.

#### FORHOST(*oldhost*)

specifies the old host ID.

*oldhost* indicates a 1-to 8-character host ID. This host must not be marked active in the CDS. A HSC can be down but still marked active. SET HOSTID would fail to run. To be marked inactive, the HSC, in the host that is down, must be brought up and then shut down. Also, another host could perform cross-host recovery on the HSC that is down, but marked active. Either of these change the status in the CDS of the down HSC from active to inactive. See "SET HSCLEVEL" on page 3-324 for information about resetting operating flags.

---

**Note:** You must delimit the HOSTID (*newhost* or *oldhost*) with quotes when this ID is in lower case.

---

## SET HSCLEVEL

### Interfaces:

- SLUADMIN utility only
- UI Support: No

### Subsystem Requirements:

Active HSC not required

### Description

The `SET HSCLEVEL` command clears HSC active and release level indicators for a designated host.

---

**Note:** If used, `HSCLEVEL` should be performed only when the host designated in `host-id` is inactive.

---

These indicators remain set after an abrupt termination of the HSC has occurred. For example, a cancellation of the HSC or a crash of the operating system.

### Syntax

Figure 3–150 shows `SET HSCLEVEL` command syntax.

**Figure 3–150** SET HSCLEVEL command syntax

```
▶▶—SET—HSCLEVEL(OFF), FORHOST(host-id)————▶▶
```

### Parameters

As shown in Figure 3–150, the `SET HSCLEVEL` command includes the following parameters:

#### OFF

specifies that the HSC active and HSC release level indicators for the designated host are to be cleared.

#### FORHOST(host-id)

specifies that the active indicators are to be cleared for the designated host.

`host-id` indicates the 1- to 8-character host ID of the host to which the operation is restricted.

---

**Note:** Oracle recommends the following alternatives to executing `SET HSCLEVEL` because they reset the HSC indicators and recover resources owned by the failing host:

- Restart the HSC on the host.
  - Enter the `RECOVER host-id FORCE` operator command.
-

## SET LOGFILE

### Interfaces:

- SLUADMIN utility only
- UUI Support: No

### Subsystem Requirements:

Active HSC not required

### Description

The `SET LOGFILE` command is part of the HSC Transaction Logging Service, introduced in HSC 7.0.

This command defines primary and secondary log files used by the HSC transaction logging service to record VTCS logical transactions necessary to recover the CDS.

If it becomes necessary to restore a CDS, the information in these log files can be used to reduce the amount of time required to restore the VTCS data in the CDS back to the proper state.

---



---

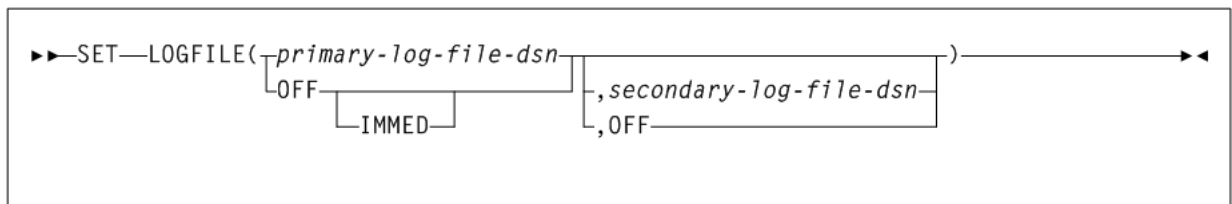
### Note:

- Log files must be pre-formatted using the `FMTLOG` utility. See "[FMTLOG](#)" on page 3-192 for more information.
  - If logging is not currently active, log file names specified in the `SET LOGFILE` command are activated immediately.
  - If logging is currently active, you must run the `OFFload LOGFILE` utility using the `OFFload LOGFILE` command to change the log files. See "[OFFload](#)" on page 3-286 for more information.
- 
- 

### Syntax

Figure 3-151 shows `SET LOGFILE` command syntax.

**Figure 3-151** *SET LOGFILE* command syntax



### Parameters

As shown in Figure 3-151, the `SET LOGFILE` command includes the following parameters:

#### primary-log-file-dsn

the data set name for the primary log file. This parameter is required.

**OFF**

Specify *OFF* in place of the *primary-log-file-dsn* to indicate that logging should be terminated. Logging will continue on the HSC subsystem until you run *OFFload LOG*, unless you also specify the *IMMED* parameter.

**IMMED**

specifies that logging should be immediately stopped on all active HSC subsystems. When *IMMED* is specified, any previous log data cannot be offloaded.

**secondary-log-file-dsn**

optionally, the data set name for the secondary log file. This parameter is required if a secondary log file was previously defined.

When changing only the secondary log file data set name, you must re-specify the existing primary log file data set name in the primary log file data set name position.

**OFF**

Specify *OFF* in place of the *secondary-log-file-dsn* to change from duplex to simplex logging.

**SET MAJNAME****Interfaces:**

- SLUADMIN utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required. HSC must be shut down on all systems before changing the *QNAME*.

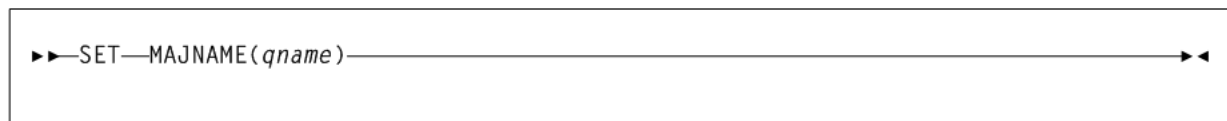
**Description**

The *SET MAJNAME* command sets the *ENQ/DEQ/RESERVE QNAME*.

**Syntax**

Figure 3–152 shows *SET MAJNAME* command syntax.

**Figure 3–152 SET MAJNAME command syntax**

**Parameters**

As shown in Figure 3–152, the *SET MAJNAME* command includes the following parameters:

**qname**

a 1- to 8-character *QNAME*. The *QNAME* is blank padded on the right. The name should conform to the requirements for a *QNAME*. If any hosts are MVS hosts, to prevent conflicts with the operating system, the *QNAME* should not start with *SYSA* through *SYSZ*. Since the HSC is authorized, the *SYSA* through *SYSZ* names would be allowed, but deadlock may result.



---



---

**Note:** HSC must be shut down on all systems before changing the QNAME.

---



---

## SET MIGOPT

### Interfaces:

- Console or utility
- UII Support: Yes

### Subsystem Requirements:

Active HSC/VTCS

### Description

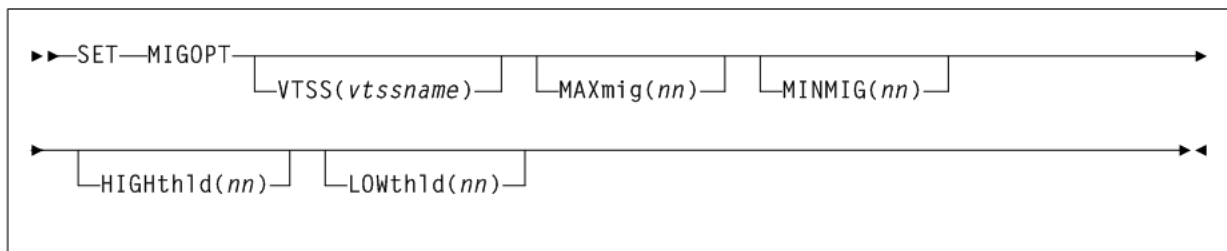
The SET MIGOPT command changes the following migration parameters:

- Maximum and minimum concurrent automatic migration, immediate migration, and migrate-to-threshold tasks.
- High and low AMTs.

### Syntax

Figure 3–153 shows SET MIGOPT command syntax.

**Figure 3–153 SET MIGOPT command syntax**



### Parameters

As shown in Figure 3–153, the SET MIGOPT command includes the following parameters:

#### VTSS(vtssname)

optionally, specifies the VTSS whose migration parameters you want to change. If you do not specify a VTSS, the changes affect all VTSSs.

*vtssname* indicates the VTSS identifier.

#### MAXmig(nn)

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

*nn* indicates the number of tasks. Valid values are 1 to the number of RTDs attached to the VTSS. There is no default; if you do not specify a value, the current value is unchanged.

**MINMIG(nn)**

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

*nn* indicates the number of 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(nn)**

optionally, specifies the new high AMT.

*nn* indicates 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(nn)**

optionally, specifies the new low AMT.

*nn* indicates the new low AMT as a percent of VTSS space. Valid values are 5 to 95 and must be less than the `HIGHthld` value.

**SET NEWHOST****Interfaces:**

- `SLUADMIN` utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

**Description**

The `SET NEWHOST` command defines a new host.

**Syntax**

Figure 3–154 shows `SET NEWHOST` command syntax.

**Figure 3–154** *SET NEWHOST* command syntax

```
▶▶—SET—NEWHOST(newhost), LIKEHOST(model-host)—————▶▶
```

**Parameters**

As shown in Figure 3–154, the `SET NEWHOST` command includes the following parameters:

**newhost**

a 1-8 character host ID, which represents the SMF system ID for both JES2 and JES3.

- The new host to be added cannot exceed the maximum limitation of 16 hosts.
- *newhost* cannot already be defined in the CDS.
- You must delimit the `HOSTID` with quotes when *newhost* is in lower case.

**LIKEHOST(model-host)**

specifies that an existing configuration, as currently defined in the `LIBGEN`, is to be used for the new host. Settings used include the following:

- `SLLIBRY NNLBDRV` (nonlibrary drive esoteric)
- `SLIACS ACSDRV` (ACS drive esoteric)
- `SLISTATN ADDRESS` (ACS 3270 station addresses)
- `SLIDRIVS ADDRESS` (drive addresses)

*model-host* indicates a 1- to 8-character host ID.

**SET RMM****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

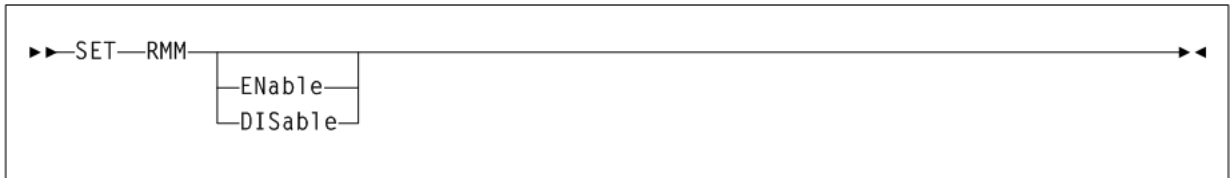
**Description**

The `SET RMM` command enables or disables the VTCS to `DFSMSrmm` interface.

**Syntax**

Figure 3–155 shows `SET` command syntax.

**Figure 3–155** *SET RMM command syntax*

**Parameters**

As shown in Figure 3–155, the `SET RMM` command includes the following 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.

---

---

**Note:**

- 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, Oracle recommends adding the `VTCS SET RMM ENA` command to the `SLSSYSxx PARMLIB` member read at HSC initialization.
  - Enabling this interface can cause delays in scratch mount processing.
- 
- 

## SET SCRLABL

**Interfaces:**

- `SLUADMIN` utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

**Description**

The `SET SCRLABL` command specifies the scratch label type.

To add to the scratch pool, the `REPLaceall` function is run before the `Scratch Update` utility. Otherwise, the `Scratch Update` utility should be run to update the scratch pools after using this function.

**Syntax**

Figure 3–156 shows `SET SCRLABL` command syntax.

**Figure 3–156** *SET SCRLABL* command syntax



**Parameters**

As shown in Figure 3–156, the `SET SCRLABL` command includes the following parameters:

**SL**

specifies that nonspecific requests for standard-labeled tapes are to be automated.

**AL**

specifies that nonspecific requests for ANSI-labeled tapes are to be automated.

**NL**

specifies that nonspecific requests for nonlabeled tapes are to be automated.

**NSL**

specifies that nonspecific requests for nonstandard labeled tapes are to be automated.

**SET SLIDRIVS****Interfaces:**

- SLUADMIN utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

**Description**

The SET SLIDRIVS command adds or changes device numbers for drives in a particular panel.

- For 9310 and 9740 libraries, these changes take effect when you recycle the affected HSC system(s). See ["Running SET SLIDRIVS With the HSC Active"](#) on page 3-333 for more information.
- For SL3000 and SL8500 libraries, these changes take effect when you issue a MODIFY CONFIG UPD ACS(XX) for the ACS that the drives were added to. NCO adds the drives to the configuration. There is no need to recycle the HSC.

---

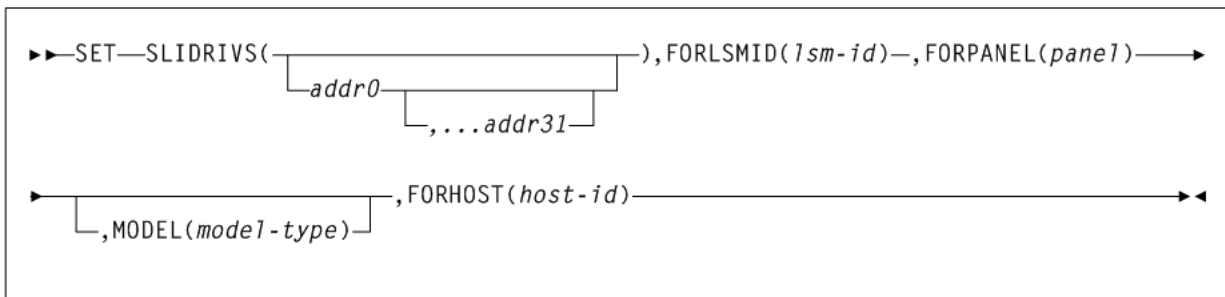
**Caution:** For 9310 and 9740 libraries, Oracle recommends you bring the HSC down on all hosts before specifying this parameter, and recycle the HSC after every SET SLIDRIVS operation. [Table 3-15](#) describes some instances where the HSC can remain active, however, unpredictable results can occur.

---

**Syntax**

[Figure 3-157](#) shows SET SLIDRIVS command syntax.

**Figure 3-157 SET SLIDRIVS command syntax**



## Parameters

As shown in [Figure 3–157](#), the `SET SLIDRIVS` command includes the following parameters:

### **addr0,...addr31**

the device numbers for the panel. The numbers are from the top of the panel down. If a device number is omitted, this means that the host cannot access a drive in that particular position in the panel, or the drive does not exist. Blanks as well as commas may be used to separate the drive specifications.

The total number of drives specified (including comma placeholders) must be:

- 4, 10, or 20 for a 9310
- 16 for an SL8500
- 8, 16, 24 or 32 for an SL3000
- You cannot specify 20-drive panels on 9740 (TimberWolf) LSMs or SL8500 libraries.
- 10-drive and 20-drive panels cannot be specified on 4410 LSMs.

Before this utility can be executed, all cells in the panel must be empty if the user is moving to or from a 20-drive panel configuration.

The user can change a drive panel between a 4- or 10-drive panel (normal) and a 20-drive panel (wide) without running a reconfiguration. Affected panels must be empty and no cells in the panels can be allocated to cartridges.

To make sure that drive panels being changed from normal to wide configurations (or vice versa) remain empty, freeze them with the `SET FREEZE` utility. Then, move all cartridges to other panels or LSMs.

If a panel type is changed by `SET SLIDRIVS`, the new panel will not be frozen. Frozen panels whose panel type did not change remain frozen after `SET SLIDRIVS`.

HSC does not allow duplicate addresses for drives. If it becomes necessary to exchange the drive addresses on one panel with the drive addresses on another panel, the addresses on one of the panels must first be changed to temporary addresses that are not currently defined. For example:

```
LSM00, PANEL10, ADDRESSES-400,401,402,403
LSM01, PANEL11, ADDRESSES-404,405,406,407
```

If the 400-403 addresses are to be moved to LSM01 and 404-407 are to be moved to LSM0, the `SET` utility must first be run to change the LSM0 addresses (400-403) to 900-903 (or some other addresses that are not currently defined). The `SET` utility is then run to change the LSM01 addresses to 400-403. The utility is run a third time to change the temporary LSM0 addresses (900-903) to 404-407.

On all 9310 LSMs, drives are defined to the HSC from top to bottom, with `addr0` representing the topmost drive and `addrn` the bottommost drive.

However, on a 9740 10-drive panel LSM, the drives are populated and configured to the 9740 LSM from bottom to top. (9740 4-drive panels are configured to the 9740 LSM from top to bottom, as are all other LSM drive panels.)

An example showing how to define a 9740 10-drive panel containing five 9840 drives is:

```
SET SLIDRIVS(,,,,BD4,BD3,BD2,BD1,BD0),FORLSMID(lsm-id),
FORPANEL(panel)
```

For the SL3000, drives are defined from top to bottom, right to left, as looking from the front to the back of the library.

For the SL8500, drives are defined from top to bottom, right to left, as viewed from the outside of the library.

#### **FORLSMID(*lsm-id*)**

specifies that the operation is being restricted to a specific LSM.

*lsm-id* indicates the LSMid (*AA:LL*) containing the drive panel whose addresses are being changed. An LSMid (*lsm-id*) is made up of the ACSid (decimal 00-99) and the LSM number (decimal 00-99) separated by a colon (:).

#### **FORPANEL(*panel*)**

specifies that the operation is being restricted to a specific panel.

*panel* indicates the 2-digit ID of the panel containing the drives whose addresses are being changed. See "Panels" on page B-2 for a list of valid values.

---



---

**Note:** The specified *panel* must be an existing drive panel in the LSM.

---



---

#### **MODEL(*model-type*)**

optionally, specifies the transport model number.

*model-type* indicates the model type.

See [Appendix A, "MEDia, RECtech, and MODEL Values"](#) for a list of valid *model-type* values.

---



---

**Note:** The MODEL parameter is only used to specify the drive type for manual RTDs., which are defined in a nonexistent ACS. When an LSM is brought online, the (non-manual) drive type is automatically set.

---



---

#### **FORHOST(*host-id*)**

specifies that the operation is being restricted to a specific host. Only the specified hosts' drive device numbers are set. If omitted, the device numbers for all hosts are changed.

*host-id* indicates the 1- to 8-character host ID of the host, to which the operation is restricted.

---



---

**Note:** The number of drive positions for a specific host must equal the number of drive positions defined globally (issuing SET SLIDRIVS without the FORHOST parameter).

---



---

### **Running SET SLIDRIVS With the HSC Active**

The following applies to 9310 and 9740 LSMs only. It does not apply to the SL8500 or SL3000.

Ideally, the HSC should be shut down on all hosts when you specify the SET SLIDRIVS parameter. In some cases, however, the HSC can be left active without causing adverse results. [Table 3-15](#) defines options for running SET SLIDRIVS.

**Table 3–15 HSC State/SET SLIDRIVS Operation**

HSC State	Effect on SET SLIDRIVS
Down	If the HSC is down on all hosts, the HSC recognizes all new drive locations when it is initialized.
Active	The HSC can be up on all hosts only if new drives are being added at new drive locations. The HSC recognizes the new drive locations when it is recycled on a host.
Active	If unit addresses are changed or deleted for an existing drive location, either the affected LSM must be offline until the HSC has been recycled on all hosts, or the affected ACS must be offline to all affected hosts that access an HSC that has not been recycled.

### SET SLIDRIVS Procedure

The following procedures show one method for changing unit addresses with the HSC active.

#### For the 9310 and 9740:

For one host:

1. Modify the affected LSM offline (`MODify lsm-id OFFline`). The LSM will be offline to all hosts.
2. Use `SET SLIDRIVS` to update the CDS drive records while the Oracle StorageTek CSE(s) is making the hardware changes.
3. Recycle the HSC on one host.
4. Vary the affected ACS offline (`Vary ACS(acs-id) OFFline`) on all hosts running an HSC that has not been recycled.
5. Modify the affected LSM online (`MODify lsm-id ONline`). The LSM will be online to all hosts where the ACS is online. Only hosts that are online to the ACS and running a recycled HSC will mount to drives connected to this LSM.

For the remaining hosts, recycle the HSC, and the affected ACS will come up.

#### For the SL3000 and SL8500:

1. Make necessary hardware changes (add/delete/move drives).
2. Modify the `SET SLIDRIVS JCL` deck to reflect the changes.
3. Run `SET SLIDRIVS`.
4. Issue `MODIFY CONFIG` (or `MODIFY CONFIG ACS(xx)`)

The changes will cascade to all other hosts.

## SET SLISTATN

### Interfaces:

- `SLUADMIN` utility only
- UUI Support: No

### Subsystem Requirements:

Active HSC not required



### Description

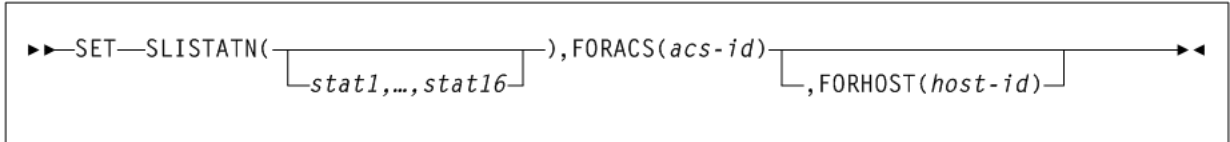
The SET SLISTATN command adds or changes LMU station addresses in an ACS. The change does not take effect until the affected HSC(s) is recycled.

An affected HSC is one that resides in the host specified in FORHOST. If FORHOST is omitted, all HSCs are relevant because the operation is not restricted to a specific host.

### Syntax

Figure 3–158 shows SET SLISTATN command syntax.

**Figure 3–158 SET SLISTATN command syntax**



### Parameters

As shown in Figure 3–158, the SET SLISTATN command includes the following parameters:

#### **stat1,...,stat16**

the LMU station addresses associated with a particular ACS. No station addresses are required. Up to 16 can be specified, separated by commas.

When adding one or more station addresses to an existing list of stations, you must specify all old stations as well as new ones. Any stations (for this ACS and Host ID) not specified here will be deleted and no longer available for use.

To remove station addresses for an ACS or host, do not specify station addresses following the SLISTATN parameter. For example:

```
SET SLISTATN( ), FORACS(01) , FORHOST(HSCA)
```

#### **FORACS(acs-id)**

specifies the ACS for which station addresses are being changed.

*acs-id* indicates the ACSid whose stations are being changed (00 - FF).

#### **FORHOST(host-id)**

specifies that the operation is being restricted to a specific host. Only the specified hosts' stations are set. If omitted, the stations for all hosts are changed.

*host-id* indicates the 1- to 8-character host ID for the host to which the operation is restricted.

## SET SMF

### Interfaces:

- SLUADMIN utility only
- UUI Support: No

### Subsystem Requirements:

Active HSC not required

**Description**

The `SET SMF` command adds a new SMF record type for HSC.

Until the HSC is recycled on all CPUs, the `ACTIVITIES` command produces incorrect results because some HSC systems are writing records with the old SMF type, and some with the new record type.

**Syntax**

Figure 3–159 shows `SET SMF` command syntax.

**Figure 3–159 SET SMF command syntax**

```
▶▶ SET—SMF(libtype)—————▶▶
```

**Parameters**

As shown in Figure 3–159, the `SET SMF` command includes the following parameters:

**libtype**

the SMF record type. The range is from 0 to 255, inclusive.

Since 128 through 255 are for user-defined records, it is recommended that a number from 128 through 255 be chosen, and that the number does not conflict with other user-defined record types.

**SET TAPEplex****Interfaces:**

- SLUADMIN utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

**Description**

The `SET TAPEplex` command sets a TapePlex name.

**Syntax**

Figure 3–160 shows `SET TAPEplex` command syntax.

**Figure 3–160 SET TAPEplex command syntax**

```
▶▶ SET—TAPEplex(tapeplex-name)—————▶▶
```

**Parameters**

As shown in Figure 3–160, the `SET TAPEplex` command includes the following parameters:

**tapeplex-name**

the 1-8 character TapePlex name.

The TapePlex name must start with an alphabetic or numeric character. Remaining characters may be letters, numbers, or hyphens, however, a hyphen may not be the last character. You can specify a NULL value (TAPEPLEX()) to allow the TapePlex name to be set automatically by the next SMC transaction.

---

**Note:** This name must match the TapePlex name specified on the SMC TAPEPLEX command on ALL SMC systems that communicate with this HSC system. See "TAPEPLEX" on page 2-79 for more information about the SMC TapePlex command.

---

**SET TCHNIQE****Interfaces:**

- SLUADMIN utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

**Description**

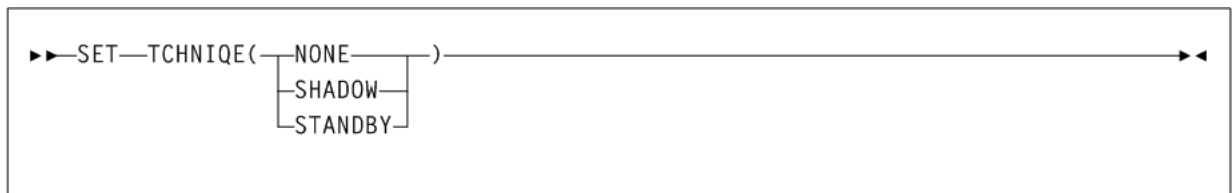
The SET TCHNIQE command sets the control data set recovery technique. This replaces the recovery technique currently defined in the CDS.

Refer to Oracle's ELS publication *Configuring HSC and VTCS* for information about the LIBGEN SLIRCVRY macro. A recovery technique value must be specified. There is no default value.

**Syntax**

Figure 3-57 shows SET TCHNIQE command syntax.

**Figure 3-161 SET TCHNIQE command syntax**

**Parameters**

As shown in Figure 3-161, the SET TCHNIQE command includes the following parameters:

**NONE**

specifies no form of recovery is used for the control data set. Thus, the primary control data set must be rebuilt, if inaccessible.

**SHADOW**

specifies that there is to be two distinct copies of the control data set (primary and secondary) for recovery purposes. It is recommended that these data sets reside on separate HDAs and separate strings. A journal is not recorded.

**STANDBY**

specifies that primary, secondary, and standby control data sets are to be recorded for recovery purposes. No journals are recorded during HSC operation.

**SET TECHNIQUE Procedure**

Use the following procedure to invoke the `TECHNIQUE` utility and to set the appropriate recovery technique value:

1. `ALLOCATE`, based on existing CDS definitions, any new CDS copies that will be needed for the subsequent `SET` recovery technique.
2. Stop the HSCs on all hosts configured to use the CDS you are intending to update with `SET` recovery technique.
3. Run the `BACKUP` utility.
4. Run the `RESTORE` utility to restore all CDS copies required to support the current recovery technique. If the recovery technique update in the following step requires new CDS copies, then include them in this restore.
5. Run the `SET` utility with the appropriate recovery technique value. It is important to note that the `SET` utility for this recovery technique update must include `DD` statements for all CDS copies defined in either the old or the new recovery technique.
6. Run the `BACKUP` utility again. This backup provides you with the ability to restore the updated CDS.

**SET VAULT****Interfaces:**

- `SLUADMIN` utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

**Description**

The `SET VAULT` command defines `VAULT` information for a server complex. This information is stored in the CDS and serves as the `VAULT` information source for all server systems that access the CDS.

---

---

**Note:** Before running this utility for the first time, you must run the `SET VAULTVOL` utility to create the Vault Volume records in the CDS.

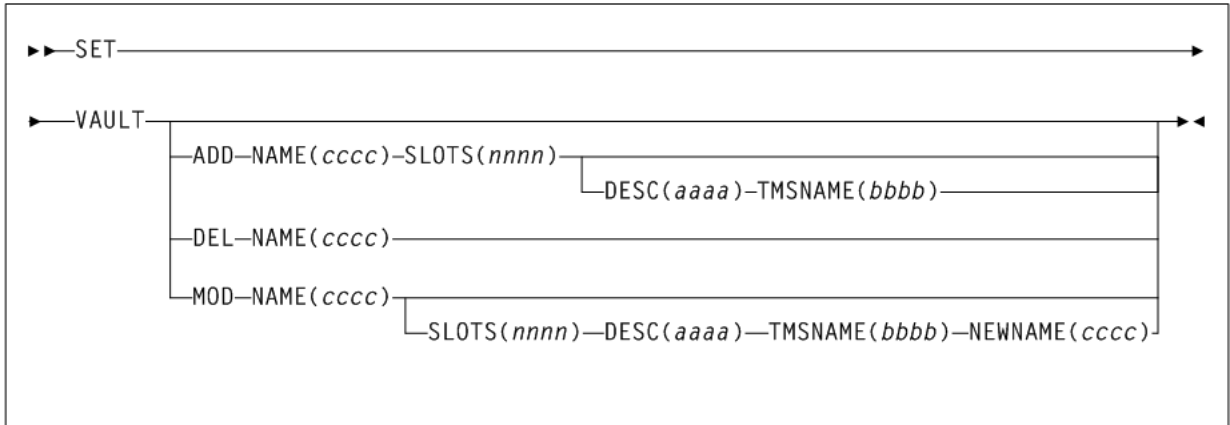
---

---

**Syntax**

Figure 3-162 shows `SET VAULT` command syntax.

Figure 3–162 SET VAULT command syntax



### Parameters

As shown in Figure 3–162, the SET VAULT command includes the following parameters:

#### ADD NAME(cccc)

Add a specified Vault to the CDS.

*cccc* indicates the Vault name. You can specify a maximum of 8 characters. NAME (*cccc*) is required.

#### SLOTS(nnnn)

optionally, specifies the number of slots in the Vault.

*nnnn* indicates the number of slots. The slot number can be 1-999,999,999. When specified with the MOD option, the SLOTS number must be greater than the current number of SLOTS defined to the Vault.

The total number of slots for all Vaults cannot exceed the number of Vault Volumes set by the SET VAULTVOL utility.

#### DESC(aaaa)

optionally, specifies the description of the Vault.

*aaaa* indicates the description. You can specify a maximum of 120 characters. If blank characters are included, the entire value must be enclosed in single quotes. This parameter is only valid with the ADD and MOD parameters.

#### TMSNAME(bbbb)

optionally, specifies the TMS Vault name.

*bbbb* indicates the Vault name. You can specify a maximum of 8 characters. This parameter is only valid with the ADD and MOD parameters.

#### DEL NAME(cccc)

Delete a specified Vault from the CDS.

*cccc* indicates the Vault name. You can specify a maximum of 8 characters. NAME (*cccc*) is required.

#### MOD NAME(cccc)

Modify a specified Vault from the CDS.

*cccc* indicates the Vault name. You can specify a maximum of 8 characters. `NAME (cccc)` is required.

**SLOTS(nnnn)**

optionally, specifies the number of slots in the Vault.

*nnnn* indicates the number of slots. The slot number can be 1-999,999,999. When specified with the `MOD` option, the `SLOTS` number must be greater than the current number of `SLOTS` defined to the Vault.

The total number of slots for all Vaults cannot exceed the number of Vault Volumes set by the `SET VAULTVOL` utility.

**DESC(aaaa)**

optionally, specifies the description of the Vault.

*aaaa* indicates the description. You can specify a maximum of 120 characters. If blank characters are included, the entire value must be enclosed in single quotes. This parameter is only valid with the `ADD` and `MOD` parameters.

**TMSNAME(bbbb)**

optionally, specifies the TMS Vault name.

*bbbb* indicates the Vault name. You can specify a maximum of 8 characters. This parameter is only valid with the `ADD` and `MOD` parameters.

**NEWNAME(cccc)**

optionally, specifies the new name of the Vault.

*cccc* indicates the Vault name. You can specify a maximum of 8 characters. This parameter is only valid with the `MOD` parameter.

## SET VAULTVOL

**Interfaces:**

- `SLUADMIN` utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

**Description**

The `SET VAULTVOL` command defines the `VAULT` volume area. This information is stored in the CDS and serves the `VAULT` volume area information source for all server systems that access the CDS.

---

---

**Note:**

- You must run this utility before running the `SET VAULT` utility.
  - There is no `SET` utility command to increase, decrease, or remove all vault volume records once the `SET VAULTVOL` utility is run. To increase the number of vault volume records, use the `Merge CDS` utility to migrate vaults and vaulted volumes to a CDS that has been configured with a greater number of vault volumes.
- 
-

## Syntax

Figure 3–163 shows SET VAULTVOL command syntax.

**Figure 3–163 SET VAULTVOL command syntax**

```
▶▶—SET—VAULTVOL—NBRVOLS—(nnnnnn)————▶▶
```

## Parameters

As shown in Figure 3–163, the SET VAULTVOL command includes the following parameters:

### NBRVOLS(nnnnnn)

specifies the number of VAULT volumes to be added to the CDS.

*nnnnnn* indicates the number of volumes. This number can be 1-9,999,999,999.

---



---

#### Note:

- Refer to Oracle’s ELS publication *Configuring HSC and VTCS* for details on calculating CDS space requirements for Vaulting.
  - The number of Vault volumes must be greater than the total number of slots for all Vaults created with the SET VAULT utility.
- 
- 

## SET VOLPARM

### Interfaces:

- SLUADMIN utility only
- UUI Support: No

### Subsystem Requirements:

Active HSC not required

### Description

The SET VOLPARM command defines the volume and pool information for a server complex. This information is stored in the CDS and is the volume and pool information source for all server systems (ELS 7.0 or higher) that access the CDS.

SET VOLPARM reads POOLPARM and VOLPARM control statements that define VOLUME/POOL information in the form of named pools of specific types, and their associated volser ranges.

Once SET VOLPARM is run, older volume and pool definitions are disabled for all server systems (ELS 7.0 or higher) that access the common CDS.

**Note:**

- HSC 6.2 or lower systems are unaware of the presence of the SET VOLPARM data and continue to operate as though the SET VOLPARM utility had not been run.
- Specify REGION=0M on the EXEC PGM=SLUADMIN statement when the number of input statements exceeds 10,000 card images.
- If you delete MVCs or VTVs from the configuration as a result of either SET VOLPARM or CONFIG MVCVOL or CONFIG VTVVOL processing:

If MVCs were deleted, you cannot re-enter the volsers into the configuration as VTVs.

If VTVs were deleted, you cannot re-enter the volsers into the configuration as MVCs.

Do not use the volsers for native HSC tapes.

Message SLS6944I indicates the number of MVCs or VTVs that have been deleted.

The following commands are disabled for systems (ELS 7.0 or higher) after SET VOLPARM:

HSC:

- VOLDEF
- SCRDEF

VTCS:

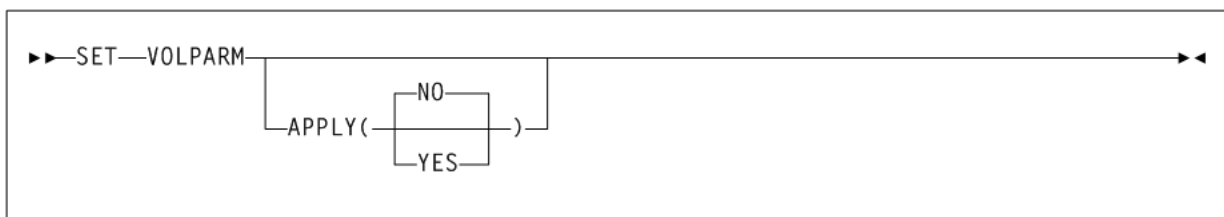
- MVCDEF
- CONFIG VTVVOL or CONFIG MVCVOL

The HSC/VTCS systems do not need to be recycled for the new volume and pool definitions to take effect. The attached HSC/VTCS in the CDS complex are notified of the definition changes and the volume and pool definitions are automatically active.

**Syntax**

Figure 3–164 shows SET VOLPARM command syntax.

**Figure 3–164 SET VOLPARM command syntax**

**Parameters**

As shown in Figure 3–164, the SET VOLPARM command includes the following parameters:



**APPLY**

specifies whether the `POOLPARAM` or `VOLPARAM` definitions are to be applied to the CDS:

**NO**

Validate the definitions but do not apply them to the CDS (the default).

**YES**

Apply the definitions. If you specify `YES`, the `SLSPARM` data set containing `POOLPARAM` and `VOLPARAM` control statements is read and applied to the CDS. Additionally, the date and time of this action is stored in the CDS, and is output by the `Display CDS` command.

`POOLPARAM` and `VOLPARAM` control statements are defined in the `SLSPARM` data set, and define `VOLUME` or `POOL` information in the form of named pools of specific types, and their associated `volser` ranges.

- When a `POOLPARAM` statement is read, all subsequent `VOLPARAMs` encountered until the next `POOLPARAM` statement are applied to the prior `POOLPARAM`.
- When `VOLPARAM` statements are read without a prior `POOLPARAM`, those `VOLSER` volumes will be considered non sub-pool (subpool 0) volumes.

See "[POOLPARAM Control Statement](#)" on page 3-343 and "[VOLPARAM Control Statement](#)" on page 3-348 for more information.

**Additional JCL Requirements**

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statement applies to the `SET VOLPARAM JCL`:

**SLSPARM**

specifies the data set that contains the `POOLPARAM` and `VOLPARAM` control statements that define volumes and pools.

**Using SET VOLPARAM to Disable POOLPARAM/VOLPARAM**

To disable `POOLPARAM` or `VOLPARAM`, run the `SET VOLPARAM` utility with an empty `SLSPARM` input file.

Once `POOLPARAM` or `VOLPARAM` is disabled:

- The previous version of `POOLPARAM` or `VOLPARAM` remains in effect for `MVC` and `VTV` definitions until you run `VTCS CONFIG`.
- The previous `POOLPARAM` or `VOLPARAM` `POOL` definitions remain in effect until you run the `VOLDEF`, `SCRPDEF` and `MVCDEF` commands on the HSC host(s).

**POOLPARAM Control Statement**

`POOLPARAM` control statements define pool information for a server complex in the form of named pools of specific types.

`VOLPARAM` statements define a `volser`, `volser` list, or `volser` range to be associated with the specified `POOLPARAM` statement.

`POOLPARAM` and `VOLPARAM` statements are defined in the `SLSPARM` data set, and are loaded using the `SET VOLPARAM` utility. See "[SET VOLPARAM](#)" on page 3-341 for more information.

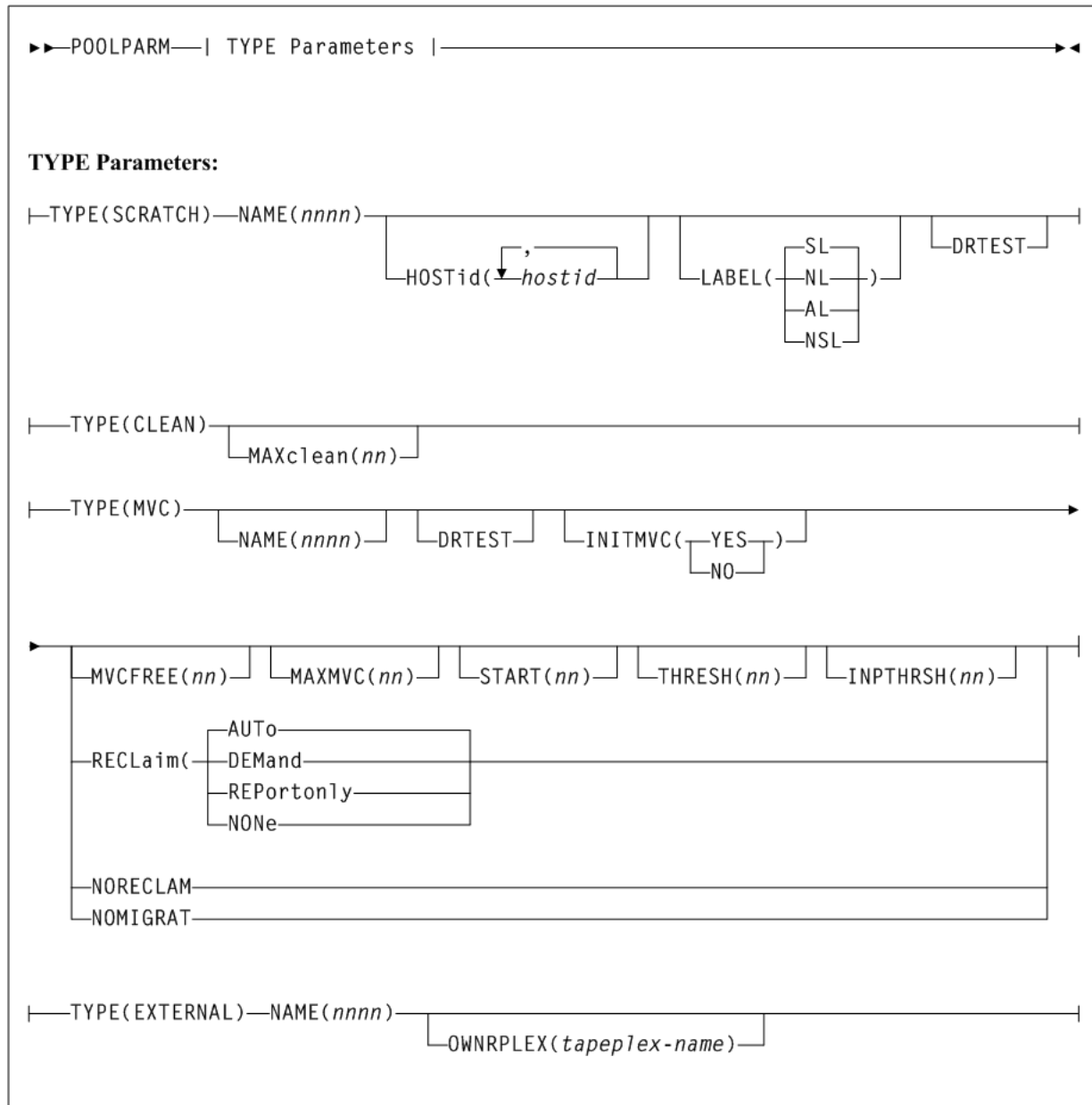
- When a `POOLPARAM` statement is read, all subsequent `VOLPARAM` statements encountered before the next `POOLPARAM` statement are applied.

- When VOLPARM statements are read without a prior POOLPARM, those VOLSER volumes are considered non-subpool (subpool 0) volumes.

**Syntax**

Figure 3–165 shows POOLPARM control statement syntax.

**Figure 3–165 POOLPARM control Statement syntax**



**Parameters**

As shown in Figure 3–165, the POOLPARM control statement includes the following parameters:

**TYPE**

specifies the type of pool definition. Valid types include SCRATCH, CLEAN, MVC, and EXTERNAL.

**TYPE(SCRATCH) Subparameters**

If you specify `TYPE(SCRATCH)`, the following subparameters apply:

**NAME(nnnn)**

specifies the name of the volume pool.

`NAME` is required for `TYPE(SCRATCH)` pools. Additionally, `NAME(DEFAULTPOOL)` is not valid for `TYPE(SCRATCH)` pools.

`nnnn` indicates the volume pool name. A maximum of 13 characters is allowed. Blank spaces are not allowed.

**HOSTid(host-id)**

optionally, defines the hosts that can access the named sub-pool.

`host-id` indicates the host id. Multiple `host-ids` may be specified, separating each `host-id` value with a comma. If a `host-id` is not specified, the default is all hosts.

`HOSTid` determines which client hosts can mount volumes from the named subpool. If a mount request is received from a client not specified in the `HOSTid` list, the HSC rejects the mount request and the mount is not satisfied. If you did not specify an `ALIASHOST` on the `SMC SERVER` command, then specify the 4 character `SMFID` of the MVS hosts that will access the subpool. Otherwise, you can specify up to 8 characters for the host name matching the `SMC ALIASHOST` specification.

**LABEL**

optionally, specifies the label type associated with the named subpool. Specify one of the following:

- `SL` indicates standard-labeled tape. This is the default if the parameter is omitted.
- `AL` indicates ANSI-labeled tape.
- `NL` indicates nonlabeled tape.
- `NSL` indicates nonstandard labeled tape.

---

**Note:** Scratch volume requests without a sub-pool (sub-pool 0) will come from volumes not defined in named subpools.

---

**DRTEST**

optionally, specifies that the scratch pool is to be used in `DRTEST` environments.

**TYPE(CLEAN) Subparameters**

If you specify `TYPE(CLEAN)`, the following subparameters apply:

**MAXclean**

optionally, specifies the maximum cleaning cartridge usage for the named sub-pool. Do not specify values that exceed the manufacturer's recommendation for usage.

The volumes defined in the `CLEAN` sub-pools must begin with the cleaning cartridge prefix defined in the `LIBGEN`.

---



---

**Note:** If `MAXclean` is not specified in the `VOLPARM` statement, the value or default of the `MAXclean` parameter in the `MNTD` operator command is used for all cleaning cartridge counts.

---



---

### TYPE(MVC) Subparameters

If you specify `TYPE(MVC)`, the following subparameters apply:

#### NAME(nnnn)

specifies the name of the volume pool. This parameter is optional for `TYPE(MVC)` pools. Additionally, `NAME(DEFAULTPOOL)` is not valid for `TYPE(MVC)`. `NAME(ALL)` is not valid for `TYPE(MVC)` pools.

*nnnn* indicates the volume pool name. A maximum of 13 characters is allowed. Blank spaces are not allowed.

#### DRTEST

optionally, specifies that the MVC pool is to be used only in `DRTEST` environments.

#### INITMVC

optionally, enables or disables the MVC initialization feature.

#### YES

enables the MVC initialization feature.

#### NO

disables the MVC initialization feature.

#### MVCFREE(nn)

optionally, 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.

*nn* indicates the number of MVCs. Valid values are 0 to 255. There is no default; if not specified, the `CONFIG RECLAIM` value (or default) is used.

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

#### MAXMVC(nn)

optionally, specifies the maximum number of MVCs to be processed in a single space reclamation run.

*nn* indicates the number of MVCs. 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.

#### START=nn

optionally, 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.

*nn* is a percentage value, which is equal to:

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

Where:

- *Reclaim Candidates* indicates 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. There is no default. If not specified, the `CONFIG RECLAIM` value (or default) is used.

### **THRESH(nn)**

optionally, specifies the percentage of fragmented space that makes an MVC eligible for demand or automatic reclamation.

*nn* indicates the percentage. Valid values are 4 to 98. There is no default; if not specified, the `CONFIG RECLAIM` value (or default) is used.

### **INPTHRSH(nn)**

optionally, specifies the percentage of fragmented space that makes an MVC in partitioned format (within the pool) eligible for dynamic reclaim processing.

This value overrides the global `INPTHRSH` value specified in the `CONFIG RECLAIM` statement. If this parameter is not specified, the current active global `INPTHRSH` value is used.

*nn* indicates a percentage between 3 and 97. This value must be less than the `THRESHLD` value.

---



---

**Note:** If `INPTHRSH` is specified, `THRESHLD` must also be specified. Both `INPTHRSH` and `THRESHLD` play roles when space reclamation processes MVCs in partitioned format. Refer to Oracle's ELS publication *Configuring HSC and VTCS* for more information.

---



---

### **RECLaim**

optionally, specifies space reclaim behavior of the `MVCPool`.

#### **AUTO**

VTCS will output message `SLS6699E` and schedule automatic space reclaims. Reclaims may also be invoked using the `RECLAIM` command. This is the default.

#### **DEMAND**

VTCS will not output message `SLS6699E` and will not schedule automatic space reclaims. Reclaims may be invoked using the `RECLAIM` command.

`RECLaim(DEMAND)` is mutually exclusive with `MVCFREE` and `START`.

#### **REPORTonly**

VTCS will output message `SLS6699E` but will not schedule automatic space reclaims. Reclaims may not be invoked using the `RECLAIM` command.

`RECLaim(REPORTonly)` is mutually exclusive with `MAXMVC`, `START`, and `THRESH`.

---



---

**Note:** `REPORTonly` is equivalent to, and replaces, the `NORECLAM` parameter though `NORECLAM` is still tolerated if the `RECLaim` parameter is not specified.

---



---

#### **NONE**

VTCS will not output message `SLS6699E` and will not schedule automatic space reclaims. Reclaims may not be invoked using the `RECLAIM` command.

RECLaim(NONe) is mutually exclusive with MVCFREE, MAXMVC, START, and THRESH.

**NORECLAM**

optionally, specifies that the named MVCPOOL will not be eligible for space reclamation.

---

---

**Note:** The use of the MVCFREE, MAXMVC, THRESH, INPTHR and START parameters override the values specified on CONFIG for the named *sub-pool*.

---

---

**NOMIGRAT**

optionally, specifies that the MVCPOOL will not be eligible for selection for migration MVCs This enables MVC information to be retained in the CDS virtual area. The MVCs will not be used for new migrations.

**TYPE(EXTERNAL) Subparameters**

If you specify TYPE(EXTERNAL), the following subparameters apply:

**NAME(nnnn)**

specifies the name of the volume pool. This parameter is required for TYPE(EXTERNAL) pools.

*nnnn* indicates the volume pool name. A maximum of 13 characters is allowed. Blank spaces are not allowed.

**OWNRPLEX**

optionally, specifies the name of the TapePlex that owns the volumes in the external pool.

For VTVs, this controls volumes imported via Cross-TapePlex Replication. A VTV can only be imported if the sending system's TapePlex name matches the OWNRPLEX value.

Volumes in an EXTERNAL pool can be scratched only if the scratch request originates from the TapePlex matching the OWNRPLEX name. The TapePlex name is defined via the SMC TAPEPLEX NAME parameter and the CONFIG TAPEPLEX THISPLEX parameter.

Alternatively, volumes in an external pool with an OWNRPLEX name of LOCALHSC can be scratched on the local system (but not used as scratch volumes on the local system).

---

---

**Note:** Refer to Oracle's ELS publications *Managing HSC and VTCS* and *ELS Disaster Recovery and Offsite Data Management Guide* for more information about Cross-TapePlex Replication.

---

---

**VOLPARM Control Statement**

VOLPARM control statements define volume attributes for a server complex. These statements define a volser, volser list, or volser range to be associated with a specified POOLPARM statement.

POOLPARM and VOLPARM statements are defined in the SLSPARM data set, and are loaded using the SET VOLPARM utility. See "SET VOLPARM" on page 3-341 for more information.

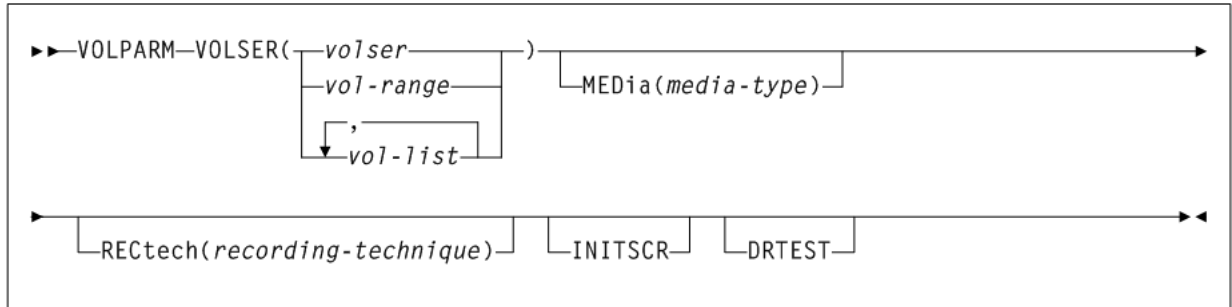
VOLPARM statements follow a POOLPARM statement and apply only to that POOLPARM.

- When a POOLPARM statement is read, all subsequent VOLPARM statements encountered before the next POOLPARM statement are applied.
- When VOLPARM statements are read without a prior POOLPARM, those VOLSER volumes are considered non-subpool volumes.

### Syntax

Figure 3–166 shows VOLPARM control statement syntax.

**Figure 3–166 VOLPARM Control Statement syntax**



### Parameters

As shown in Figure 3–166, the VOLPARM control statement includes the following parameters:

#### **VOLSER(volser, vol-range, or vol-list)**

specifies one or more volumes to which this definition applies.

*volser*, *vol-range*, or *vol-list* indicates a single VOLSER, a VOLSER range, or a list of VOLSERs.

Do not overlap the ranges. Doing so causes an error which invalidates the entire POOLPARM definition.

#### **MEDia(media-type)**

optionally, specifies the media type for the volume(s) specified on the VOLSER parameter.

*media-type* indicates the media type.

See Appendix A, "MEDia, RECtech, and MODel Values" for a list of valid *media-type* values.

Note the following:

- Only specific media types can be entered. Generic media types, that is, LONGItud and HELical cannot be specified.
- If this parameter is not specified, a default is chosen based on the value of the RECtech parameter. Table A–2 shows default values used if MEDia is omitted.
- If both media type and recording technique are omitted, all media types and recording techniques are assumed to be eligible. It is recommended that MEDia always be specified on all VOLPARM statements.

**RECtech(recording-technique)**

optionally, specifies the method used to record data on the media for the volume(s) specified on the VOLSER parameter.

*recording-technique* indicates the recording technique.

See [Appendix A, "MEDia, RECtech, and MODel Values"](#) for valid *recording-technique* values.

If this parameter is not specified on any matching VOLPARM statement, a default is chosen based on the value of the MEDia parameter. [Table A-4](#) shows default values used if RECtech is omitted.

**INITSCR**

optionally, specifies that the VIRTUAL volume has an initial status of SCRATCH.

The INITSCR parameter is only valid with MEDIA (VIRTUAL) and/or RECTECH (VIRTUAL) volumes. INITSCR only has meaning when the VTV is first defined.

**DRTEST**

optionally, specifies that the volumes in the defined range are to be used only in DRTEST environments. The DRTEST parameter is only valid for VOLPARM statements defined before the first POOLPARM statement.

**SET VOLPARM UPDATE****Interfaces:**

- SLUADMIN utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

**Description**

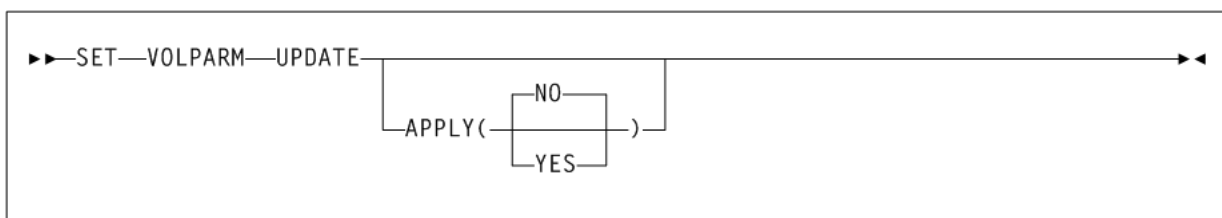
The SET VOLPARM UPDATE command defines the changes to the volume and pool information for a server complex. The information stored in the CDS by the SET VOLPARM utility is changed to reflect the new VOLUME or POOL information.

The HSC/VTCS systems do not need to be recycled for the updated volume pool definitions to take effect. The attached HSC/VTCS in the CDS complex are notified of the definition changes and the volume and pool definitions are automatically active.

**Syntax**

[Figure 3-167](#) shows SET VOLPARM UPDATE command syntax.

**Figure 3-167 SET VOLPARM UPDATE command syntax**





## Parameters

As shown in [Figure 3–167](#), the `SET VOLPARM UPDATE` command includes the following parameters:

### APPLY

specifies whether the `POOLPARM` or `VOLPARM` changes are to be applied to the CDS:

#### NO

Validate the changes but do not apply them to the CDS (the default).

#### YES

Apply the changes. If you specify `YES`, the `SLSPARM` data set containing the `POOLPARM` and `VOLPARM` change control statements is read and applied to the CDS.

`POOLPARM` and `VOLPARM` change control statements are defined in the `SLSPARM` data set and define `VOLUME` or `POOL` information in the form of named pools of specific types, and their associated `volser` ranges.

- When a `POOLPARM` statement is read, all subsequent `VOLPARM` statements encountered until the next `POOLPARM` statement are applied to the prior `POOLPARM`.
- When `VOLPARM` statements are read without a prior `POOLPARM`, those `VOLSER` volumes will be considered non-subpool (subpool 0) volumes.

## Additional JCL Requirements

In addition to the required JCL definition statements described in ["Invoking SLUADMIN"](#) on page 5-1, the following definition statement applies to the `SET VOLPARM UPDATE` JCL:

### SLSPARM

specifies the data set that contains the `POOLPARM` and `VOLPARM` change control statements that define the pool and volume changes to the CDS volume and pool information.

## POOLPARM Change Control Statement

`POOLPARM` Change control statements define new pool information for a server complex in the form of named pools of specific types.

`VOLPARM` Change control statements define or remove a `volser`, `volser` list, or `volser` range associated with the specified `POOLPARM` Change statement.

`POOLPARM` and `VOLPARM` Change statements are defined in the `SLSPARM` data set, and are loaded using the `SET VOLPARM UPDATE` utility. See ["SET VOLPARM"](#) on page 3-341 for more information.

- When a `POOLPARM` Change statement is read, all subsequent `VOLPARM` Change statements encountered before the next `POOLPARM` Change statement are applied.
- When `VOLPARM` Change statements are read without a prior `POOLPARM`, those `VOLSER` volumes are considered non-subpool (subpool 0) volumes.

## Syntax

[Figure 3–64](#) shows `POOLPARM` Change control statement syntax.

**Figure 3–168 POOLPARM Change control statement syntax**

```

▶▶POOLPARM—| change type |—| pool parameters |—————▶▶

```

**Parameters**

As shown in [Figure 3–168](#), the POOLPARM Change control statement includes the following parameters:

**change-type**

specify one of the following to add, delete, or update a pool:

**ADD**

A new pool is to be added to the server complex. VOLPARM change statements that follow this POOLPARM change statement will define the volumes associated with this pool.

**DELETE**

An existing pool is to be deleted. When the pool is deleted, the associated volumes will also be deleted.

**UPDATE**

An existing pool definition is to be changed. UPDATE can be used to change POOLPARM parameters. For example, the value of the MVCFREE parameter may be changed. UPDATE is also used to associate the VOLPARM change statements that follow with the correct pool.

---



---

**Note:** UPDATE cannot be used to change the name of a pool.

---



---

**pool-parameters**

The POOLPARM statement parameters. See "[POOLPARM Control Statement](#)" on page 3-343 for valid parameters.

---



---

**Note:** When deleting or updating a POOLPARM statement, specify all of the parameters of the original POOLPARM statement.

---



---

**VOLPARM Change Control Statement**

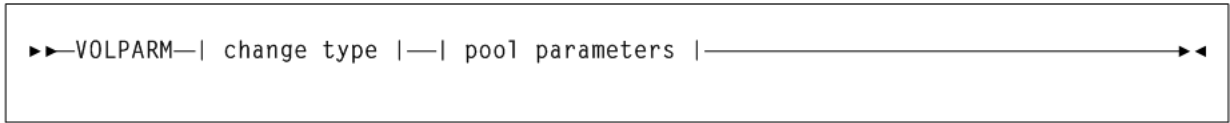
VOLPARM change control statements define or remove a specific volser, volser list, or volser range associated with the previous POOLPARM UPDATE change statement.

POOLPARM and VOLPARM change statements are defined in the SLSPARM data set, and are loaded using the SET VOLPARM UPDATE utility.

- When a POOLPARM change statement is read, all subsequent VOLPARM change statements encountered before the next POOLPARM change statement are applied.
- When VOLPARM change statements are read without a prior POOLPARM, those VOLSER volumes are considered non-subpool (subpool 0) volumes.

**Syntax**

[Figure 3–169](#) shows VOLPARM Change control statement syntax.

**Figure 3–169 VOLPARM Change control statement syntax****Parameters**

As shown in [Figure 3–169](#), the VOLPARM Change control statement includes the following parameters:

**change-type**

specify one of the following to add or delete a volume:

**ADD**

A new volume, volume list or range is added to the pool defined by the previous POOLPARM UPDATE statement. When no prior POOLPARM UPDATE statement exists, the VOLPARM information will be added as non-subpool (subpool 0) volumes.

**DELETE**

An existing volume, volume list or range is to be deleted.

**volume-parameters**

The VOLPARM statement parameters. See "[VOLPARM Control Statement](#)" on page 3-348 for valid parameters.

---



---

**Note:** When deleting a VOLPARM statement, specify all of the parameters of the original VOLPARM statement.

---



---

**SET VOLPARM JOIN****Interfaces:**

- SLUADMIN utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

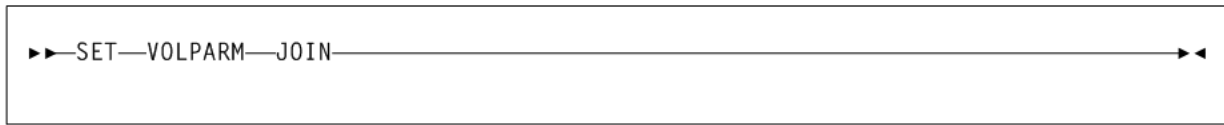
**Description**

The SET VOLPARM JOIN command merges contiguous volume ranges defined on multiple VOLPARM statements into one new VOLPARM statement.

The output of this utility can be used as input to the SET VOLPARM utility to replace the original server complex volume and pool information in the CDS.

**Syntax**

[Figure 3–170](#) shows SET VOLPARM JOIN command syntax.

**Figure 3–170 SET VOLPARM JOIN command syntax****Parameters**

None.

**Additional JCL Requirements**

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statement applies to the SET VOLPARM JOIN JCL:

**SLSPARM**

specifies the input data set that contains the POOLPARM and VOLPARM control statements that define the pool and volume contained in the CDS.

**SLSPARM2**

specifies the output data set that contains the POOLPARM and the VOLPARM statements after the volume range merge operations have been accomplished.

## SRVlev

**Interfaces:**

- Console or PARMLIB only
- UI Support: No

**Subsystem Requirements:**

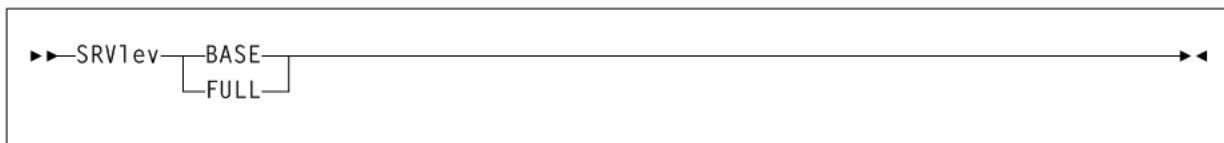
Active HSC at BASE or FULL service level

## Description

The SRVlev command sets the HSC service level. See "[HSC Service Levels](#)" on page 1-8 for information about BASE and FULL service levels.

## Syntax

[Figure 3–171](#) shows SRVlev command syntax.

**Figure 3–171 SRVlev command syntax****Parameters**

As shown in [Figure 3–171](#), the SRVlev command includes the following parameters:

**BASE**

specifies that the HSC is to operate at the `BASE` service level.

---

**Note:** When the HSC service level drops to `BASE`, host-to-host communications using the LMU method are switched to the CDS method. When the HSC `FULL` service level is restored, you must issue the `COMMPATH` command to return to LMU communications.

---

**FULL**

specifies that the HSC is to operate at the `FULL` service level.

---

**Note:** When the HSC service level is brought from `BASE` up to `FULL`, outstanding mount requests are resolved.

---

## STOPMN

**Interfaces:**

- Console or `PARMLIB` only
- UUI Support: No

**Subsystem Requirements:**

Active HSC at `BASE` or `FULL` service level

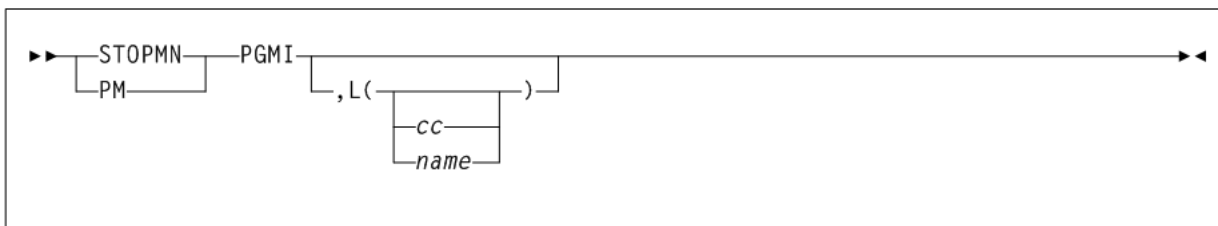
## Description

The `STOPMN` command stops monitoring of cartridge move requests from the programmatic interface.

## Syntax

Figure 3–172 shows `STOPMN` command syntax.

**Figure 3–172** *STOPMN command syntax*



## Parameters

As shown in Figure 3–172, the `STOPMN` command includes the following parameters:

**PGMI**

specifies that monitoring of cartridge move requests from the programmatic interface is to be terminated.

**,L(cc or name)**

optionally, identifies the console where the monitoring information is being displayed. If this parameter is omitted or if L= is specified without a console ID or console name, the monitoring being displayed on the console that issued the command is terminated.

*cc* optionally indicates the console ID. Allowable values are decimal in the range from 00 through 99. Specifying 00 stops information from being sent explicitly to the hardcopy log.

*name* optionally indicates the console name.

## SWitch

**Interfaces:**

- Console or PARMLIB only
- UUI Support: No

**Subsystem Requirements:**

Active HSC at FULL service level

## Description

The `SWitch` command can be used to manually reverse the roles of the master and standby LMUs or Library Controllers (LCs). The following configurations support switching:

- dual LMU
- SL8500 dual Library Controller (LC)

In a dual LMU environment, this command initiates an IPL in the master LMU, causing the current standby LMU to become the master LMU. If the former master LMU completes the IPL successfully, it then assumes the role of the standby LMU.

In a dual LC configuration for a multiple SL8500 library ACS, the `SWitch` command issues a request to the LMU to switch the library LCs by library ID.

---

---

**Caution:**

- Issue the `Display ACS` command before you enter the `SWitch` command to ensure the library to switch includes an assigned TCP/IP address or host name for the standby. If it does not, do not switch the library because doing so causes the HSC to lose communication with the library.
  - Before you issue the `SWitch` command, stop all HSC activity.
  - After you issue the `SWitch` command, do not start HSC on any other hosts and do not start any HSC activity (mounts, dismounts, `Display ACS`, or any other commands) until the switch process is complete.
- 
-

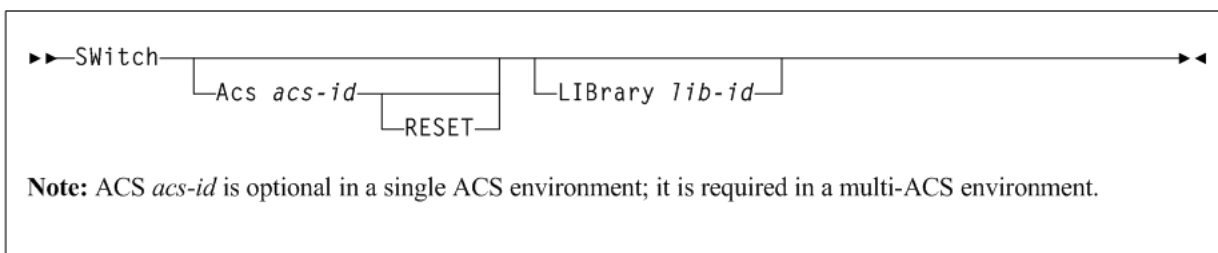
**Note:**

- For this release, only the SL8500 library is supported for dual LC switching.
- For an automatic switch to occur (for example, from LCA to LCB or visa versa), the LC must fail. In this case, if there is a network/communication problem with LCA, the HSC goes into network recovery for the LCA connection(s). If you cannot recover or fix the connection, issue the `SWitch` command to switch LCA to LCB manually.
- Before you issue the `SWitch` command, if the library is partitioned, be aware of other host software groups (ACSLs, HSC, ELS) using other partitions in the same library.

**Syntax**

Figure 3–173 shows `SWitch` command syntax.

**Figure 3–173** *SWitch command syntax*

**Parameters**

As shown in Figure 3–173, the `SWitch` command includes the following parameters:

**Acs acs-id**

optionally, indicates that an ACSid is being specified. This parameter is required for a multiple-ACS environment configuration or if an *acs-id* is specified.

*acs-id* identifies the ACS that must switch the current active connection as standby and bring the standby connection online as active.

In a single ACS configuration, *Acs acs-id* is optional and defaults to ACS 00. If the *acs-id* defaults to 00 or is specified with no other parameters, then all of the ACS's active connections are switched to standby, and the standby connections are switched to active.

**RESET**

Optionally, used in conjunction with ACS *acs-id* to reset the internal switch setting and internal conflict counts for an ACS. This is helpful when a `SWitch` is invoked for an SL8500 library and one or more LSMs fail to reach ready state after the `SWitch` process completes.

Before issuing a `SWitch RESET` command, it is recommended that you enter the `Display Status` command to determine which ACS(s) are in the switch process and which ACSs are conflict active.

Verify that the library has completed the `SWitch` process before you consider issuing a `SWitch RESET` command. Also verify that a `SWitch RESET` is complete before you consider issuing another `SWitch` command.

Use the following guidelines to determine when to issue a `SWitch RESET` command after a `SWitch ACS` command is issued:

1. Issue `SWitch RESET` after the `SWitch` process completes, as indicated by the following message:

```
SLS0694I ACS aa: Switch has completed
```

If this message is not displayed, wait for the `SLS6013I` message to be displayed for the active connection before issuing the `SWitch RESET` command.

2. If you receive the following message after the switch process has completed:

```
SLS1004I ACS aa cannot switch; ACS is busy
```

Issue the `SWitch RESET` command on another host. If the `SLS1004I` message is displayed again, bring up another HSC host and issue the `SWitch RESET` command, or restart an HSC host and then issue the `SWitch RESET` command.

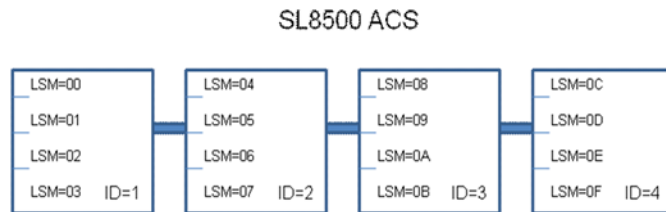
**Library lib-id**

optionally, specifies the SL8500 library ID for the connection switch. This parameter applies only to the SL8500 library.

*lib-id* indicates the library id from 1 to 9 or A to G.

For an SL8500 four library ACS cluster, each library is assigned an ID from 1 to 4, as shown in the following figure:

**Figure 3–174 SL8500 Cluster Library IDs**



An ACS can have up to 16 library connections with A and B Library Controller (LC) connections. For the initial release, only one library (preferred to be ID 1) can have A and B LC connections. The other libraries can have a single LC connection. Each LC can handle dual TCP/IP (2B and 2A). The following table describes the LSM ID correlation to Library ID:

**Table 3–16 LSM ID and Library ID**

Library ID	LSM ID Range
1	00-03
2	04-07
3	08-0B
4	0C-0F
5	10-13



**Table 3–16 (Cont.) LSM ID and Library ID**

Library ID	LSM ID Range
6	14-17
7	18-1B
8	1C-1F
9	20-23
A	24-27
B	28-2B
C	2C-2F
D	30-33
E	34-37
F	38-3B
G	3C-3F

**Note:**

- Before issuing this command, use the `Display ACS acs-id` command to see if the library to switch has a TCP/IP address or Host Name assigned to the standby. If there is not a TCP/IP address or Host Name assigned to the standby, do not switch the library as this will cause HSC to lose communication to the library.
- For a switch to occur (for example, from LCA to LCB), the LC must fail. For example, if there is a network/communication problem with LCA, HSC enters network recovery for the LCA connection(s). If you cannot recover/fix the connection, then issue the `HSC Switch` command to manually switch LCA to LCB.
- If the library is partitioned, please be aware of the other host software groups (ACSLs, HSC, ELS) that are using other partitions in the same library before issuing the `Switch` command.

**Examples**

The following example shows the use of the `SWitch` command to switch control of ACS 00 to the Standby ACS:

```
SWITCH ACS 00
```

The following example shows the use of the `SWitch` command to switch control of ACS 00 to the Standby Library:

```
SWITCH ACS 00 LIB 1
```

**TRace****Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at BASE or FULL service level

**Description**

The TRace command enables or disables tracing of events for selected HSC and VTCS components.

The MVS GTF GTRACE facility performs the actual tracing. To create GTF output for the event, GTF must be started and the TRace command must be issued before the event.

**Syntax**

Figure 3–175 shows TRace command syntax.

**Figure 3–175 TRace command syntax**

**Parameters**

As shown in Figure 3–175, the TRace command includes the following parameters:

**OFF**

optionally, turns off tracing for the specified component(s).

**comp-name or comp-list**

one or more components for which tracing is to be enabled or disabled. If a list of component names is specified, separate the identifiers with blank spaces or commas. Valid component names are listed below. Uppercase letters denote the minimum abbreviations allowed.

The following table describes valid components:

**Table 3–17 TRace Components**

Component Name	Description
ALloc	Allocation volume lookup
AScomm	Address Space Communications
CAP	CAP Common
COntfigur	Configuration Control
Database	Database server
FP	Function Points. FP tracing is used only by Oracle developers.
HComm	Host Communications
Initiali	Initialization
Lmu	LMU server
LS	LibraryStation

**Table 3–17 (Cont.) TTrace Components**

Component Name	Description
Mount	Mount/Dismount
Operator	Operator Commands
Recovery	Recovery (Note: must also trace SERVICE)
SERVICE	Services
Utilities	Utilities
UUI	Unified User Interface
Volume	Volume/Cell
VTcs	VTCS
Wto	WTO Services
XML	XML Interface

**Note:**

- When tracing is enabled or disabled for one or more components, the status of all component tracing is displayed.
- By default, all TTrace command output goes to GTF. For example, to trace an LMU and send the output to GTF, issue TR LMU. No other parameters are necessary.
- To trace the Recovery component, you must also trace the SERVICE component (a service is used to trace Recovery).
- The LS component displays in message SLS0068I only if LibraryStation has been initialized.

## TRACELKP

**Interfaces:**

- Console or PARMLIB only
- UUI Support: No

**Subsystem Requirements:**

Active HSC at BASE or FULL service level

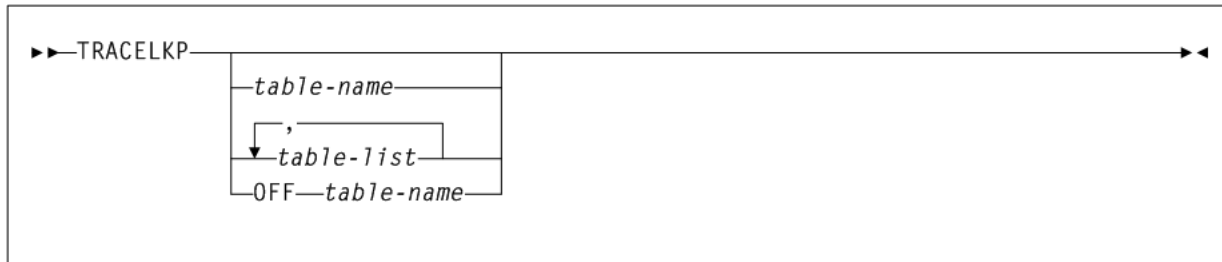
### Description

The TRACELKP command enables or disables tracing of LOOKUP events associated with HSC definition files. The GTF GTRACE facility is used to perform the tracing.

### Syntax

Figure 3–176 shows TRACELKP command syntax.

**Figure 3–176 TRACELKP command syntax**



## Parameters

As shown in [Figure 3–176](#), the TRACELKP command includes the following parameters:

### table-name or table-list

one or more LOOKUP definition files for which tracing is to be enabled or disabled. If a list of event names is specified, separate the identifiers with blank spaces or commas.

When LOOKUP tracing is enabled or disabled for one or more files, the status of all LOOKUP tracing is displayed. All LOOKUP tracing output goes to GTF.

Valid LOOKUP table names are listed in the table below. Uppercase letters denote the minimum abbreviations allowed.

**Table 3–18 TRACELKP Lookup Names**

LOOKUP name	Description
VOLATTR	VOLATTR (VOLDEF) table
LMUPATH	LMUPATH (LMUPDEF) table
MVCPOOL	MVCPOOL (MVCDEF) table
MGMTCLAS	MGMTCLAS (MGMTDEF) table
STORCLAS	STORCLAS (STORDEF) table
LOOKFAIL	Trace the LOOKUP failures in detail. If LOOKFAIL is specified, the detail failure trace records are output for all events that are ON.
MIGRSEL	MIGRSEL (MGMTDEF) table
MIGRVTV	MIGRVTV (MGMTDEF) table

### OFF table-name

optionally, disables tracing for the specified component(s). *table-name* indicates the components to be disabled.

## UEXIT

### Interfaces:

- Console or PARMLIB only
- UUI Support: No

### Subsystem Requirements:

Active HSC at BASE or FULL service level

## Description

The UEXIT command defines how HSC processes your user exits.

HSC user exits allow you to invoke your own processing routines at particular points during HSC processing. User exits controlled by the HSC are loaded at HSC initialization from the load library identified by DDNAME SLSUEXIT. The following user exits are managed by the HSC using the UEXIT command:

- SLSUX03
- SLSUX05
- SLSUX06
- SLSUX14
- SLSUX15

Using the UEXIT command, you can start the HSC with a user exit disabled and then enable the exit at any time the HSC is operational. If an exit does not perform as expected, make the necessary changes and load it again.

The UEXIT command optionally enables you to provide unique user exit load module names and to create different versions of an exit that can be run at different times (for example, day shift versus night shift). The load modules are contained in a user-defined load module library described at HSC startup.

---



---

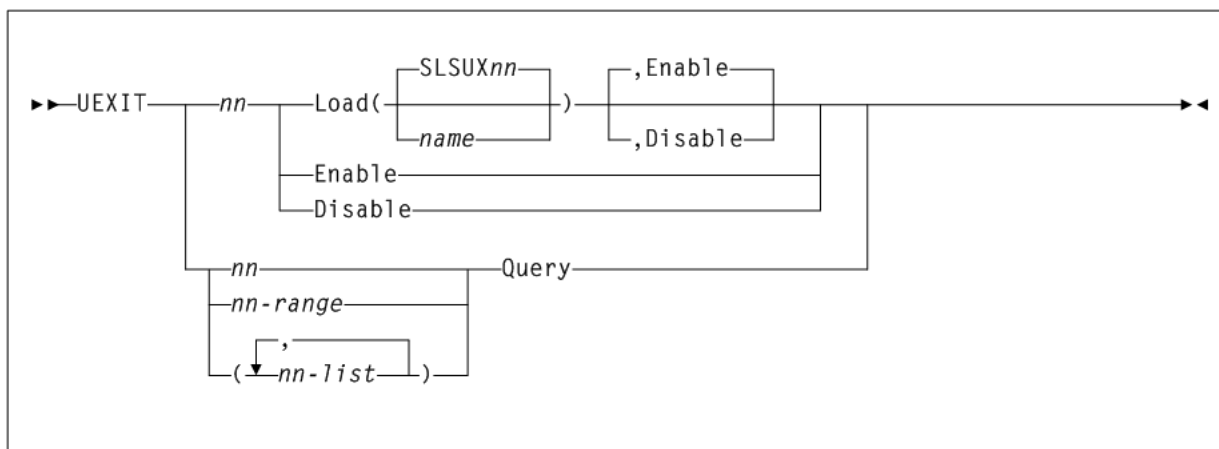
### Note:

- User Exit 03 and User Exit 05 are legacy interfaces and have been replaced by alternative facilities that do not require a User Exit.
  - User Exit 03 is enabled at HSC initialization. The UEXIT command cannot be used to enable, disable, or reload this exit.
- 
- 

## Syntax

Figure 3–177 shows UEXIT command syntax.

**Figure 3–177 UEXIT command syntax**



## Parameters

As shown in Figure 3–177, the UEXIT command includes the following parameters:

**nn**

specifies the user exit number. Refer to Oracle's ELS publication *ELS Legacy Interfaces Reference* for user exit numbers.

**Load**

causes the specified module to be loaded into storage.

**SLSUXnn**

specifies the default name for the user exit. If name is not specified, then the default name is used when the HSC loads the module into storage.

**name**

the name of the module to be used for this exit. This is the entry point name which is used to link-edit the load module.

**Enable**

indicates that upon completion of the command the specified exit is considered as being active. *Enable* is the default.

**Disable**

indicates that upon completion of the command the specified exit is considered as being inactive.

**Enable**

optionally, specifies that the most current module for the specified user exit (*nn*) is to be enabled. This parameter may also be specified with the *Load* parameter.

**Disable**

optionally, specifies that the most current module for the specified user exit (*nn*) is to be disabled. This parameter may also be specified with the *Load* parameter.

**nn, nn-range, or nn-list**

can be used with the *Query* parameter to specify a single user exit, a range of exits, or a list of exits.

When specifying a range of user exit numbers, the beginning number must be separated from the ending number by a hyphen.

For example: (04-09)

In a list of user exits, the user exit numbers must be separated by commas and the list must be enclosed in parentheses.

For example: (01,04,10)

**Query**

requests the status of all the currently loaded versions of the specified user exit number(s). A display of user exit status may be specified for a single user exit, a range of exits, or a list of exits.

## UNSCratch

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at FULL service level

## Description

The UNSCratch command unscratches one or more volumes.

## Syntax

Figure 3–178 shows UNSCratch command syntax.

**Figure 3–178 UNSCratch command syntax**



## Parameters

As shown in Figure 3–178, the UNSCratch command includes the following parameters:

### **VOLser(vol-list)**

specifies the list of volume serial numbers to be added, deleted, or replaced in the scratch list(s).

*vol-list* indicates the volume serial numbers; this can be a single volume, a list of volume serial numbers, ranges of volume serial numbers, or combinations of lists with ranges delimited by commas. You can specify a maximum of 100 volume serial numbers. The entire list must be enclosed in parentheses.

## UNSElect

### **Interfaces:**

- SLUADMIN utility only
- UUI Support: No

### **Subsystem Requirements:**

Active HSC not required

## Description

The UNSElect command unselects a volume that is erroneously in a selected state.

**Note:**

- Use this utility only if you are certain that the HSC has incorrectly left the volume selected. Incorrect use of this utility can result in HSC abends and errant volumes.

Issue a `Display Volume DEtail` command to determine which host has the volume in question. Then, issue `Display DRives` and `Display Requests` commands on the host that has the volume in question to see if that volume is being used.

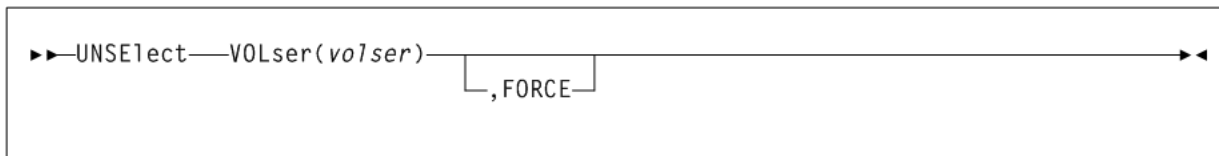
If the selected volume is mounted on a transport, issue a `DISMount` command for the transport. If there is a request active to the LMU for the volume, wait for the request to complete. If the overdue response handler indicates the request has timed out, you may want to abort the request.

- The `UNSElect` function can only be invoked when `SLUADMIN` is invoked APF authorized, key 0 through 7, or supervisor state. The load module invoked by `UNSElect` is `SLUNSEL`. Restricting access to this module restricts access to this utility function.

**Syntax**

Figure 3–179 shows `UNSElect` command syntax.

**Figure 3–179 UNSElect command syntax**

**Parameters**

As shown in Figure 3–179, the `UNSElect` command includes the following parameters:

**VOLser(volser)**

indicates that a `VOLSER` is being specified for unselection.

`volser` indicates the `VOLSER` to be unselected.

**,FORCE**

optionally, indicates that the volume is to be unselected even if communication with the host that has the volume selected is not possible.

**Vary****Interfaces:**

- Console or `PARMLIB` (Vary `ACS`)
- Console or utility, `UUI All` (Vary `CLINK`, `RTD`, or `VTSS`)
- `UUI Support`: Yes

**Subsystem Requirements:**



- Active HSC at FULL service level (Vary ACS)
- Active HSC/VTCS (Vary CLINK, RTD, or VTSS)

## Description

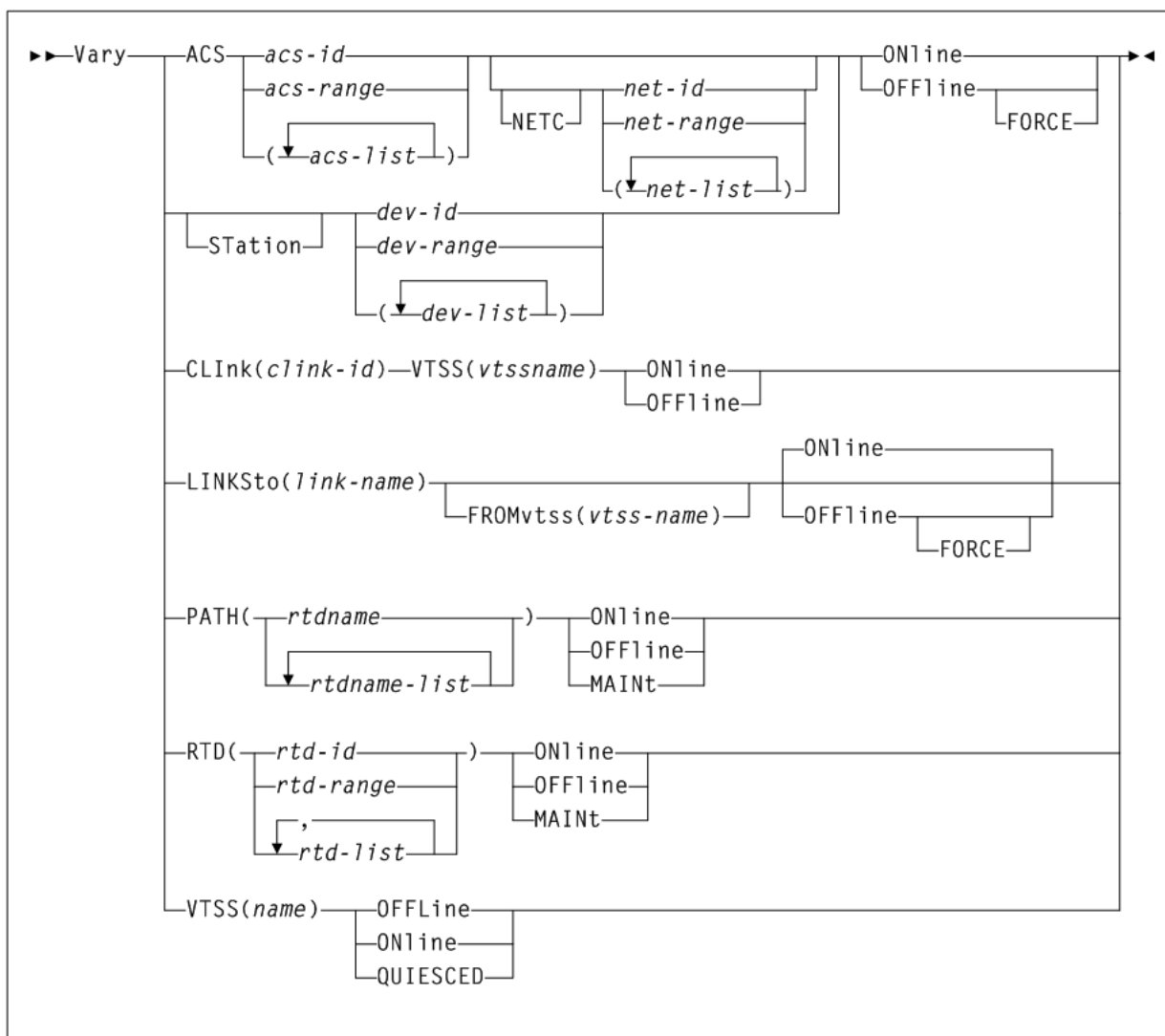
The Vary command changes the online/offline state for the following:

- ACS or station
- CLINK
- PATH
- RTD
- VTSS

## Syntax

Figure 3–180 shows Vary command syntax.

**Figure 3–180** Vary command syntax



## Parameters

As shown in Figure 3–180, the `Vary` command includes the following parameters:

### **ACS(*acs-id*, *acs-range*, or *acs-list*)**

optionally, specifies the ACS to be made online, offline, or standby to this host.

*acs-id*, *acs-range*, or *acs-list* indicates one or more ACSs to be varied. Each *acs-list* element can be either a single ACSid or an ACSid range. Ranges are separated by a dash. The elements in a list must be separated by commas or blanks, and the entire list must be enclosed in parentheses.

### **NETC(*net-id*, *net-range*, or *net-list*)**

optionally, used in conjunction with ACS *acs-id* to vary online/offline a single, range, or list of HSC library network connections.

*net-id*, *net-range*, or *net-list* indicates one or more network connection to be varied. Each *net-list* element can be either a single *net-id* or a *net-id* range. Ranges are separated by a dash. The elements in a list must be separated by commas or blanks, and the entire list must be enclosed in parentheses.

### **ONline**

specifies that the stations are to be made online or standby to this host.

### **OFFline**

specifies that the stations are to be made offline to this host.

### **FORCE**

optionally, indicates that the stations are to be made offline immediately. The `FORCE` option can be specified when the HSC is at the `FULL` service level or is past the `BASE` service level going to the `FULL` service level.

This parameter applies only to the `OFFline` operand.

### **Station(*dev-id*, *dev-range*, or *dev-list*)**

optionally, specifies the stations to be made online, offline, or standby to this host.

In an ACS, a connection between the host CPU and an LMU is referred to as a “station.” Each station appears to the host CPU as a 3278-2 device and is physically connected to a port on a supported 3174, 3274 or compatible terminal controller. Each LMU can contain a maximum of 16 stations. Depending on the number of host CPUs connected to the ACS, each host CPU can have either one or several stations to each LMU.

*dev-id*, *dev-range*, or *dev-list* indicates the stations to an LMU to be varied. Each *dev-list* element can be either a single device number or a range of device numbers. Ranges are separated by a dash. The elements in a list must be separated by commas or blanks, and the entire list must be enclosed in parentheses.

Each device number identifies a device attached to the host issuing the command. Allowable values are 000-FFF.

### **CLink(*clink-id*)**

the specified CLINK.

*clink-id* indicates the link ID.

### **VTSS(*vtssname*)**

the sending VTSS in the Cluster.

*vtssname* indicates the 1 to 8 character identifier of the sending VTSS.

**ONLine**

specifies that the stations are to be made online or standby to this host.

**OFFLine**

Vary the specified CLINK offline.

**LINKSto(link-name)**

specifies a replication link used for enhanced synchronous replication, to be varied into a desired state.

*link-name* indicates the name of the replication link that you want to vary to the specified state. This name represents the target VTSS subsystem name.

**FROMvtss(vtss-name)**

optionally, specifies the *vtss-name* that owns, or is the source VTSS, for any specified link-name. If not specified, all replication links with the specified name are varied into the specified state.

**ONLine**

Vary the specified replication link into an online state.

**OFFLine**

Vary the specified replication link into an offline state. Active and pending replications will complete before the link is taken offline.

Additionally, specify `FORCE` to terminate all active and pending replications for the specified replication link.

---

**Note:** You must specify either `ONLine` or `OFFLine`. These parameters are mutually exclusive.

---

**PATH(rtd-name or rtdname-list)**

optionally, specifies the path (from a VTSS to RTDs) to be made online, offline, or standby to this host. This permits the individual `RTDPATH` from a VTSS to a device to be varied to the desired state. This path may represent either a direct connection to a RTD from a VTSS or a connection to a Virtual Library.

*rtd-name* or *rtdname-list* indicates the names assigned to the paths for accessing RTDs for Virtual Libraries from a VTSS

**ONLine**

Vary the specified path (from a VTSS to RTDs) to an online state.

**OFFLine**

Vary the specified path (from a VTSS to RTDs) to an offline state.

**MAINT**

Vary the specified path (from a VTSS to RTDs) to an offline (maintenance mode) state.

**RTD(rtd-name, rtd-list, or rtd-range)**

Change the state of the specified RTDs.

*rtd-name*, *rtd-range*, or *rtd-list* indicates 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.

**VTSS(name)**

Change the state of the specified VTSS.

*name* indicates the VTSS identifier.

---

---

**Note:** Vary VTSS does not change the state of the VTDs or RTDs associated with the specified VTSS.

---

---

**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.

---

---

## View

**Interfaces:**

- Console or PARMLIB only
- UUI Support: No

**Subsystem Requirements:**

Active HSC at FULL service level

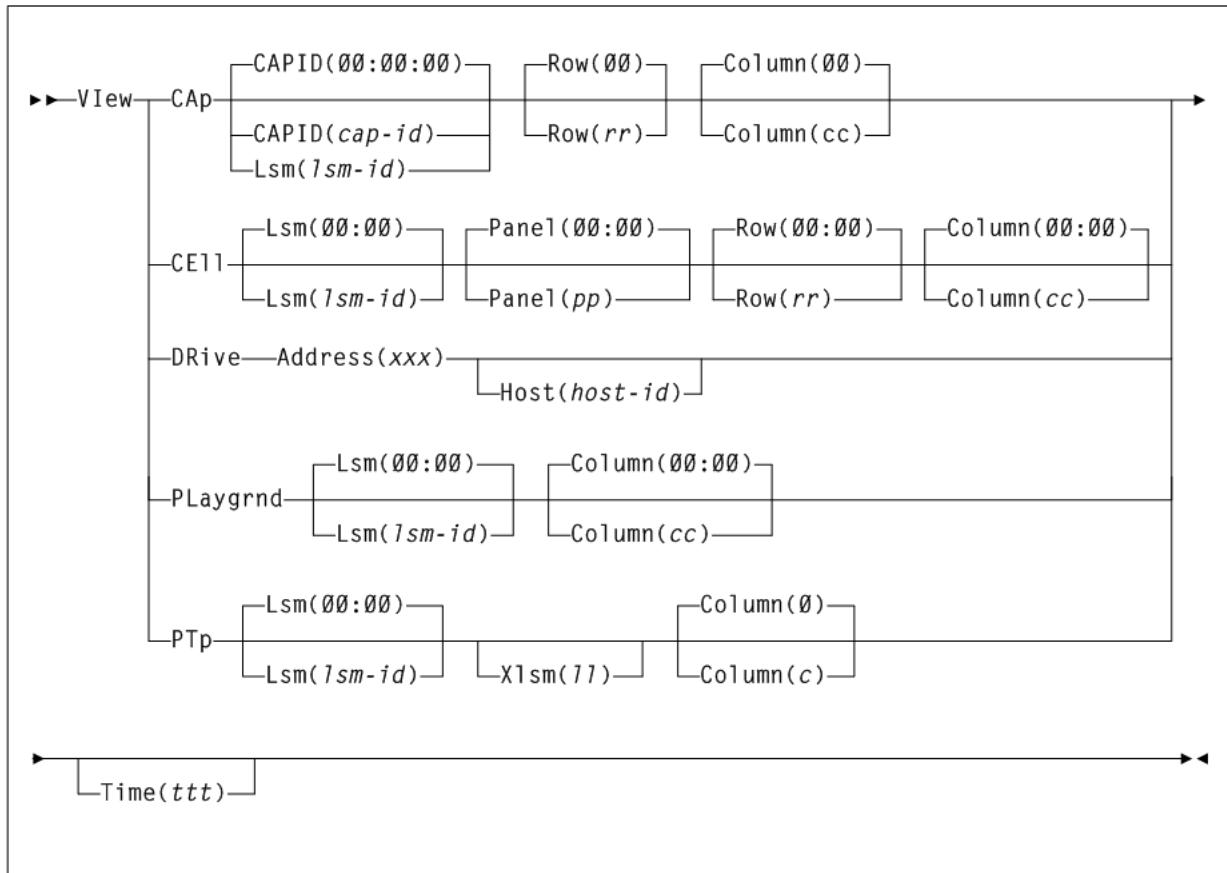
## Description

The `VIEW` command enables you to see specified internal components of the LSM when video monitors are attached to the LSM.

## Syntax

Figure 3–181 shows `VIEW` command syntax.

Figure 3–181 View command syntax



## Parameters

As shown in Figure 3–181, the `VIEW` command includes the following parameters:

### CAP

indicates that a camera is to focus on a cell location in a CAP. If you do not specify the CAPID or `Lsm` parameter, the command defaults to CAPid 00:00:00. If you do not specify the `Row` and `Column` parameters, the command defaults to row 0, column 0.

#### CAPID(cap-id)

identifies one CAP in an LSM that contains multiple CAPs.

`cap-id` indicates the CAP that you want to view. The format of a `cap-id` is `AA:LL:CC`, where `AA:LL` indicates the LSMid, and `CC` indicates the CAP. See "CAPid" on page B-1 for a list of valid values.

#### Lsm(lsm-id)

identifies an LSM that contains a single CAP.

`lsm-id` indicates the single CAP that you want to view. The format of an LSMid is `AA:LL`, where `AA` indicates the ACSid (decimal 00-99) and `LL` indicates the LSM number (decimal 00-99).

#### Row(rr)

identifies a row number in a CAP. If you do not specify this parameter the command defaults to row 0.

*rr* indicates the row number. Allowable values for *rr* are decimal and are specific to the LSM type. See "Rows" on page B-3 for a list of valid values.

**Column(cc)**

identifies the column number within a CAP. If you do not specify this parameter the command defaults to column 0.

*cc* indicates the column number. Allowable values for *cc* are decimal and are specific to the LSM type. Columns are numbered starting at the left of the panel. See "Columns" on page B-4 for a list of valid values.

**CEII**

indicates that a camera is to focus on a cartridge storage or diagnostic cell in an LSM panel. If you do not specify the *Lsm*, *Panel*, *Row*, and *Column* parameters, the command defaults to *LSM 00:00*, *panel 0*, *row 0*, *column 0*.

**Lsm(lsm-id)**

identifies an LSM. If you do not specify this parameter, the command defaults to *LSMid 00:00*.

*lsm-id* indicates the *LSMid*. The format of an *LSMid* is *AA:LL*, where *AA* indicates the *ACSid* (decimal 00-99) and *LL* indicates the LSM number (decimal 00-99).

**Panel(pp)**

designates the LSM panel number containing the cartridge storage or diagnostic cell that the camera is to view. If you do not specify this parameter the command defaults to panel 0.

*pp* indicates the panel number. Allowable values for *pp* are decimal and are specific to the LSM type. See "Panels" on page B-2 for a list of valid values.

**Row(rr)**

identifies a row number in an LSM panel. If you do not specify this parameter the command defaults to row 0.

*rr* indicates the row number. Allowable values for *rr* are decimal and are specific to the LSM type. See "Rows" on page B-3 for a list of valid values.

Column 3 on panel 2 allows row entries only on rows 28-41. The cells on panel 3 are optional.

**Column(cc)**

identifies the column number within a panel. If you do not specify this parameter the command defaults to column 0.

*cc* indicates the column number. Allowable values for *cc* are decimal and are specific to the LSM type. Columns are numbered starting at the left of the panel. See "Columns" on page B-4 for a list of valid values.

**DRive**

indicates that a camera is to focus on a cartridge drive. You must designate the *Address* parameter to identify the drive. The *Host* parameter enables you to view a drive defined to another host.

**Address(xxx)**

specifies the address of a cartridge drive. The *Address* parameter defines the operating system address of the transport you want to inspect.

*xxx* indicates the hexadecimal address of the transport. The HSC verifies that the specified address matches the LIBGEN-defined address for the given host.

**Host(host-id)**

gives meaning to the Address parameter when the drive being inspected is not defined in the LIBGEN for the host from which the VIEW command is being entered.

If you do not specify the Host parameter, the HSC checks the LIBGEN-defined drive list for the host from which you entered the VIEW command. If the address is found in the drive list, the command is executed.

*host-id* indicates the host where the transport is defined.

**PLaygrnd**

indicates that a camera is to focus on a playground cell. If the Lsm and Column parameters are not specified, the command defaults to the relative location of the cell in the playground in LSM 00:00.

---

**Note:** The playground is a reserved area of cells where the robot deposits cartridges that it finds in its hands during LSM initialization. Normal LSM initialization recovery processing moves cartridges from the playground cells to either their home cells or their intended destinations, but under abnormal circumstances cartridges may be left in playground cells. Refer to the appropriate Oracle StorageTek hardware publication for the location of the playground in the LSM.

---

**Lsm(lsm-id)**

identifies an LSM. If you do not specify this parameter, the command defaults to LSMid 00:00.

*lsm-id* indicates the LSMid. The format of an LSMid is *AA:LL*, where *AA* indicates the ACSid (decimal 00-99) and *LL* indicates the LSM number (decimal 00-99).

**Column(cc)**

identifies the relative location of the cell in the playground for a panel. If you do not specify this parameter, the command defaults to the relative location of the cell in the playground.

*cc* indicates the relative location of the cell in the playground. Allowable values for *cc* are decimal and are specific to the LSM type. (Some LSMs have multiple playground cells and some have only one.)

**PTp**

indicates that a camera is to focus on a pass-thru port (PTP) cell. You can designate a particular PTP using the Lsm and Xlsm parameters, a particular cell using the Column parameter, and specify the viewing time using the Time parameter.

---

**Note:** When the PTP is being shared by two different LSM types, the upper limit of the column value is determined by the LSM with the smaller PTP capacity.

---

**Lsm(lsm-id)**

identifies an LSM. If you do not specify this parameter, the command defaults to LSMid 00:00.

*lsm-id* indicates the LSMid. The format of an LSMid is *AA:LL*, where *AA* indicates the ACSid (decimal 00-99) and *LL* indicates the LSM number (decimal 00-99).

**Xlsm(l)**

identifies the PTP to be inspected by defining the LSM that shares the PTP.

Each PTP is identified by the two LSMs it services. The `Lsm` parameter identifies which robot is used to view the PTP, and the `Xlsm` parameter identifies the LSM that shares the PTP. This distinction is necessary whenever an LSM contains more than one PTP.

If `Xlsm` is not specified for an LSM containing two or more PTPs, the HSC examines the `LIBGEN` and selects the first PTP defined there for `Lsm(lsm-id)`.

`ll` indicates the LSM number of the adjacent LSM. Values for `ll` are decimal in the range from 00-99. (The ACS is identified in the `Lsm` parameter.)

**Column(c)**

identifies the column number within a panel. If you do not specify this parameter the command defaults to `column 0`.

`c` indicates the column number. Allowable values for `c` are decimal and are specific to the LSM type:

- 0-3 for LSM Models 4410, 9310, and 9740
- 0 or 1 for WolfCreek LSMs (all models).

---

---

**Note:** When the PTP is being shared by two different LSM types, the upper limit of the column value is determined by the LSM with the smaller PTP capacity.

---

---

**Time(ttt)**

specifies the video monitor viewing time.

`ttt` indicates the time, in seconds.

## VLEMAINT

**Interfaces:**

- Console or utility
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC/VTCS

## Description

The `VLEMAINT` utility command moves or copies VLE based MVC (VMVC) data content to a specified VLE system, without using VTSS storage resources.

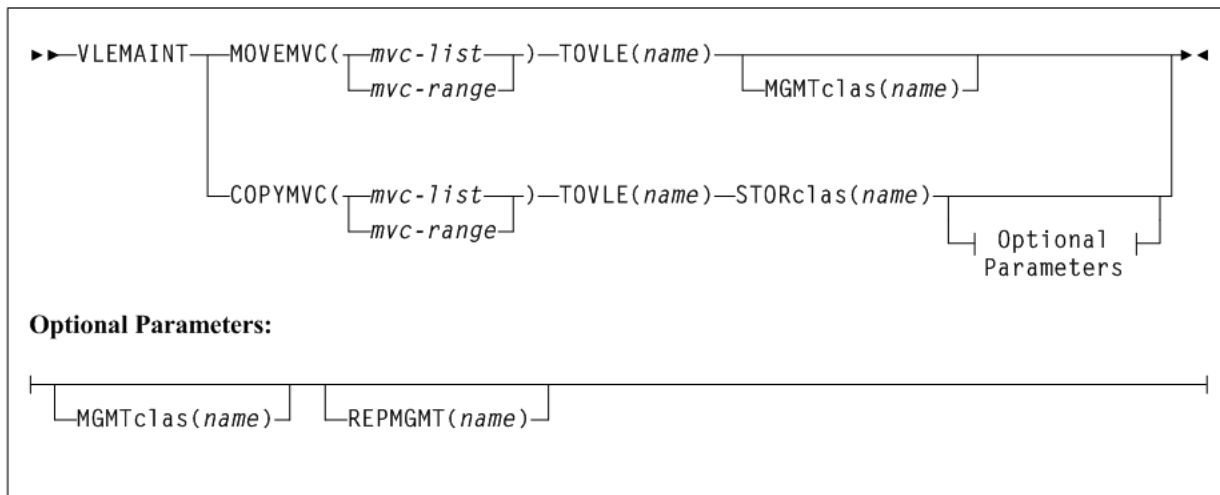
This utility should only be used under the guidance of Oracle Support personnel.

## Syntax

Figure 3–182 shows `VLEMAINT` command syntax.



Figure 3–182 VLEMAINT command syntax



## Parameters

As shown in Figure 3–182, the VLEMAINT command includes the following parameters:

### MOVEMVC(*mvc-list* or *mvc-range*)

specifies a list or range of VLE based MVCs whose content is to be moved to the specified target VLE system. MVCs selected in the target VLE system inherit the storage class of the source MVC.

*mvc-list* or *mvc-range* indicates the list or range of MVCs.

MOVEMVC is mutually exclusive with COPYMVC.

### TOVLE(*name*)

specifies the target VLE to receive the source VMVC data. This parameter is required.

*name* indicates the target VLE name. This VLE must be valid (defined to the configuration). Otherwise, the utility aborts with a return code 8.

### MGMTclas(*name*)

optionally, specifies a single management class used to filter data sent to the target VLE. Only VTVs with this management class are sent to the target VLE.

*name* indicates the management class name. This name must be valid (defined in MGMTDEF). Otherwise, the utility aborts with return code 8.

### COPYMVC(*mvc-list* or *mvc-range*)

specifies a list or range of VLE based MVCs whose content is to be copied to the specified target VLE system. COPYMVC creates an additional copy of the data to another storage class on the target VLE system.

*mvc-list* or *mvc-range* indicates the list or range of MVCs.

COPYMVC is mutually exclusive with MOVEMVC. COPYMVC requires a STORclas name to be specified.

### TOVLE(*name*)

specifies the target VLE to receive the source VMVC data. This parameter is required.

*name* indicates the target VLE name. This VLE must be valid (defined to the configuration). Otherwise, the utility aborts with a return code 8.

**STORclas(name)**

optionally, specifies the storage class assigned to the VMVC selected at the target VLE for COPYMVC processing. This parameter is required when COPYMVC is specified.

*name* indicates the storage class name. This name must be valid (defined in MGMTDEF). Otherwise, the utility aborts with a return code 8.

**Optional Parameters (VLEMAINT COPYMVC)**

If you specify VLEMAINT COPYMVC, the following optional parameters apply:

**MGMTclas(name)**

optionally, specifies a single management class used to filter data sent to the target VLE. Only VTVs with this management class are sent to the target VLE.

*name* indicates the management class name. This name must be valid (defined in MGMTDEF). Otherwise, the utility aborts with return code 8.

**REPMGMT(name)**

optionally, specifies a single management class name assigned to the VTV data after it is received by the target VLE system for COPYMVC processing.

*name* indicates the management class name. This name must be valid (defined in MGMTDEF). Otherwise, the utility aborts with return code 8.

---

---

**Note:** During COPYMVC processing, any VTV found with the management class specified in REPMGMT is bypassed. This could potentially serve as a copy complete acknowledgment if utility restarts occur.

---

---

## VOLPCONV

**Interfaces:**

- SLUADMIN utility only
- UUI Support: No

**Subsystem Requirements:**

Active HSC not required

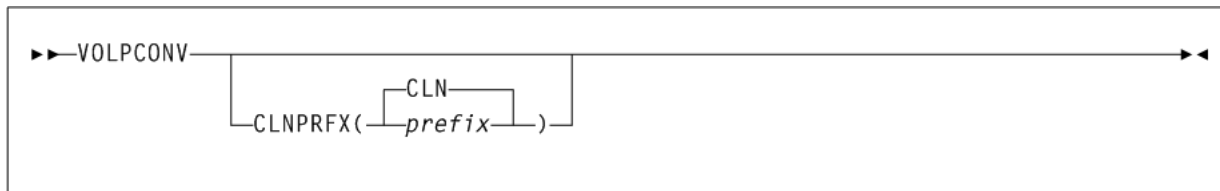
## Description

The VOLPCONV command reads existing VOLDEF, SCRDEF, MVCDEF, and VTCS CONFIG input statements and converts them to SET VOLPARAM statements. The existing VTCS configuration is also input to the utility.

## Syntax

Figure 3–183 shows VOLPCONV command syntax.

Figure 3–183 VOLPCONV command syntax



## Parameters

As shown in [Figure 3–183](#), the VOLPCONV command includes the following parameters:

### CLNPRFX(prefix or CLN)

specifies a prefix to apply to all types of cleaning cartridges.

*prefix* indicates a three-character prefix. CLN is the default.

## Additional JCL Requirements

In addition to the required JCL definition statements described in "[Invoking SLUADMIN](#)" on page 5-1, the following definition statements apply to the VOLPCONV JCL:

### SLSVOLA

input VOLDEF definitions in the form of VOLATTR statements.

---

**Note:** VOLATTR statements coded with the masking character (% , ? or \*) within the SERIAL parameter are not supported and will cause message SLS0227I to be issued. Any VOLATTR statement using these masking characters must be converted to a range that spans the intent of the masking characters prior to running VOLPCONV.

---

### SLSSCRP

input SCRPEDEF definitions in the form of SCRPOOL statements.

### SLSMVCP

input MVCDEF definitions in the form of MVCPOOL statements.

### SLSVTCS

input VTCS configuration definitions. The VTCS configuration definitions can be obtained by running the VTCS DEComp utility.

### SLSPARM

output VOLUME or POOL definitions in the form of POOLPARM and VOLPARM control statements. The logical record size of the SLSPARM data set is 80.

## VOLRpt

### Interfaces:

- Utility only
- UUI Support: Yes

### Subsystem Requirements:

Active HSC not required

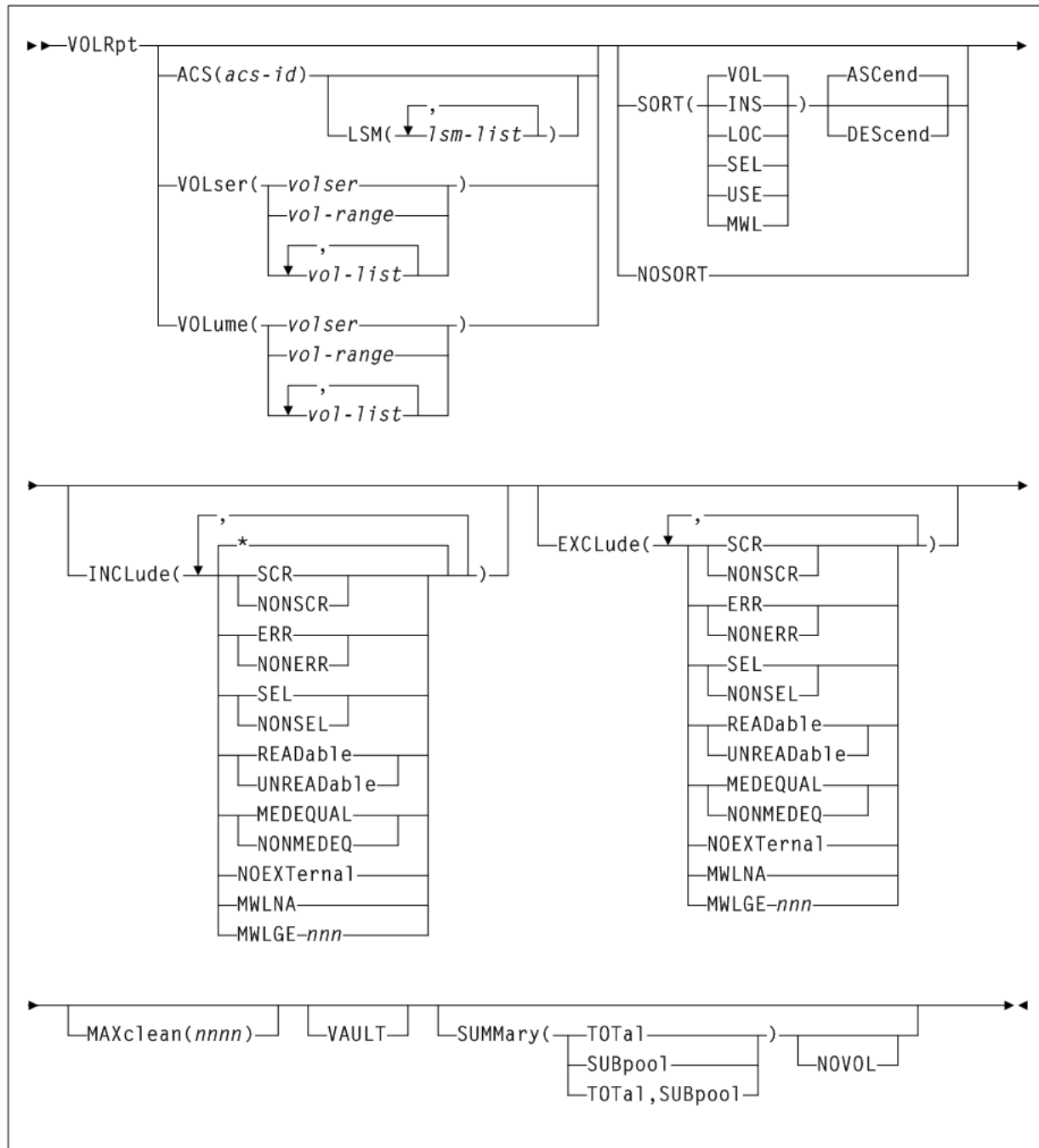
## Description

The VOLRpt command produces a volume report.

## Syntax

Figure 3–184 shows VOLRpt command syntax.

Figure 3–184 VOLRpt command syntax



## Parameters

As shown in [Figure 3-184](#), the VOLRpt command includes the following parameters:

### **ACS(acs-id)**

optionally, specifies that a report is to be produced for only a particular ACS.

*acs-id* indicates the one or two digit decimal ACS identifier.

### **LSM(lsm-list)**

optionally, specifies that a report is to be produced for only certain LSMs within an ACS.

*lsm-list* indicates the LSMs. An *lsm-list* can be a single LSM number or a list of LSM numbers. An LSM number is decimal 00-99.

An LSM range is not allowed. If a list is specified, the elements must be separated by blanks or commas, and the entire list enclosed in parentheses.

### **VOLser or VOLume(volser, vol-range, or vol-list)**

optionally, specifies that the report only contain information on certain VOLSERS.

*volser*, *vol-range*, or *vol-list* indicates the volume serial numbers requested. Any subranges of volumes specified in the *vol-list* that are not in the control data set are listed in the Control Card Image Listing portion of the report using one line per subrange.

A percent sign (“%”) may be used as a “wildcard” character in the VOLSER to specify pattern matching. The percent sign designates that any single character can match the corresponding position in the VOLSER.

For example, A9%%%% specifies that all of the six-character volume serial numbers that begin with the characters “A9” are selected for the report.

Q%12% specifies that all five-character VOLSERS that begin with “Q” and have a “12” in the third and fourth positions of the VOLSER are selected for the report. The percent sign cannot be specified in a range specification. Therefore, “A%0000-A%9999” is invalid.

In the Volume Report Listing of the utility, the volumes requested which are not in the control data set are not listed.

### **SORT**

optionally specifies a specified sort sequence.

Multiple sort criteria may be specified. The order, from left to right, specifies the order in which the report is to be sorted.

For example, SORT (INS, USE) produces a report sorted by date inserted in the control data set, and then for each date, sorted by select count.

---



---

**Note:** SORT and NOSORT are mutually exclusive.

---



---

The following SORT options are available:

### **VOL**

indicates that the report is to be sorted by volume serial number (VOL is the default).

**INS**

indicates that the report is to be sorted by date and time the volume was inserted into the control data set.

**LOC**

indicates that the report is to be sorted by location.

**SEL**

indicates that the report is to be sorted by date and time the volume was last selected.

**USE**

indicates that the report is to be sorted by selection count.

**MWL**

indicates that the report is to be sorted by Media Warranty Life (MWL) percentage, which indicates how much of the media life has been used. Media warranty life is considered to be expired at 100%.

To collect and report media warranty life, tape libraries and transports must meet the following requirements:

- SL8500 or SL3000 libraries
- LMU compatibility level 21 or higher
- T9x40: all media and models at firmware level 1.42 or higher (except 9840B)
- T10000: all models and media at firmware level 1.38 or higher

Refer to Oracle's ELS publication *Managing HSC and VTCS* for more information about media warranty life.

**ASCend**

optionally, specifies that the report is to be sorted in ascending order. ASCend is the default. This parameter is ignored if NOSORT is specified. ASCend is mutually exclusive with NOSORT and DEScend.

**DEScend**

optionally, specifies that the report is to be sorted in descending order. This parameter is ignored if NOSORT is specified. DEScend is mutually exclusive with NOSORT and ASCend.

**NOSORT**

specifies that an unsorted report is to be produced. NOSORT is mutually exclusive with SORT, ASCend, and DEScend.

**INCLude**

optionally, specifies the criteria for including volume information in the report. If this keyword is specified, all volumes that match at least one of the specified criteria are tentatively selected for the report. Information about volumes may be removed from the list of volumes selected for the report if other options, such as EXCLUDE, VOL, ACS, or LSM are specified.

- INCLude parameters are applied before EXCLude parameters in volume selection.
- Parameter values are not checked for the existence of their opposite values (for example, SEL is still flagged even if NONSEL is specified, and vice versa).
- Positive attributes are applied before negative attributes (for example, SCR is applied before NONSCR).

\*

indicates that all volumes in the library are considered for being included in the report. The default is \*. If more than one of the following parameters is specified, the parameters must be separated by commas.

**SCR**

specifies that scratch volumes match the specified criteria.

**NONSCR**

specifies that nonscratch volumes match the specified criteria.

**ERR**

specifies that errant volumes match the criteria.

**NONERR**

specifies that non-errant volumes match the criteria.

**SEL**

specifies that selected volumes match the criteria.

**NONSEL**

specifies that nonselected volumes match the criteria.

**READable**

specifies that volumes with a readable external label match the criteria.

**UNREADable**

specifies that volumes with an unreadable external label match the criteria.

**MEDEQUAL**

specifies that volumes for which the media types of the VOLATTR and the VAR are equal match the criteria.

**NONMEDEQ**

specifies that volumes for which the media types of the VOLATTR and the VAR are not equal match the criteria.

**NOEXternal**

specifies that volumes without an external label match the criteria.

For example, INCLUDE (SEL, ERR) tentatively chooses only selected and errant volumes for the report. INCLUDE parameters are applied before EXCLUDE parameters in volume selection.

**MWLNA**

specifies that volumes with an unknown media warranty life match the criteria. Media warranty life is obtained at volume dismount for tape libraries with an LMU compatibility level of 21 or above. Use the Display ACS command to obtain the LMU compatibility level.

**MWLGE nnn**

specifies that volumes with a media warranty life greater than or equal to *nnn* percent match the criteria. It also excludes all volumes with an unknown media warranty life.

Specify INCLUDE (MWLNA) to force inclusion of all volumes with an unknown media warranty life. Valid values for nnn are 0-254.

To collect and report media warranty life, tape libraries and transports must meet the following requirements:

- SL8500 or SL3000 libraries
- LMU compatibility level 21 or higher
- T9x40: all media and models at firmware level 1.42 or higher (except 9840B)
- T10000: all models and media at firmware level 1.38 or higher

Refer to Oracle's ELS publication *Managing HSC and VTCS* for more information about media warranty life.

**EXCLude**

optionally, specifies the criteria for excluding volume information from the report. Any volumes that match one or more of the exclusion criteria are excluded from the report.

EXCLude parameter values are the same as INCLude values, except for the \* parameter (see the INCLude options list above).

As an example, EXCL(NONSEL) excludes nonselected volumes from the report.

- INCLude parameters are applied before EXCLude parameters in volume selection.
- If MWLGE is specified in both the INCLUDE and EXCLUDE parameters, the EXCLUDE limit value must be greater than the INCLUDE limit value.
- Parameter values are not checked for the existence of their opposite values (for example, SEL is still flagged even if NONSEL is specified, and vice versa).
- Positive attributes are applied before negative attributes (for example, SCR is applied before NONSCR).

\*

indicates that all volumes in the library are considered for being included in the report. The default is \*. If more than one of the following parameters is specified, the parameters must be separated by commas.

**SCR**

specifies that scratch volumes match the specified criteria.

**NONSCR**

specifies that nonscratch volumes match the specified criteria.

**ERR**

specifies that errant volumes match the criteria.

**NONERR**

specifies that non-errant volumes match the criteria.

**SEL**

specifies that selected volumes match the criteria.

**NONSEL**

specifies that nonselected volumes match the criteria.

**READable**

specifies that volumes with a readable external label match the criteria.

**UNREADable**

specifies that volumes with an unreadable external label match the criteria.



**MEDEQUAL**

specifies that volumes for which the media types of the VOLATTR and the VAR are equal match the criteria.

**NONMEDEQ**

specifies that volumes for which the media types of the VOLATTR and the VAR are not equal match the criteria.

**NOEXternal**

specifies that volumes without an external label match the criteria.

For example, INCLUDE (SEL, ERR) tentatively chooses only selected and errant volumes for the report. INCLUDE parameters are applied before EXCLUDE parameters in volume selection.

**MWLNA**

specifies that volumes with an unknown media warranty life match the criteria. Media warranty life is obtained at volume dismount for tape libraries with an LMU compatibility level of 21 or above. Use the Display ACS command to obtain the LMU compatibility level.

**MWLGE nnn**

specifies that volumes with a media warranty life greater than or equal to *nnn* percent match the criteria. It also excludes all volumes with an unknown media warranty life.

Specify INCLUDE (MWLNA) to force inclusion of all volumes with an unknown media warranty life. Valid values for *nnn* are 0-254.

To collect and report media warranty life, tape libraries and transports must meet the following requirements:

- SL8500 or SL3000 libraries
- LMU compatibility level 21 or higher
- T9x40: all media and models at firmware level 1.42 or higher (except 9840B)
- T10000: all models and media at firmware level 1.38 or higher

Refer to Oracle's ELS publication *Managing HSC and VTCS* for more information about media warranty life.

**MAXclean(nnnn)**

optionally, specifies the maximum cleaning cartridge usage for the "over max clean" column. The MAXclean specified for each POOLPARM TYPE (CLEAN) overrides this value.

*nnnn* indicates the maximum usage value. Valid values are 0-32767.

**VAULT**

optionally, specifies that vaulted volumes are to be included in the detail volume report. VAULT is mutually exclusive with NOVOL, ACS, and LSM.

If the vault CDS subfile does not exist, this parameter is ignored.

Vaulted volumes are not included in the SUMMARY reports. In the detail report, the cell location for a vaulted volume is identified by the word VAULT and the insertion date/time is the date/time that the volume was inserted into the vault; i.e., ejected from the tape library. Refer to the *LCM User's Guide* for more information about vaulting.

**SUMMARY**

optionally, specifies that the utility provide totals of volume attributes on an LSM and ACS basis, and/or subpool data on an ACS or LSM basis. Totals are affected by the use of limiting parameters such as ACS (and possibly LSM), VOLser or VOLume, and INCLude and/or EXCLude.

If both TOTal and SUBpool are specified, both reports are provided. The time and date displayed in the header for the Summary Report(s) are the same as the Volume Report Listing header.

**TOTAL**

specifies that totals of scratch, selected, errant, available cells, and external label status types be provided on an LSM, ACS, and library basis in the report.

The totals are listed on a separate listing, on a new page from the Volume Report Listing and the Control Card Listing.

**SUBpool**

specifies that subpool totals be provided on an LSM, ACS, and library basis in the report.

Subpool information can be provided using the VOLPARM facility. See "[VOLPARM Control Statement](#)" on page 3-348 for more information about defining VOLPARM information. For alternative methods of specifying subpool information, refer to Oracle's ELS publication *ELS Legacy Interfaces Reference*.

The totals are listed on a separate listing, on a new page from the Volume Report Listing, the Control Card Listing, and the Volume Report Totals Listing.

If subpools overlap (that is, a volume belongs to more than one subpool), then each volume is reported only within the first subpool in which it occurs, and totals for the other subpools may be inaccurate.

**NOVOL**

optionally, when used with SUMMARY, specifies to display summary and/or subpool totals without producing volume detail.

NOVOL should not be specified by itself. NOVOL is mutually exclusive with ACS, VOLser, VOLume, SORT, NOSORT, INCLude, EXCLude, MAXclean, and VAULT.

**Additional JCL Requirements**

If you are not using the VOLPARM facility to define your volume characteristics, refer to Oracle's ELS publication *ELS Legacy Interfaces Reference* for a description of additional VOLRPT JCL statements that may be required.

**Volume Report**

The following is an example of a volume report produced by the VOLRpt command issued with the SUMMARY(SUBpool) parameter:

**Example 3-61 VOLRpt report**

```

Volume                               Cell Loc   Err   Sel Ext Cln           |--- Inserted --| |-- Last
Used --|   Times  MWL
Serial Media  Rectech AA:LL:Pa:Ro:Co | Scr | Lbl Use  Subpool ID   Date    Time    Date
Time Selected %
B00141 T10000T1 T1B35   01:01:02:08:01   Y    R    SMCT10K     20110725 15:52:13 20110920
10:44:52         13   0

```

## Subpool Totals, all Ranges

Subpool ID	Label Type	Range Limits	DRTEST?
DEFAULTPOOL	SL	N/A - N/A	No

**Fields**

The VOLRpt report includes the following fields:

**Volume Serial**

the volume serial number

**Media**

the media type. See [Appendix A, "MEDia, RECtech, and MODEL Values"](#).

The media field may be immediately preceded by an asterisk \* or dash -.

- An asterisk \* indicates that the media defined by a VOLATTR statement is not compatible with the media label value.
- A dash - indicates that the volume has no VOLATTR or VOLPARM defined media value.

**Rectech**

the recording technique. See [Appendix A, "MEDia, RECtech, and MODEL Values"](#).

**Cell Loc**

the cell location field shows cartridge location by ACS (AA), LSM(LL), panel (PA), row (RO), and column (CO).

**Err, Scr, Sel**

A Y (yes) flag under the **Errant**, **Scratch**, or **Selected** headings indicates that the volume is currently in that status.

A volume in Errant (Err) status shows the volume's home cell in the **Cell Loc** column.

A Y (yes) flag under the **Selected** heading indicates that the volume is currently selected, that is, being mounted, dismounted, ejected, or moved. An M flag indicates that the volume is currently mounted. The **Selected** column on the totals report includes both selected and mounted volumes. The volume's home cell appears in the **Cell Loc** column.

**Ext Lbl**

External Label:

- R indicates that the volume has a readable external label.
- U indicates that the volume has an unreadable external label.
- N indicates that the volume has no external label.

**CIn Use**

Cartridge Usability:

- N indicates that the volume is not usable, for example, a spent cleaning cartridge that is not usable because it has exhausted its cleaning surface.
- M indicates that a cleaning cartridge has exceeded its maximum usage set by the MAXclean value (as specified by the MNTD MAXclean or VOLATTR MAXclean parameters).

**Subpool ID**

the subpool ID.

Scratch subpools for volumes are listed if scratch subpool definitions have been provided using VOLPARM control statements.

- \*NON-VOLPARM\* indicates that VOLPARM is in effect, but this volume is not defined in VOLPARM.
- DEFAULTPOOL indicates that the volume is defined in VOLPARM with no associated pool.
- \*MVC\* indicates that the volume is reserved for VTCS usage as a migration volume.
- \*DEFAULT\* indicates that the volume is a cleaning cartridge but has not been included in a VOLPARM defined subpool, \*DEFAULT\* appears.

If VOLPARM is not used, the generated report is the same as in the HSC 6.2 release. Refer to the *HSC 6.2 System Programmer's Guide* for more information.

**Inserted**

date and time that the volume was inserted.

**Last Used**

date and time that the volume was last used

**MWL Media Warranty Life**

Media Warranty Life (MWL) is a percentage value that is obtained at volume dismount for tape libraries that are at LMU compatibility level 21 or later. If you do not know the compatibility level, enter the HSC `Display Acs` command to display it. See "[Display ACS](#)" on page 3-74 for more information.

The MWL percent indicates how much of the media life has been used and is considered to be "expired" at 100%. Values may range from 0 to 254. If the field is blank, the MWL is unknown.

**Total Reports**

Two totals reports can be produced:

- volume report totals
- subpool totals

**Volume Report Totals**

Total of cartridges with various characteristics are displayed for LSMs, ACSs, and the entire library.

The total selected volumes is the sum of the number of volumes currently selected and the number of volumes currently mounted.

---

---

**Note:** The free storage cells reported do not include free cells on frozen panels. An LSM with at least one frozen panel is indicated by an asterisk after the free cell count.

---

---

**Subpool Totals**

Totals of cartridges by subpool are produced for LSMs, ACSs, and the entire library.

---



---

**Note:** A value of Y in the **DRTEST** column indicates that the pool is defined using the VOLPARM control statement feature with the keyword DRTEST(YES).

---



---

## VTVMaint

### Interfaces:

- Utility only
- UUI Support: Yes

### Subsystem Requirements:

Active HSC not required

## Description

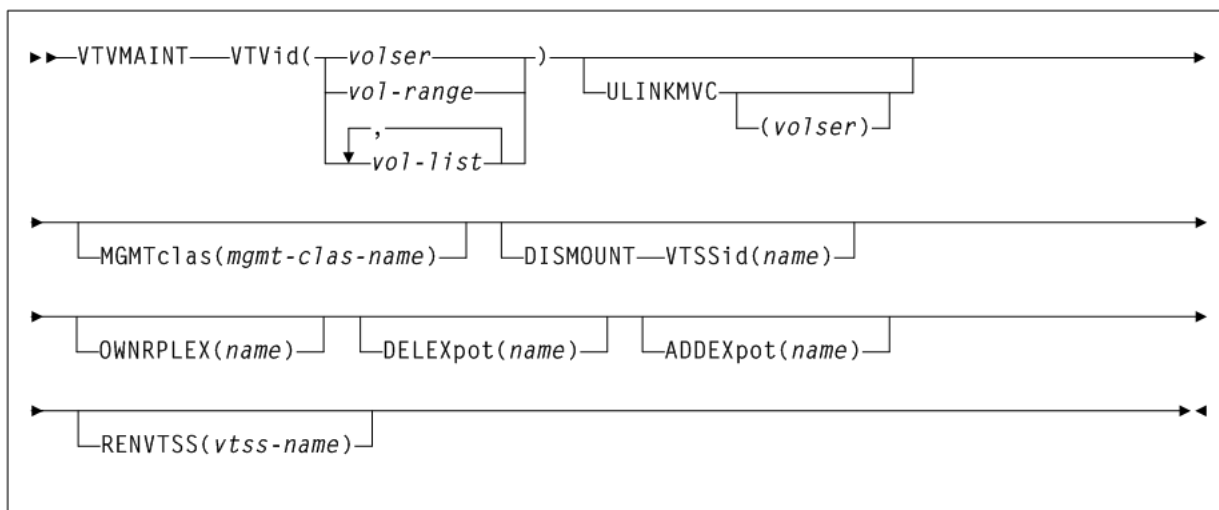
The VTVMaint command does the following:

- Unlinks VTVs from MVCs,
- Sets the VTV Management Class
- Logically dismounts specified VTVs in an offline VTSS.
- Specifies ownership of a VTV that has been electronically imported
- Adds or deletes references to electronic exported copies of VTVs.

## Syntax

Figure 3–185 shows VTVMaint command syntax.

**Figure 3–185** VTVMaint command syntax



## Parameters

As shown in Figure 3–185, the VTVMaint command includes the following parameters:

### **VTVid(volser, vol-range, or vol-list)**

specifies one or more VTVs.

*volser*, *vol-range*, or *vol-list* indicates the volsers of one or more VTVs.

### **ULINKMVC**

optionally, unlink (logically delete) MVC copies of the specified VTVs.

*volser* indicates the volume serial number of the MVC. If no value is specified, then all MVC copies of the VTVs are deleted.

### **MGMTclas(mgmtclas-name)**

optionally, set the Management Class of the VTVs.

*mgmt-class-name* indicates the Management Class name that you specified on the `MGMTclas` control statement. For more information, see "[MGMTclas Control Statement](#)" on page 3-222.

### **DISMOUNT VTSSid(name)**

optionally, logically dismount the specified VTVs in the specified VTSS.

*name* indicates the VTSS name.

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, Oracle strongly recommends that you do not recall these non-current MVC copies!

---



---

When the offline VTSS is ready to be brought back online, it is strongly recommended 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.

### **OWNRPLEX(name)**

optionally, specifies ownership of a VTV that has been electronically imported.

*name* indicates the name of the TapePlex that owns the original copy of the VTV. This name must match one of the known TapePlex names in the configuration.

If this name matches that indicated for the `THISPLEX` parameter on the `CONFIG TAPEplex` statement, the electronic importing status is removed.

### **DELEXpot(name)**

optionally, specifies which references to electronic exported copies of the VTVs are removed.

*name* indicates the name of the TapePlex that is to be removed. If this parameter value is omitted, then all references to electronic export copies are removed.

This parameter merely removes the reference to the external copy of the VTV from the VTV record. It does not cause a contact of the other TapePlex to modify the remote copy in any way. Depending upon the management class definition for the VTV, removing a reference could cause an automatic re-drive of the export at some later time.

### **ADDEXpot(name)**

optionally, specifies the addition of references to the electronic export copy of the VTV.

*name* indicates the name of the TapePlex that is to be added. This name must match to one of the names that has been specified in the VTCS configuration.

This parameter merely adds the reference to the external copy to the VTV record. It is assumed that the required copy of the VTV will have been physically placed there by some other process. The VTV record can track a maximum of two TapePlex locations.

---



---

**Note:** The DELEXPOT operation executes before the ADDEXPOT operation. This makes it possible to replace the exported location information in a single call. For configurations where the electronic export function is not used, it is acceptable to use the ADDEXPOT and DELEXPOT parameters to add dummy tracking or user information to the VTV. The information in the VTV record is always cleared when the VTV data is updated.

---



---

### RENTSS(*vtss-name*)

optionally, sets the name of the VTSS where the VTV was last mounted to *vtss-name*.

*vtss-name* indicates the new VTSS name. This must be the name of an existing VTSS in your configuration.

---



---

**Note:** To use RENTSS, the host on which VTVMaint is run must have access to the owning VTSS.

---



---

## Return Codes

VTVMaint includes the following return codes:

- 0 indicates that all requested updates for the VTV completed successfully.
- 4 indicates that one requested update for the VTV failed and at least one other requested update completed successfully.
- 8 indicates that 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.

---



---

## VTVMaint Report

The following is an example of a VTVMaint report produced by the following command:

```
VTVMaint VTV(X00000-X00002) ULINKMVC MGMTCLAS(M1)
```

### Example 3-62 VTVMaint report

```
VTV          RC
X00000      04
X00001      04
X00002      04
```

```
VTVMaint EXCEPTION REPORT
VTV X00000 IS ALREADY IN MGMTCLAS M1
```

## VTVRpt

---

```
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
```

```
SLUADMIN (7.3.0)   STORAGETEK ENTERPRISE LIBRARY SOFTWARE UTILITY
SSYS=HSCI        PAGE 00003
TIME 15:50:09    VTCS VTV REPORT
DATE 2015-06-17
VTV             SIZE COMP <----CREATION----> <--LAST USED--> MGR SCR REPL  MGMT
MAX
VOLSER         (MB)  % DATE      TIME      DATE      TIME RES IMP EXPT CLASS   MVC1   MVC2   MVC3
MVC4   VTV  VTSSNAME

X00000    0.01  0    2012MAY19 05:02:08 2012MAY19 05:22 - - R - -   M1      022550 022551 022552
022553 800
X00001    0.01  0    2012MAY19 05:02:08 2012MAY19 05:22 - - R - -   M1      033550 033551 033552
033553 800
X00002    0.01  0    2012MAY19 05:02:08 2012MAY19 05:22 - - R - -   M1      044550 044551 044552
044553 800

      3 INITIALIZED VTVS PROCESSED
      0 NON-INITIALIZED VTVS PROCESSED
```

The VTVMaint report shows the following:

- 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` command reports the status of VTVs in your VSM system.

Issue this command with either of the options listed in the following table. Each option is described individually, and in more detail, on the pages to follow.

**Table 3-19** *VTVRpt Options*

Option	Description
BASIC	generates a basic VTV report
COPIES	generates a VTV report indicating where all current copies of a VTV reside

### VTVRpt BASIC

**Interfaces:**

- Utility only
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC not required



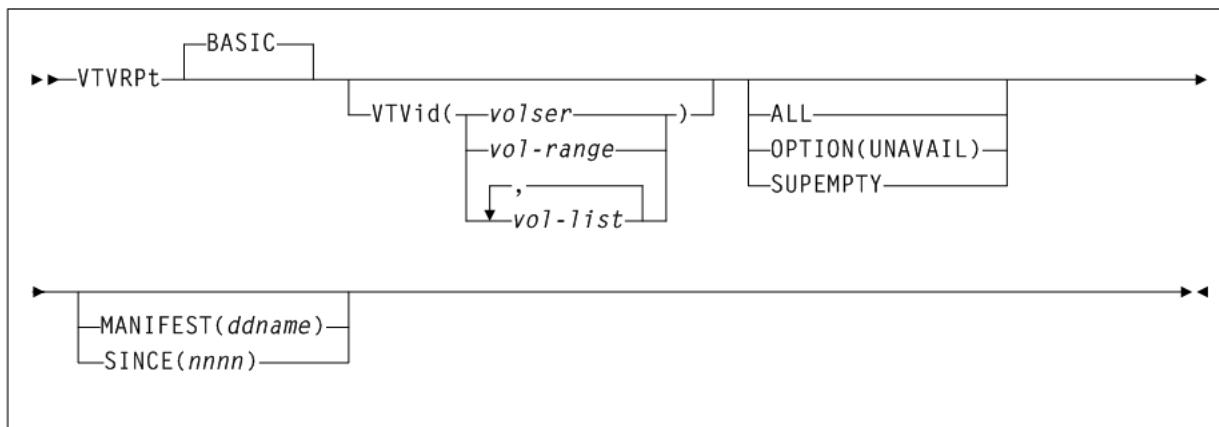
## Description

The `VTVRpt BASIC` command generates a basic VTV report.

## Syntax

Figure 3–186 shows `VTVRpt BASIC` command syntax.

Figure 3–186 `VTVRpt BASIC` syntax



## Parameters

As shown in Figure 3–186, the `VTVRpt BASIC` command includes the following parameters:

### **VTVid(volser, vol-range, or vol-list)**

optionally, 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* indicates the volsers of one or more VTVs.

### **ALL**

optionally, specifies to report on all VTVs (including non-initialized volumes). If you do not specify `ALL`, only initialized VTVs are reported.

### **OPTION(UNAVAIL)**

optionally, specifies to report only on unavailable VTVs (VTVs in an offline VTSS). The report lists only unavailable VTVs in three sections: **unavailable mounted on a VTD**, **unavailable VTSS-resident**, and **unavailable VTSS-resident and fenced**.

### **SUPEMPTY**

optionally, suppress the printing of empty VTVs.

The VTV report will not print details of VTVs that contain no data or VTVs that are fenced. `SUPEMPTY` is mutually exclusive with `ALL` and `OPTION(UNAVAIL)`.

### **MANIFEST(ddname)**

optionally, specifies the input DD name of the manifest file used to generate the report.

*ddname* indicates the DD name of the manifest file. Note that you can specify a merged manifest file or multiple manifest files. `MANIFEST` is mutually exclusive with `SINCE`.

**SINCE(nnnn)**

optionally, specifies a time in minutes. Only VTVs that have been accessed within the specified time are processed.

*nnnn* indicates the time in minutes, from 1 to 9999. SINCE is mutually exclusive with MANIFEST.

**VTV Report (Basic)**

The following is an example of a VTV report produced by the VTVRpt BASIC command:

**Example 3-63 VTVRpt BASIC report**

```

SLUADMIN (7.3.0)   STORAGETEK ENTERPRISE LIBRARY SOFTWARE UTILITY
SSYS=HSCI        PAGE 00002
TIME 15:50:09    VTCS VTV REPORT
DATE 2015-06-17
SLS1315I HSC1.V70EC21.CDS WAS SELECTED AS THE PRIMARY CONTROL DATA SET

SLUADMIN (7.3.0)   STORAGETEK ENTERPRISE LIBRARY SOFTWARE UTILITY
SSYS=HSCI        PAGE 00003
TIME 15:50:09    VTCS VTV REPORT
DATE 2015-06-17
VTV      SIZE  COMP <----CREATION----> <--LAST USED--> MGR SCR REPL  MGMT
MAX
VOLSER   (MB)   % DATE      TIME      DATE      TIME RES IMP EXPT CLASS   MVC1   MVC2   MVC3
MVC4   VTV VTSSNAME
A00240   66.4  85 2010MAY22 16:27:44 2010MAY22 16:57 R - - - - TRACE
4.0 VTSS92
A00241   14.0   0 2010MAY21 13:02:30 2010MAY22 07:48 - M - - - MGMT2CP 702972
0.4 VTSS90
A00406   14.0   0 2010MAY21 13:03:50 2010MAY22 08:45 - M - - - EXHPDM  700486
4.0 VTSS90
...
X50000   2,430.0  0 2009NOV18 09:04:18 2009NOV18 09:07 R - - I - PE EEXPORT
4.0 VTSS91
X50047   280.0  96 2009DEC03 14:29:35 2010APR30 06:48 R M - I - PE QUADYES 700483 700491 700494
700498 4.0 VTSS91
X50048   280.0  96 2009DEC03 14:29:41 2010MAY27 21:06 - M - I - QUADYES 702979 700492 700497
700499 4.0 VTSS91
X50060    0.0   0                - - S I -
X50066   280.0  96 2009DEC03 14:56:17 2010MAY27 21:06 - M - I - DLVTVYES 702978
4.0 VTSS92
X50067   42.0   0 2010JAN21 11:28:55 2010JAN21 11:28 R - - I - EEXPORT
4.0 VTSS91
Y50000    4.8   0 2009NOV20 09:59:52 2009DEC18 11:56 R - - - - E OTHERDIR
0.4 VTSS91
Y50008   1,260.0  75 2009NOV20 09:59:56 2010MAY27 21:09 - M - - - E OTHERDIR 702978
0.4 VTSS92
Y50184   578.6  34 2010FEB02 22:25:27 2010MAY22 08:42 - M S - - APOOL  702972
0.4 VTSS90
485434 INITIALIZED VTVS PROCESSED
116566 NON-INITIALIZED VTVS PROCESSED
SLS0155I CONDITION CODE FOR UTILITY FUNCTION IS 0

```

---



---

**Note:** VTV Report when run against a DR Test CDS will now show the fact in its report header. For example, "VTCS VTV Report (DR Test CDS)".

---



---

### Fields

The following list describes the fields for the report generated by a `VTVRpt` or `VTVRpt BASIC` command:

#### 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 Oracle 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 100 MB VTV compresses to 40 MB 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.

#### Mgr/Res

indicates the copy status of the VTV. The absence of a flag in the columns below indicates that there are no current copies of the VTV within this TapePlex.

- An `R` in the left column indicates that there is a VTSS resident copy of the VTV
- An `M` in the right column indicates that there is at least one MVC copy of the VTV. As an alternative, if this column contains a `C` then the VTV is also consolidated.

#### Scr/Imp

indicates the usability of the VTV.

- An `S` in the left column indicates that the VTV is in a scratched state. If the VTV has also been imported from a remote TapePlex, then the VTV cannot be picked for a scratch mount.
- An `I` in the right column indicates that the VTV has been imported from another TapePlex. Imported VTVs cannot be modified.
- An `O` in the left column indicates that the VTV has partial (write once) `VOLSAFE` protection.
- An `A` in the left column indicates that the VTV has full `VOLSAFE` protection.

#### Repl/Expt

indicates the replication or electronic export status of the VTV. A VTV that is not resident does not have any replication requirements.

It can be one of the following statuses in the left column:

**PR**

The VTV is currently queued for replication.

**SR**

Replication of the VTV has started.

**R**

Replication has completed and another copy of the VTV resides in the partner to the currently resident VTSS.

It can be one of the following statuses in the right column:

**PE**

The VTV requires electronic export.

**SE**

Electronic export of the VTV has started.

**FE**

Electronic export of the VTV has been rejected by the remote TapePlex.

**E**

Electronic export has completed and another copy of the VTV resides in the partner to the currently resident VTSS.

**ME**

electronic export of this VTV to one TapePlex completed successfully, but an electronic export of this VTV to another TapePlex failed.

VTVRPT COPIES will provide the names of the two TapePlexes and indicate which one succeeded and which one failed.

**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 one 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 VTV was last resident. If this field is empty, the VTV is non-existent (not created or used, scratched, and deleted) or has been manually imported.

## VTVRpt COPIES

**Interfaces:**

- Utility only
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC not required

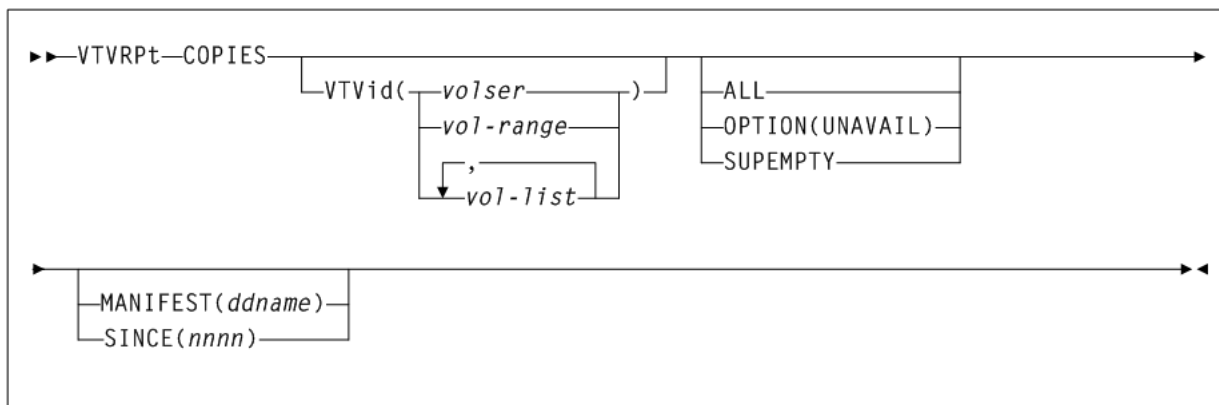
### Description

The `VTVRpt COPIES` command generates a VTV report indicating where all current copies of a VTV reside.

### Syntax

Figure 3–187 shows `VTVRpt COPIES` command syntax.

**Figure 3–187** *VTVRpt COPIES syntax*



### Parameters

As shown in Figure 3–187, the `VTVRpt COPIES` command includes the following parameters:

#### **VTVid(volser, vol-range, or vol-list)**

optionally, 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* indicates the volsers of one or more VTVs.

#### **ALL**

optionally, specifies to report on all VTVs (including non-initialized volumes). If you do not specify `ALL`, only initialized VTVs are reported.

#### **OPTION(UNAVAIL)**

optionally, specifies to report only on unavailable VTVs (VTVs in an offline VTSS). The report lists only unavailable VTVs in three sections: **unavailable mounted on a VTD**, **unavailable VTSS-resident**, and **unavailable VTSS-resident and fenced**.

#### **SUPEMPTY**

optionally, suppress the printing of empty VTVs.

The VTV report will not print details of VTVs that contain no data or VTVs that are fenced. `SUPEMPTY` is mutually exclusive with `ALL` and `OPTION(UNAVAIL)`.

#### **MANIFEST**

optionally, specifies the input DD name of the manifest file used to generate the report.

*ddname* indicates the DD name of the manifest file. Note that you can specify a merged manifest file or multiple manifest files. `MANIFEST` is mutually exclusive with `SINCE`.

**SINCE(nnnn)**

optionally, specifies a time in minutes. Only VTVs that have been accessed within the specified time are processed.

*nnnn* indicates the time in minutes, from 1 to 9999. SINCE is mutually exclusive with MANIFEST.

**VTV Report (COPIES)**

The following is an example of a VTV report produced by the VTVRpt COPIES command:

**Example 3–64 VTVRpt COPIES report**

```

SLUADMIN (7.3.0)   STORAGETEK ENTERPRISE LIBRARY SOFTWARE UTILITY           SSYS=HSCI       PAGE 00002
TIME 15:50:43    VTCS VTV REPORT                                           DATE 2015-06-17
SLS1315I HSC1.V70EC21.CDS WAS SELECTED AS THE PRIMARY CONTROL DATA SET

SLUADMIN (7.3.0)   STORAGETEK ENTERPRISE LIBRARY SOFTWARE UTILITY           SSYS=HSCI       PAGE 00003
TIME 15:50:43    VTCS VTV REPORT                                           DATE 2015-06-17
VTV              SIZE  MGMT      <--VTSS COPIES-->  OWNING    <---MVC COPIES AND LOCATIONS--->  <-EXPORT COPIES-->
VOLSER           (MB)  CLASS   SCR PRIMARY REPLIC TAPEPLEX  MVC1/3  ACS      MVC2/4  ACS      TAPEPLEX TAPEPLEX DRT

00240            10.5  TRACE   -  VTSS92  -      -      -      -      -      -      -
A00241           14.3  MGMT2CP -  -      -      -      -      702972  00
A00406           14.1  EXHPDM -  -      -      -      -      700486  00
...
X50000           2,462.3  EEXPORT -  VTSS91  -      -      EC20REM
X50047           11.4  DLVTVYES -  VTSS91  -      -      EC20REM  700483  00
X50048           11.4  DLVTVYES -  -      -      -      EC20REM  702979  00
X50060           NO COPY DLVTVYES -  -      -      -      SEC20REM
X50066           11.4  DLVTVYES -  -      -      -      EC20REM  702978  00
X50067           42.5  EEXPORT -  VTSS91  -      -      EC20REM
Y50000           13.3  OTHERDIR -  VTSS91  -      -      -      -      EC20REM
Y50008           316.6  OTHERDIR -  -      -      -      -      702978  00
Y50184           385.2  APOOL   S  -      -      -      -      702972  00

      485434 INITIALIZED VTVS PROCESSED
      116566 NON-INITIALIZED VTVS PROCESSED
SLS0155I CONDITION CODE FOR UTILITY FUNCTION IS 0

```

**Fields**

The following list describes the fields for a report generated by the VTVRpt COPIES command:

**VTV Volser**

the VTV volser.

**Size (MB)**

the compressed 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 Oracle StorageTek Software Support.

**MGMT Class**

the name of the Management Class for the VTV specified.

**Scr**

indicates whether the VTV is in a scratched state. An S in this column indicates that the VTV is in a scratched state.

**Primary**

the name of the VTSS that contains the primary VTSS resident copy of the VTV. If this field is blank, then there are currently no VTSS resident copies of the VTV.

**Replica**

the name of the VTSS that contains a replica copy of the VTV. If this field is blank, then there are currently no VTSS resident replica copies of the VTV.

**Owning Tapeplex**

the name of the TapePlex that has ownership of this VTV. If this field is blank, then the VTV is owned by another TapePlex and can be processed normally. If this field is set, then the VTV has been imported from another TapePlex and its contents cannot be modified.

**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. The number next to each MVC volser is the ACS location of the MVC. If this is set to --, then the MVC is not library resident.

**Export Tapeplex**

the names of the TapePlexes to which copies of the VTV have been exported. The copies within these TapePlexes will show the VTV as imported and having an owning TapePlex name. This list is not exhaustive as some of these TapePlexes may have forwarded copies onto further locations. An asterisk (\*) next to the name indicates an export that has been rejected by the target TapePlex.

**DRT**

indicates whether the VTV is a DR Test created version. A 'D' in this column indicates the VTV was created on a DR Test system. A blank indicates the VTV was created on a Production system..

## VVAUDIT

**Interfaces:**

- Utility only
- UUI Support: Yes

**Subsystem Requirements:**

Active HSC at BASE or FULL service level

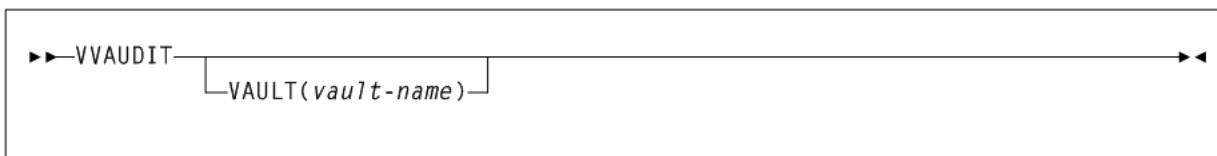
## Description

The VVAUDIT command synchronizes the vault allocation map with the vault volume records. If the slot in the map indicates that it is allocated, then the vault volume records are searched. If the slot number is not found in any vault volume record, then the slot is freed.

## Syntax

Figure 3–188 shows VVAUDIT command syntax.

**Figure 3–188 VVAUDIT command syntax**



## Parameters

As shown in [Figure 3–188](#), the `VVAUDIT` command includes the following parameters:

**VAULT(vault-name)**

optionally, audits the specified vault-name. If `VAULT` is not specified, then all vaults are be audited.

*vault-name* indicates the vault name.

## Warn

**Interfaces:**

- Console or `PARMLIB` only
- UI Support: No

**Subsystem Requirements:**

Active HSC at `BASE` or `FULL` service level

## Description

The `Warn` command sets scratch warning threshold values.

Setting a threshold value causes a warning message to be issued when the number of scratches falls below the specified threshold. The `Warn` command enables you to specify parameters that can narrow the scope of the threshold values you set:

- If `SUBpool`, `MEDIA`, and `RECTech` are not supplied, the threshold value applies to scratch volumes for the specified ACS or LSM.
- If `SUBpool` is supplied, but `MEDIA` and `RECTech` are not, the threshold value applies to scratch volumes for the specified subpool and ACS or LSM.
- If `MEDIA` and `RECTech` are supplied, but `SUBpool` is not, the threshold value applies to scratch volumes for the specified media type, recording technique, and ACS or LSM.
- If `SUBpool`, `MEDIA`, and `RECTech` are supplied, the threshold value applies to scratch volumes for the specified subpool, media type, and recording technique in the ACS or LSM.

---

---

**Note:** Use the `Display THReshd` command to display the current `Warn` threshold values. See "[Display THReshd](#)" on page 3-149 for more information.

---

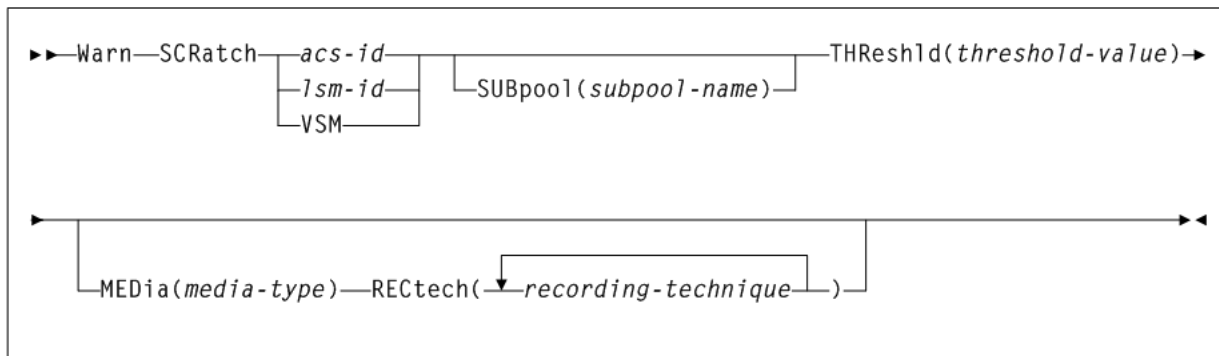
---

## Syntax

[Figure 3–189](#) shows `Warn` command syntax.



Figure 3–189 Warn command syntax



## Parameters

As shown in Figure 3–189, the Warn command includes the following parameters:

### SCRatch

indicates that scratch threshold values are to be set.

#### acs-id

the ACS on which to alter threshold values. The ACSid is a decimal value from 00-99.

#### lsm-id

the LSM on which to alter threshold values. The LSMid is comprised of the ACSid and the LSM number. The format of an LSMid is *AA:LL*, where *AA* indicates the ACSid (decimal 00-99) and *LL* indicates the LSM number (decimal 00-99).

#### VSM

specifies that threshold values are to be set only for subpools for virtual volumes. VSM is not allowed if *acs-id* or *lsm-id* are also specified.

### SUBpool(subpool-name)

optionally, indicates that you want to designate a subpool. This parameter is optional.

*subpool-name* indicates the name of the subpool.

---

**Note:** Scratch subpool names are specified with the SCRPOOL control statement. Refer to Oracle's ELS publication *ELS Legacy Interfaces Reference* for information.

---

### THReshd(threshold-value)

indicates that you want to alter the threshold value.

*threshold-value* indicates the minimum number of scratch volumes that are allowed before the HSC issues a warning message. Allowable threshold values are decimal in the range from 0 through 99,999.

The warning interval values are preset at five minutes.

---

**Note:** When the number of scratch volumes in an ACS or LSM drops below the threshold value, the system issues a warning message.

---

**MEDia(media-type)**

optionally, specifies the media type for the threshold. When MEDia is specified, RECTech must also be specified.

*media-type* indicates the media type.

See [Appendix A, "MEDia, RECTech, and MODel Values"](#) for a list of valid *media-type* values.

**RECTech(recording-technique)**

optionally, specifies the recording technique for the threshold. RECTech indicates the method used to record data tracks on the tape surface. When RECTech is specified, MEDia must also be specified.

You can enter a list of recording techniques, but they must be separated by commas.

*recording-technique* indicates the recording technique.

See [Appendix A, "MEDia, RECTech, and MODel Values"](#) for valid *recording-technique* values.

---

---

**Note:** A list specifies a generic pool from which a selection is made. There is no implied priority.

---

---

---

## SMC Utilities

This chapter describes the following SMC utilities:

- Unified User Interface utility (SMCUUUI)  
This utility enables you to issue HSC Unified User Interface (UUI) commands, VTCS commands, or SMC commands from a batch job.
- JES3 Configuration Report utility (SMCUPJS)  
This utility generates JES3 configuration data which can be used to help verify JES3 initialization deck behavior.
- POLICY and TAPEREQ Batch Test utility (SMCUTRQ)  
This utility enables you to test POLICY and TAPEREQ lookup behavior.
- Tape Management Extract utility (SMCUEBX)  
This utility supports scratch synchronization between the SMC client host and the HSC CDS.
- UUI Simulation utility (SMCUSIM)  
This utility enables you to issue SMC UUI commands in a test environment. Changes to SMC command settings, tape policies, and allocation user exits can be tested before they are put into production without affecting existing SMC clients and HSC servers.
- VSM console SMF Offload utility (SMCUSMF)  
This utility enables you to offload SMF records from a VSM console server.
- Trace Format utility (SMCUGTF)  
This utility supports scratch synchronization between the SMC client host and the HSC CDS.

Unlike HSC and VTCS utilities, all SMC utilities are packaged as standalone utility programs. SMC utilities are not initiated using SMC operator commands.

### SMC Utility Return Codes and Messages

The following table describes SMC utility return codes:

**Table 4–1** SLUADMIN Return Codes

Return Code	Description
0	All utility functions completed normally.

**Table 4–1 (Cont.) SLUADMIN Return Codes**

Return Code	Description
4	At least one utility function encountered an abnormal situation, but it did not cause termination of that utility function or any subsequent utility functions.
8	At least one utility function encountered an error condition that prevented continuation of that utility function, but any subsequent utility functions represented by control statements were attempted.
12	An error condition was detected that either prevented the start of any utility functions, or terminated the active utility function and prevented processing of subsequent utility functions.

Refer to the publication *ELS Messages and Codes* for a listing of SMC messages that may be returned by SMC utilities. These messages are identified by the SMCU prefix.

## Unified User Interface Utility (SMCUUUI)

The SMCUUUI utility is used to issue SMC and HSC Unified User Interface (UII) commands, or VTCS requests from a batch job. The commands can be routed to a local or remote HSC TapePlex.

Several types of output can be received for a UII command, including:

- Plain text output

This type of output is written to `TEXTOUT`, and corresponds to the normal output produced by the HSC or VTCS command or report utility. If plain text output is requested for an HSC or VTCS report, the report headings are not produced by the UII.

- Structured XML

This type of output is written to `XMLOUT`. Structured XML is a well formed XML data stream with XML header tags and XML data tag elements, and XML data element content. Refer to the *ELS Programming Reference* for a complete list of SMC, HSC and VTCS XML tag names and types.

- Comma Separated Values (CSV)

This output type is optional and is generated only if the CSV command is specified. See "[CSV Command](#)" on page 5-6 for more information.

This type of output is written to `CSVOUT`. An input template of XML data tag names is used to define the format of the `CSVOUT` corresponding to a particular request. The output data for each request is a CSV header line that contains a comma-separated list of the XML data tags requested, followed by CSV detail lines. Each CSV detail line contains one comma-separated entry for each requested XML data tag. This type of output is readily adaptable as input into spreadsheets or customized report writers. Refer to the *ELS Programming Reference* for more information about the format for specifying CSV.

The type of output received is controlled by the data set definition statements (DD) defined in the JCL to execute the SMCUUUI utility

**Note:**

- In general, error messages are not produced for CSV output. To view error messages produced by the UUI process, you must include either a `TEXTOUT` or `XMLOUT` DD statement.
- If the SMCUUUI utility is directing utility functions that may include SORT (such as `VOLRPT`, `MVCRPT`, etc.) to a remote TapePlex, caution should be exercised to run a limited number of these jobs simultaneously for a single SMC HTTP server. The exact number is dependent on the region size of the SMC running the HTTP server as well as the amount of data being processed.

**Sample JCL**

The following example shows sample JCL that executes the SMCUUUI utility.

**Example 4-1 JCL to Execute the SMCUUUI Utility**

```
//yourstd JOB card
//STEP EXEC PGM=SMCUUUI, PARM='pgmparms'
//STEPLIB DD DSN=your.sea.linklib, DISP=SHR
//STDOUT DD DSN=yourstd.printout.dataset, DISP=SHR
//UUIIN DD *
      CSV and uui commands
//*****
/* Specify at least one of the following output data sets:      *
//*****
//CSVOUT DD DSN=yourcsv.output.dataset, DISP=SHR
//XMLOUT DD DSN=yourxml.output.dataset, DISP=SHR
//TEXTOUT DD DSN=yourtext.output.dataset, DISP=SHR
```

**Parameters**

The following SMCUUUI utility parameters may be specified for *pgmparms* in the sample JCL:

**SSYS(subsystem-name)**

Specifies the SMC subsystem name (one to four characters in length)

**TAPEPlex(name)**

Specifies the name of the SMC TapePlex to which the UUI commands will be directed. If omitted, the first active TapePlex found is used.

Use `TAPEPlex(SMC)` to indicate that the input commands are to be issued directly to the SMC, and not to a TapePlex (such as the SMC Display Volume and Display Drive command).

**XMLCASE**

Specifies the case of alphabetic data fields in XML and CSV output. Valid case formats include the following:

- M indicates mixed uppercase and lowercase. This is the default.
- U indicates uppercase.

**XMLDate(format)**

Specifies the format of date fields in XML and CSV output. Valid date formats include the following:

- *YYYYMONDD*
- *YYYY-MM-DD*
- *YYYY-MON-DD*
- *HEXSTCK* (hex character *STCK* format)

The default format is *yyyymondd*. For example, 2005Oct14.

**NOHDR**

optionally, specifies that *STDOUT* and *TEXTOUT* report headings and pagination carriage control are not produced. *NOHDR* and *LINECNT* are mutually exclusive.

**LINECNT**

optionally, specifies *STDOUT* and *TEXTOUT* report pagination line count setting. *LINECNT* and *NOHDR* are mutually exclusive.

*nn* is the number of lines per page. Valid values are 10-99.

---

---

**Note:** Settings for *STDOUT* report headings and pagination can only be specified in the *EXEC* statement program parameters. Settings for *TEXTOUT* report headings, while initially specified by the *EXEC* statement program parameters, can later be modified by specifying *TEXTOUT* *UII* control statements in the *UIIIN* command stream.

---

---

For example:

In the following figure, the *EXEC* statement program *PARM='LINECNT(55)'* sets the report pagination to 55 lines per page for both *STDOUT* and *TEXTOUT* output.

However, the *UII* command *TEXTOUT NOHDR* in the *UIIIN* command stream alters the *TEXTOUT* report parameters to disable report headers and pagination.

```
//yourstd JOB card
//STEP EXEC PGM=SMCUUUI,PARM='LINECNT(55)'
//STEPLIB DD DISP=SHR,DSN=your.sea.linklib
//STDOUT DD DISP=SHR,DSN=your.stdout.dataset
//UIIIN DD *
        TEXTOUT NOHDR
        UII commands
//CSVOUT DD DSN=yourcsv.output.dataset,DISP=SHR
//XMLOUT DD DSN=yourxml.output.dataset,DISP=SHR
//TEXTOUT DD DSN=yourtext.output.dataset,DISP=SHR
```

**NOCC**

optionally, indicates that no printer carriage control is to be included. When this option is selected, any non-filled pages will be appended with blank lines for the specified *LINECNT*.

## Required Data Set Definition (DD) Statements

This section describes data sets.

**UUIN Data Set**

You must specify the UUIN data set definition. The UUIN data set can contain the following:

- SMCUI utility control statements
- SDD command (See "SDD Command" on page 4-8.)
- SMC UI commands
- HSC UI commands
- VTCS PGMI requests

Native SMC UI commands include the following:

- Display DRive
- Display Volume
- POLicy
- SIMulate

Refer to the *ELS Programming Reference* for a list of supported HSC UI commands and VTCS PGMI requests.

**Syntax rules**

The following syntax rules apply:

- Input must be 80 byte card images.
- Only columns 1-72 of the 80 byte card image may contain actual command input. Columns 73-80 may be used for sequence numbers if desired.
- If the last non-blank character in columns 1-72 is a "+", the logical command is continued on the next physical card.
- Comments begin with "/\*" (in column 2 or beyond) and end with "\*/".
- The maximum logical command length is 32000 bytes

**Control Statements**

Utility control statements allow the report pagination for TEXTOUT output and the target TapePlex to be altered for subsequent SMC, HSC, and VTCS UI and PGMI commands.

**PLEXDest**

optionally, sets the destination TapePlex for subsequent commands to the specified *tapeplex-name*. Any communication error occurring during a subsequent SMC, HSC, or VTCS command after execution of the PLEXDest control statement results in the bypassing of all subsequent SMC, HSC, or VTCS commands until another PLEXDest control statement is encountered.

*tapeplex-name* is the TapePlex name of the target system for subsequent commands.

If "SMC" is specified, all subsequent commands are assumed to be SMC UI commands and are directed to the simulated SMC environment.

If a valid HSC TapePlex (as defined through a SMC TAPEPlex command) is specified, all subsequent commands are assumed to be HSC or VTCS commands and are directed to the specified tapeplex-name via the simulated SMC environment.

## TEXTOUT

optionally, sets the report pagination parameters for TEXTOUT output. If any errors are encountered on the TEXTOUT command, all subsequent UUI commands are bypassed until a correct TEXTOUT command is encountered.

- NOHDR optionally, specifies that TEXTOUT report headings and pagination carriage control are not produced. NOHDR and LINECNT are mutually exclusive.
- LINECNT (*nn*) optionally, specifies TEXTOUT report pagination line count setting. LINECNT and NOHDR are mutually exclusive.

*nn* is the number of lines per page. Valid values are 10-99.

---



---

### Note:

- Settings for STDOUT report headings and pagination can only be specified in the EXEC statement program parameters. Settings for TEXTOUT report headings, while initially specified by the EXEC statement program parameters, can later be modified by specifying TEXTOUT UUI control statements in the UUIIN command stream.
  - If multiple TEXTOUT statements are input before a single UUI command, the last TEXTOUT statement controls the TEXTOUT report pagination.
  - PLEXDest and TEXTOUT commands can be specified along with CSV commands to alter the destination TapePlex, and the CSV and TEXTOUT output characteristics of subsequent UUI commands. See "[CSV Command](#)" on page 5-6.
- 
- 

The following example shows SMCUUI parameter specifications:

```
//yourstd JOB card
//STEP EXEC PGM=SMCUUI
//UUIIN DD *
PLEXDEST SMC                /* Statement #1 */
D DRIVE                    /* Statement #2 */
PLEXDEST HSCN              /* Statement #3 */
DISPLAY SCRATCH           /* Statement #4 */
PLEXDEST HSCQ             /* Statement #5 */
DISPLAY SCRATCH           /* Statement #6 */
...other JCL statements
```

This example is described as follows:

- Statement #1 sets the destination for subsequent commands to the local SMC subsystem.
- The D DRIVE Statement #2 is directed to the local SMC subsystem.
- Statement #3 resets the destination TapePlex for subsequent commands to TapePlex HSCN.
- The DISPLAY SCRATCH Statement #4 is directed to HSCN.
- Statement #5 resets the destination TapePlex for subsequent commands to HSCQ.
- Finally, DISPLAY SCRATCH Statement #6 is directed to HSCQ.

The next example shows additional SMCUUI parameter specifications:



```

//yourstd JOB card
//STEP EXEC PGM=SMCUUUI
...other JCL statements
//UIIIN DD *
PLEXDEST HSCN                               /*Statement #1 */
CSV ...csv parameters                        /*Statement #2 */
SCRATCH VOL ACT001-ACT002                   /*Statement #3 */
SCRATCH VOL PAY001-PAY002                   /*Statement #4 */
SCRATCH VOL REC001-REC002                   /*Statement #5 */
PLEXDEST HSCQ                               /*Statement #6 */
SCRATCH VOL Q00001-Q00002                   /*Statement #7 */
CSV ...csv parameters                       /*Statement #8 */
DISPLAY SCRATCH                             /*Statement #9 */
...other UII commands
...other JCL statements

```

This example is described as follows:

- Statement 1 sets the destination TapePlex for subsequent commands to TapePlex HSCN.  
If HSCN is not available, or communication errors occur while processing statements 3-5, then subsequent UII commands are bypassed until the next PLEXDest command is encountered (statement 6 in this example).
- Statement 2 sets the CSV output parameters for subsequent UII commands.  
If an error occurs on the CSV command, then any subsequent UII commands are bypassed until a subsequent valid CSV command is read. In this case, if the CSV command in statement #2 is invalid, not only are statements 3-5 bypassed, but also statement #7. Statement #6 is still processed to direct subsequent commands to HSCQ.
- Statement #8 sets new CSV output parameters to be used by statement #9.

### STDOUT Data Set

The STDOUT data set is where the SMCUUUI utility reports the input and final completion code for each UII request. The STDOUT DD statement is not required. However, Oracle recommends that it be included in case any utility errors occur.

### Output Data Set Definition (DD) Statements

You must specify at least one of the XMLOUT, CSVOUT, or TEXTOUT output data set definition statements, as follows:

- XMLOUT

The XMLOUT data set is where the SMCUUUI utility outputs the structured XML output corresponding to the UIIIN command, as shown in the following example:

```

//yourstd JOB card
//STEP EXEC PGM=SMCUUUI
...
//UIIIN DD *
SCRATCH VOL(400720)
//XMLOUT DD DSN=yourxml.output.dataset,DISP=SHR
...

```

The XMLOUT DD would contain the following output:

```

<scratch request>
<header>

```

```

<hsc_version>7.3.0</hsc_version>
<date>2008Apr19</date>
<time>13:54:34</time>
<host_name>MVS</host_name>
</header>
<volume_data>
<volser>400720</volser>
<result>success</result>
<reason>SLS0167I Volume 400720 successfully added to library as scratch
</reason>
</volume_data>
</scratch_request>

```

- CSVOUT

The CSVOUT data set is where the SMCUI utility writes the CSV output for the requested UI commands that specified corresponding CSV formatting. Refer to the *ELS Programming Reference* for more information about the Comma Separated Value (CSV) specification. The maximum logical CSV input length is 4096 bytes.

- TEXTOUT

The TEXTOUT data set is where the SMCUI utility outputs plain text UI responses. This is the same output that would be received from the corresponding SMC, HSC or VTCS command, or written to the SLSPRINT DD by the SLUADMIN or SWSADMIN utility, except that page and column headings are not produced. This data set is optional

## Remote UI Input and Output

A new XAPI/UI interface supports remote UI input and output. To enable this support, Include the SDD command in the SMCUI utility.

---



---

**Note:** Remote UI input and output may also be requested using the NCSCOMM interface. Refer to the *ELS Programming Reference* for more information.

---



---

### Remote UI Input

A data set is transmitted from the client to a remote server for input, without the data set needing to be resident on the server.

Remote UI input is required for commands that require large amounts of input data from sources other than the command line. Commands include VTCS CONFIG, IMPORT, and MERGMFST.

### Remote UI Output

A remote command produces an output data set that is transmitted and physically written on the requesting client without it being physically written on the server.

Remote UI output is required for commands that produce non-text, non-XML, non-CSV output data sets. Commands include VTCS EXPORT and MERGMFST.

### SDD Command

The SDD command defines an input or output data set to be associated with a “real” HSC or VTCS command that immediately follows it.

In the following example, the `SDD` command is associated with the `IMPORT` command, but not the subsequent `VTVRPT` command:

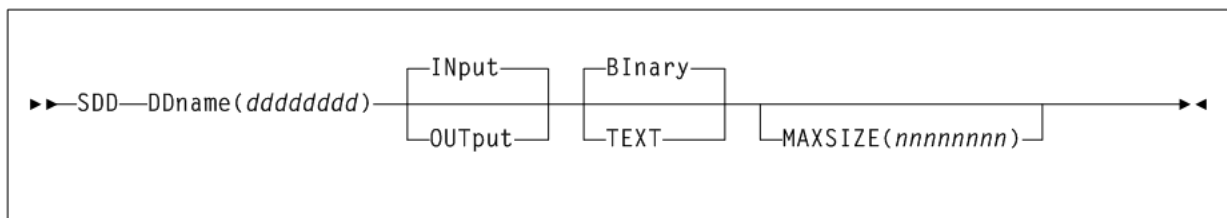
```
SDD DDNAME(MAININ) INPUT BINARY
IMPORT MANIFEST(MANIIN)
CSV FIELD(VOLSER,DATE_MIGRATED),BREAK(VTV_DATA)
VTVRPT VTV(SA2000-SA2999)
```

The `SDD` command can be used in either the `SLUADMIN` or `SMCUUUI` utility. However, only the `SMCUUUI` utility supports remote UI command execution.

### Syntax

The following figure shows the syntax for the `SDD` command:

**Figure 4–1 SDD command syntax**



### Parameters

#### DDname(dddddddd)

the name of the JCL DD that either contains the additional non-command line input, or will contain the output data set upon command completion.

#### INput|OUTput

optionally, specifies whether the data set specified by the `DDname` parameter is an input or output data set.

`INput` specifies that the data set is read to supply additional non-command line input to the command. This is the default if neither `INput` or `OUTput` are specified.

`OUTput` specifies that the data set is written to contain the non-XML, non-text, non-CSV output produced by the command.

#### BInary|TEXT

optionally, specifies whether the data read or produced contains `BInary` (that is, non display) characters, or is in `SLSIN TEXT` command format.

If you specify `TEXT` along with the `INput` parameter, this specifies that the input data is in `SLSIN` format. Only the first 72 characters are read and continuation characters may be present. Comment statements are not passed to the `HSC` or `VTCS` command that follows.

`BInary` is the default if neither `BInary` or `TEXT` is specified.

---

**Note:** You must specify `TEXT` when the `SDD` control statement is to be associated with the `VTCS CONFIG` command.

---

**MAXSIZE(nnnnnnnn)**

optionally, specifies the maximum size (in K units) of the input or output file. If MAXsize is not specified, there is no limit to the maximum size.

**CONFIg Command Example**

Without the SDD command, the VTCS CONFIg command is only supported under the SLUADMIN utility. Under SLUADMIN without the SDD command, SLUADMIN CONFIg reads the remainder of the SLSIN data set and assumes that the remaining input cards are CONFIg control statements.

The SDD command enables you to specify VTCS CONFIg under either the SMCUUUI or SLUADMIN utility. Additionally, it enables other commands to execute in the same input stream after the VTCS CONFIg command, as shown in the following examples:

SLUADMIN:

```
//jobname job (account),programmer,REGION=0M
//S1 EXEC PGM=SLUADMIN,PARM=MIXED
//STEPLIB DD DSN=your.sea.linklib,DISP=SHR
//*****
/* Optional UUI output definition DD(s) *
//*****
//SLSCSV DD DSN=yourcsv.output.dataset,DISP=SHR
//SLSXML DD DSN=yourxml.output.dataset,DISP=SHR
//*****
/* SDD associated DD(s) *
//*****
//CONFIGIN DD DSN=your.config.dataset,DISP=SHR
//*****
/* Required SLUADMIN output and input DDs *
//*****
//SLSPRINT DD SYSOUT=A
//SLSIN DD *
SDD DDNAME(CONFIGIN) INPUT TEXT
.CONFIG CDSLEVEL(V61ABOVE)
DISPLAY CLINK
/*
```

SMCUUUI:

```
//jobname job (account),programmer,REGION=0M
//S1 EXEC PGM=SMCUUUI,PARM=MIXED
//STEPLIB DD DSN=your.sea.linklib,DISP=SHR
//*****
/* Specify as least one of the following output definitions *
//*****
//CSVOUT DD DSN=yourcsv.output.dataset,DISP=SHR
//XMLOUT DD DSN=yourxml.output.dataset,DISP=SHR
//TEXTOUT DD SYSOUT=A
//*****
/* SDD associated DD(s) *
//*****
//CONFIGIN DD DSN=your.config.dataset,DISP=SHR
//*****
/* Required SMCUUUI input DD *
//*****
//UUIIN DD *
SDD DDNAME(CONFIGIN) INPUT TEXT
CONFIG CDSLEVEL(V61ABOVE)
DISPLAY CLINK
```

/\*

**Note:**

- The CONFIGIN data set is read and internally packaged for possible transmittal before the CONFIG command is read. The TEXT parameter results in any SLSIN formatting to be resolved as part of the internal packaging (that is, '+' concatenated statements are resolved).
- If any SDD command errors are encountered (that is, the CONFIGIN DD is missing in the JCL or the data set cannot be read), then the succeeding CONFIG statement is bypassed.
- In either of the above examples, the SDD command TEXT parameter is required when associated with the CONFIG command. The SDD command INPUT parameter is the default and may be omitted.

**IMPORT Command Example**

Without the SDD command, the VTCS IMPORT command is only supported for local processing under either the SLUADMIN or SMCUUUI utility.

The SDD command enables you to execute VTCS IMPORT on a remote server using the SMCUUUI utility, as shown in the following example:

```
//jobname job (account),programmer,REGION=0M
//S1 EXEC PGM=SMCUUUI,PARM=MIXED
//STEPLIB DD DSN=your.sea.linklib,DISP=SHR
//*****
/* Specify as least one of the following output definitions *
//*****
//CSVOUT DD DSN=yourcsv.output.dataset,DISP=SHR
//XMLOUT DD DSN=yourxml.output.dataset,DISP=SHR
//TEXTOUT DD SYSOUT=A
//*****
/* SDD associated DD(s) *
//*****
//MANIIN DD DSN=your.manifest.dataset,DISP=SHR
//*****
/* Required SMCUUUI input DD *
//*****
//UUIIN DD *
SDD DDNAME(MANIIN) MVCS(volser) INPUT BINARY
.IMPORT REPLACE=ALL MANIFEST(MANIIN)
/*
```

---



---

**Note:**

- The MANIIN data set is read and internally packaged for possible transmittal across the network before the IMPORT command is read. The BINARY parameter results in the transmittal of record images.
  - In the above example, both the SDD command BINARY and INPUT parameters are the defaults and may be omitted.
  - While a similar SLUADMIN job with the SDD command could be constructed, it would still be limited to local execution. Only the SMCUUUI utility supports remote UUI command execution.
- 
- 

**EXPORT Command Example**

Without the SDD command, the VTCS EXPORT command is only supported for local processing under either the SLUADMIN or SMCUUUI utility. The SDD command enables you to execute VTCS EXPORT on a remote server using the SMCUUUI utility, as shown in the following example:

```
//jobname job (account),programmer,REGION=0M
//S1 EXEC PGM=SMCUUUI,PARM=MIXED
//STEPLIB DD DSN=your.sea.linklib,DISP=SHR
//*****
/* Specify as least one of the following output definitions *
//*****
//CSVOUT DD DSN=yourcsv.output.dataset,DISP=SHR
//XMLOUT DD DSN=yourxml.output.dataset,DISP=SHR
//TEXTOUT DD SYSOUT=A
//*****
/* SDD associated DD(s) *
//*****
//MANIOUT DD DSN=your.manifest.dataset,DISP=SHR
//*****
/* Required SMCUUUI input DD *
//*****
//UUIIN DD *
SDD DDNAME(MANIOUT) OUTPUT BINARY
EXPORT MVC(volser) MANIFEST(MANIIN)
/*
```

---



---

**Note:**

- The MANIOUT data set is written on the client when remote output is received. However the MANIOUT DD will be validated before the EXPORT command is read.
  - In the above example, the SDD command parameter OUTPUT must be specified for EXPORT. However, BINARY is the default and may be omitted.
  - While a similar SLUADMIN job with SDD command could be constructed it would still be limited to local execution. Only the SMCUUUI utility supports remote UUI command execution.
- 
- 

**MERGMST Command Example**

The MERGMFST process uses a combination of input and output SDD commands in the SMCUUUI utility, as shown in the following example:

```
//jobname job (account),programmer,REGION=0M
//S1 EXEC PGM=SMCUUUI,PARM=MIXED
//STEPLIB DD DSN=your.sea.linklib,DISP=SHR
//*****
/* Specify as least one of the following output definitions *
//*****
//CSVOUT DD DSN=yourcsv.output.dataset,DISP=SHR
//XMLOUT DD DSN=yourxml.output.dataset,DISP=SHR
//TEXTOUT DD SYSOUT=A
//*****
/* SDD associated DD(s) *
//*****
//MANIIN DD DSN=your.mergein.dataset1,DISP=SHR
// DD DSN=your.mergein.dataset2,DISP=SHR
. . .
// DD DSN=your.mergein.datasetn,DISP=SHR
//MANIOUT DD DSN=your.mergeout.dataset,DISP=SHR
//*****
/* Required SMCUUUI input DD *
//*****
//UIIN DD *
SDD DDNAME(MANIIN) INPUT BINARY
SDD DDNAME(MANIOUT) OUTPUT BINARY
.MERGMFST MERGEIN(MANIIN) MERGEOUT(MANIOUT)
/*
```

---



---

**Note:**

- The MANIOUT data set is written on the client when remote output is received. However, the MANIOUT DD will be validated before the MERGMFST command is read.
  - In the above example, the SDD MANIOUT command OUTPUT parameter must be specified for the MERGEOUT associated DD. However, BINARY is the default and may be omitted.
  - While a similar SLUADMIN job with SDD commands could be constructed, it would still be limited to local execution. Only the SMCUUUI utility supports remote UUI command execution.
- 
- 

## JES3 Configuration Report Utility (SMCUPJS)

The SMCUPJS utility generates JES3 configuration data which can be used to help verify that the JES3 initialization deck defines the intended devices and esoterics. This data is derived from internal SMC JES3 configuration control structures. These control structures are built during SMC/JES3 initialization through analysis of the SETNAME and SETUNIT statements in the user's JES3 initialization deck. For this reason, the SMC must be active on the system on which the report is run.

The following reports are generated from the SMC JES3 configuration control structures:

- List of XTYPE numbers in hexadecimal order with their corresponding XTYPE names
- List of XTYPES in hexadecimal order with all esoterics in which the XTYPE appears

- List of esoterics in alphabetical order with all XTYPES that each contains and the total number of devices in the esoteric. This report also lists any discrepancies between JES3 and HCD esoterics
- List of XTYPES in hexadecimal order with all devices in that XTYPE, along with the location and recording technique of the devices in the XTYPE
- List of devices in hexadecimal order and the XTYPE to which each belongs, along with the location and recording technique of the device

## Sample JCL

The following example shows sample JCL that executes the SMCUPJS utility to produce the five individual reports listed above.

### **Example 4–2 JCL to Produce the JES3 Configuration Report**

```
//yourstd JOB card
//STEP EXEC PGM=SMCUPJS
//STEPLIB DD DSN=your.sea.linklib,DISP=SHR
//STDOUT DD DSN=yourstd.printout.dataset,DISP=SHR
```

---

---

**Note:** *yourstd.printout.dataset* should be allocated with LRECL=80.

---

---

## Sample Reports

In the following report fragments, XTYPE 27 is carried through all five reports to illustrate how the reports are related. Lines pertaining to XTYPE 27 are underlined in each report.

### **XTYPE Number to XTYPE Name (EXTON)**

Below are fragments of the XTYPE Number to XTYPE Name report. This mapping is useful when reading other reports that reference the XTYPE number instead of the name. The number in parenthesis is the XTYPE number, followed by the XTYPE name.

```
SMCUPJS (7.3.0)          SMC JES3 CONFIGURATION REPORT          PAGE 0001
TIME 15:03:42          XTYPE NUMBER TO XTYPE NAME (EXTON)    DATE 2015-01-09

XTYPE(01) DEV01D5
XTYPE(02) DEV2504
XTYPE(03) DEV2507
XTYPE(04) DEV2506
XTYPE(05) DEV01D6
XTYPE(06) DEV01DA
XTYPE(07) DEV01DD
XTYPE(08) DEV01DF
XTYPE(09) DEV01D0
...
XTYPE(27) DEV0AA0
XTYPE(28) DEV0A20
...
```



**XTYPE Number to Esoteric (EXTOE)**

Below are fragments of the XTYPE Number to Esoteric report, which lists each XTYPE in hexadecimal order, along with all esoterics that contain that XTYPE. Both the XTYPE number (in parenthesis) and name are displayed.

```
SMCUPJS (7.3.0)      SMC JES3 CONFIGURATION REPORT          PAGE 0001
TIME 15:03:42      XTYPE NUMBER TO ESOTERIC (EXTOE)      DATE 2015-01-09
```

```
XTYPE(01) DEV01D5  appears in 11 ESOTERICs
  CART   EAGLE   3590-1  HVALL   HSCV3590 MCZZZZ0  HVZZZZ01 MCZZ9840
  HV994035 HRACS1L1 HV994010
XTYPE(02) DEV2504  appears in 10 ESOTERICs
  CART
```

...

```
XTYPE(27) DEV0AA0  appears in 7 ESOTERICs
  SYS3480R  CART    3480    TL3480  SLIB    SACS0   S04480
```

**Esoteric to XTYPE (EETOX)**

Below are fragments of the Esoteric to XTYPE report, which lists each esoteric in alphabetical order, along with all XTYPES in that esoteric in hexadecimal order. The XTYPE number can be mapped to its corresponding XTYPE name using the XTYPE Name to XTYPE Number report described above.

```
SMCUPJS (7.3.0)      SMC JES3 CONFIGURATION REPORT          PAGE 0001
TIME 15:03:42      ESOTERIC TO XTYPE NUMBER (EETOX)      DATE 2015-01-09
```

```
ESOTERIC AUSALL   contains 4 XTYPEs and 12 devices
  2C 2D 3D 85
ESOTERIC AUSREDW  contains 1 XTYPEs and 2 devices
  2D
```

...

```
ESOTERIC CART     contains 84 XTYPEs and 433 devices
  01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11 21 22 23 24 25 26 27
  28 29 2A 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F 40 41 42
  43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53 54 55 56 57 58 59 5A
  5B 5C 5D 5E 5F 60 61 62 63 64 65 66
```

...

```
ESOTERIC SACS0    contains 9 XTYPEs and 22 devices
  10 11 27 42 43 44 45 5E 5F
```

SMCU0005 \*\*\* WARNING: HCD esoteric does not match JES3 esoteric

```
JES3 esoteric drives not in HCD:
  0AA3
HCD esoteric drives not in JES3:
  (NONE)
```

...

```
ESOTERIC SLIB     contains 13 XTYPEs and 34 devices
  10 11 23 24 27 42 43 44 45 46 47 5E 5F
```

...

```
ESOTERIC SYS3480R contains 102 XTYPEs and 1511 devices
  21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38
  39 3A 3B 3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50
  51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F 60 61 62 63 64 65 66 67 68
  69 6A 6B 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 7A 7B 7C 7D 7E 7F 80
  81 82 83 84 85 86
```

...

```
ESOTERIC S04480   contains 1 XTYPEs and 4 devices
  27
```

```

...
ESOTERIC TL3480  contains 5 XTYPEs and 48 devices
 25 26 27 28 29
...
ESOTERIC 3480    contains 14 XTYPEs and 132 devices
 21 22 23 24 25 26 27 28 29 83 84 85 86
...

```

### XTYPE to Device Number (EXTOD)

Below are fragments of the XTYPE to Device Number report, which lists each XTYPE in hexadecimal order along with its devices. For each XTYPE number the following information is displayed:

- XTYPE number
- XTYPE name
- Number of devices in the XTYPE
- Associated TapePlex
- Device location
- Device recording technique

The device location is one of the following:

- In a TapePlex, in which case "*tapeplex-name* LSM AA:LL" is displayed, where *tapeplex-name* is the name of the TapePlex containing the HSC where the nonlibrary device is defined via the NONLIB esoteric or UNITATTR
- In a VTSS, in which case "*tapeplex-name* VTSS (nnnnnnnn)" is displayed
- Defined with an SMC UNITATTR command, in which case "nonlib" is displayed
- Neither in the library nor defined with an SMC UNITATTR command, in which case "not lib or nonlib" is displayed.

#### Example 4-3 EXTOD report fragment

```

SMCUPJS (7.3.0)      SMC JES3 CONFIGURATION REPORT          PAGE 0001
TIME 15:03:42        XTYPE TO DEVICE NUMBER (EXTOD)                DATE 2015-01-09

XTYPE(01) DEV01D5  contains 1 DEVICES  not lib or nonlib, rectech STK1RA35
 01D5
XTYPE(02) DEV2504  contains 1 DEVICES  not lib or nonlib, rectech STK1RA35
 2505
...
XTYPE(0F) DEV2801  contains 1 DEVICES  nonlib, rectech STK1RA35
 2801
...
XTYPE(27) DEV0AA0  contains 4 DEVICES  PLEX1 LSM 00:00, rectech 18TRACK
 0AA0 0AA1 0AA2 0AA3
...
XTYPE(48) DEV9000  contains 64 DEVICES  PLEX1 VTSS SVTSS1
 9000 9001 9002 9003 9004 9005 9006 9007 9008 9009 900A 900B 900C 900D
 900E 900F 9010 9011 9012 9013 9014 9015 9016 9017 9018 9019 901A 901B
 901C 901D 901E 901F 9020 9021 9022 9023 9024 9025 9026 9027 9028 9029
 902A 902B 902C 902D 902E 902F 9030 9031 9032 9033 9034 9035 9036 9037
 9038 9039 903A 903B 903C 903D 903E 903F
...

```

**Device Number to XTYPE (EDTOX)**

Below are fragments of the Device Number to XTYPE report, which lists each device in hexadecimal order and the XTYPE to which it belongs. For each device the following is displayed:

- Associated TapePlex
- Device location
- Device recording technique

The device location and recording technique are displayed as in the XTYPE to Device Number report described above.

**Example 4–4 EDTOX report fragment**

```
SMCUPJS (7.3.0)      SMC JES3 CONFIGURATION REPORT          PAGE 0001
TIME 15:03:42      DEVICE TO XTYPE NUMBER (EDTOX)          DATE 2015-01-09

DEVICE/XTYPE - 0120/3C  not lib or nonlib, rectech 36ATRACK
DEVICE/XTYPE - 0121/3C  not lib or nonlib, rectech 36ATRACK
...
DEVICE/XTYPE - 0AA0/27  PLEX1 LSM 00:00, rectech 18TRACK
DEVICE/XTYPE - 0AA1/27  PLEX1 LSM 00:00, rectech 18TRACK
DEVICE/XTYPE - 0AA2/27  PLEX1 LSM 00:00, rectech 18TRACK
DEVICE/XTYPE - 0AA3/27  PLEX1 LSM 00:00, rectech 18TRACK
...
```

**POLicy and TAPEREQ Batch Test Utility (SMCUTRQ)**

The SMCUTRQ utility is used to test SMC tape policy lookups. The program accepts control cards containing the same type of information that is extracted by the SMC during tape allocation and mount processing.

After you input the jobname, stepname, program name and other attributes on the control cards, this utility formats those attributes and requests SMC tape policy lookups from the SMC subsystem as if it were a job with those simulated characteristics.

Next, the utility generates a report indicating what policy was matched, or what TREQDef TAPEREQ statements, if any, were matched for the specific criteria. If the currently active TREQDef file is accessible, as indicated by the TREQDef LIst command, the actual TAPEREQ statements matching each lookup are listed.

The SMCUTRQ utility requires an active SMC subsystem on the same host. If SMC is not active, an error message is produced and the program terminates.

**Sample JCL**

Execute load module SMCUTRQ in the SMC linklib to produce the report. The following example shows sample JCL that executes the SMCUTRQ utility:

**Example 4–5 JCL to Execute the SMCUTRQ Utility**

```
//yourstd JOB card
//STEP    EXEC PGM=SMCUTRQ
```

```
//STEPLIB DD DSN=your.sea.linklib,DISP=SHR
//STDOUT DD DSN=yourstd.printout.dataset,DISP=SHR
//SYSOUT DD SYSOUT=*
//SMCIN DD *
POL1 jobname stepname pgmname ddname nnnn yyyy/ddd mgmtclas
POL2 C volser dataset.name
```

## Control Cards

Each TAPEREQ POLicy lookup requires that you enter at least one control card (POL1), and optionally one or two control cards (POL2, POL3) specifying the job step characteristics to be simulated. The control card format is as follows:

Card 1:

The following table lists information for control card 1:

**Table 4–2 Control Card 1 Information**

Column	Length	Field	Format
1	4	ID	"POL1" required identifier
6	8	jobname	1-8 character non-generic jobname
15	8	stepname	1-8 character non-generic stepname
24	8	program name	1-8 character non-generic program name
33	8	ddname	1-8 character non-generic ddname
43	4	retention period	4 digit retention period
47	8	expiration date	yyddd, yyyyddd, or yyyy/ddd format expiration date
56	8	management class	1-8 character non-generic SMS management class name

Card 2:

The following table lists information for control card 2:

**Table 4–3 Control Card 2 Information**

Column	Length	Field	Format
1	4	ID	"POL2" required identifier
6	1	volume type	"S" for specific or "N" for non-specific (scratch)
8	6	volume serial	6 character volser
15	44	data set name	44 character non-PDS MVS dataset name

Card 3:

The following table lists information for control card 3:

**Table 4–4 Control Card 3 Information**

Column	Length	Field	Format
1	4	ID	"POL3" required identifier

**Table 4–4 (Cont.) Control Card 3 Information**

Column	Length	Field	Format
6	8	actual RECTECH	Valid recording technique
15	3	label type	Values SL, NL, AL, NSL
19	8	JCL esoteric	Esoteric name

**Note:**

- All variable length fields must be left justified.
- Retention period and expiration date fields are mutually exclusive. When both are specified, retention period is used.
- The management class field is used to simulate a returned SMS ACS MGMTCLAS and assumes that the SMC SMSDEF MGMTPol (ALL) or (IDAX) policy has been specified. If the SMC finds a matching policy name, then the remaining tape policy lookups are bypassed.
- If a policy name matching the specified management class name is not found, or if the management class field was not specified, then a series of TAPEREQ lookups is performed using the remaining input criteria.
- You may ignore the lookup fields that are not important to your lookup. This is determined by looking at what selection criteria you actually use on your TAPEREQ statements. However, it is recommended that you specify all non-mutually exclusive fields.
- "POL1" control card is required, even if no input fields are coded. Both "POL2" and "POL3" control cards are optional.
- You can input multiple sets of control cards.

**Sample Input**

The following are examples of TAPEREQ lookup scenarios:

**Example 1**

The following example simulates the TAPEREQ lookup that would occur when:

- Specific volume 123456 for DSN ABC.QUAL1.QUAL2.TEST
- Allocated to DD MYTAPE1 for job TSTNEWPG, step STEP1, executing program TESTPGM.

```
POL1 TSTNEWPG STEP1 TESTPGM MYTAPE1
POL2 S 123456 ABC.QUAL1.QUAL2.TEST
```

**Example 2**

The following example simulates the TAPEREQ lookup that would occur when:

- Scratch volume for DSN ABC.QUAL1.QUAL2.TEST
- Allocated to DD MYTAPE1 with a retention period of 99 days
- For job TSTNEWPG, step STEP1, executing program TESTPGM.

```
POL1 TSTNEWPG STEP1      TESTPGM MYTAPE1 0099
POL2 N          ABC.QUAL1.QUAL2.TEST
```

### Example 3

The following example simulates the TAPEREQ lookup that would occur when:

- Scratch volume for DSN ABC.QUAL1.QUAL2.TEST
- Allocated to DD MYTAPE1
- For job TSTNEWPG, step STEP1, executing program TESTPGM, when the DFSMS ACS routine returns a management class of MCLASS01.

```
POL1 TSTNEWPG STEP1      TESTPGM MYTAPE1 MCLASS01
POL2 N 123456 ABC.QUAL1.QUAL2.TEST
```

## Sample Output

The following example shows output from the SMCUTRQ utility:

### Example 4-6 SMCUTRQ Utility Output

```
SMCUTRQ 7.3.0      SMC POLICY/TAPEREQ TEST
Time 11:25:49     POLICY/TAPEREQ Lookup Report

SMCU0019 SMC subsystem=SMC0

Card 0001: *OL1|JOBNAME|STEPNAME|PGMNAME|DDNAME|RETP|EXPDT|SMSMGMT
Card 0002: *OL2|T|VOLSER|DSN
Card 0003: POL1 ANYJOB  STEP1      TESTPGM DD1
Card 0004: POL2 S 123456 ABC.QUAL1.QUAL2.TEST

TAPEREQ lookup keys:
      JOBname=ANYJOB  STEPname=STEP1  PROGname=TESTPGM DDname=DD1
      VOLType=S      VOLSer=123456    DSN=ABC.QUAL1.QUAL2.TEST

      Lookup for POLICY          not matched
      Lookup for RECTECH/MEDIA  matched TREQDEF record 5
      Statement: TAPEREQ DSN(ABC.**.TEST) MEDIA(VIRTUAL)
      Lookup for SUBPOOL        not matched
      Lookup for ESOTERIC       not matched
      Lookup for MGMTCLAS       not matched
      Lookup for DEVTYPREF      not matched

Card 0005: POL1 ANYJOB  ANYSTEP  ANYPGM  DD1
Card 0006: POL2 S 123456 ABC.D11.XYZ.ABC

TAPEREQ lookup keys:
      JOBname=ANYJOB  STEPname=ANYSTEP  PROGname=ANYPGM  DDname=DD1
      VOLType=S      VOLSer=123456    DSN=ABC.D11.XYZ.ABC

      Lookup for POLICY          not matched
      Lookup for RECTECH/MEDIA  matched TREQDEF record 10
      Statement: TAPEREQ DSN(**.XYZ.***) MEDIA(VIRTUAL)
      Lookup for SUBPOOL        not matched
      Lookup for ESOTERIC       not matched
      Lookup for MGMTCLAS       not matched
      Lookup for DEVTYPREF      not matched
```

Processing complete; RC=0

## Tape Management Extract Utility (SMCUBDX)

The SMCUBDX utility supports scratch synchronization between the SMC client host and the HSC CDS. Using common Tape Management System interface (SLUHDR\*) routines with HSC and MVS/CSC, SMC accesses the client tape management system to produce an extract file that can be read as input by the HSC Scratch Conversion Utility (SLUCONDB).

---

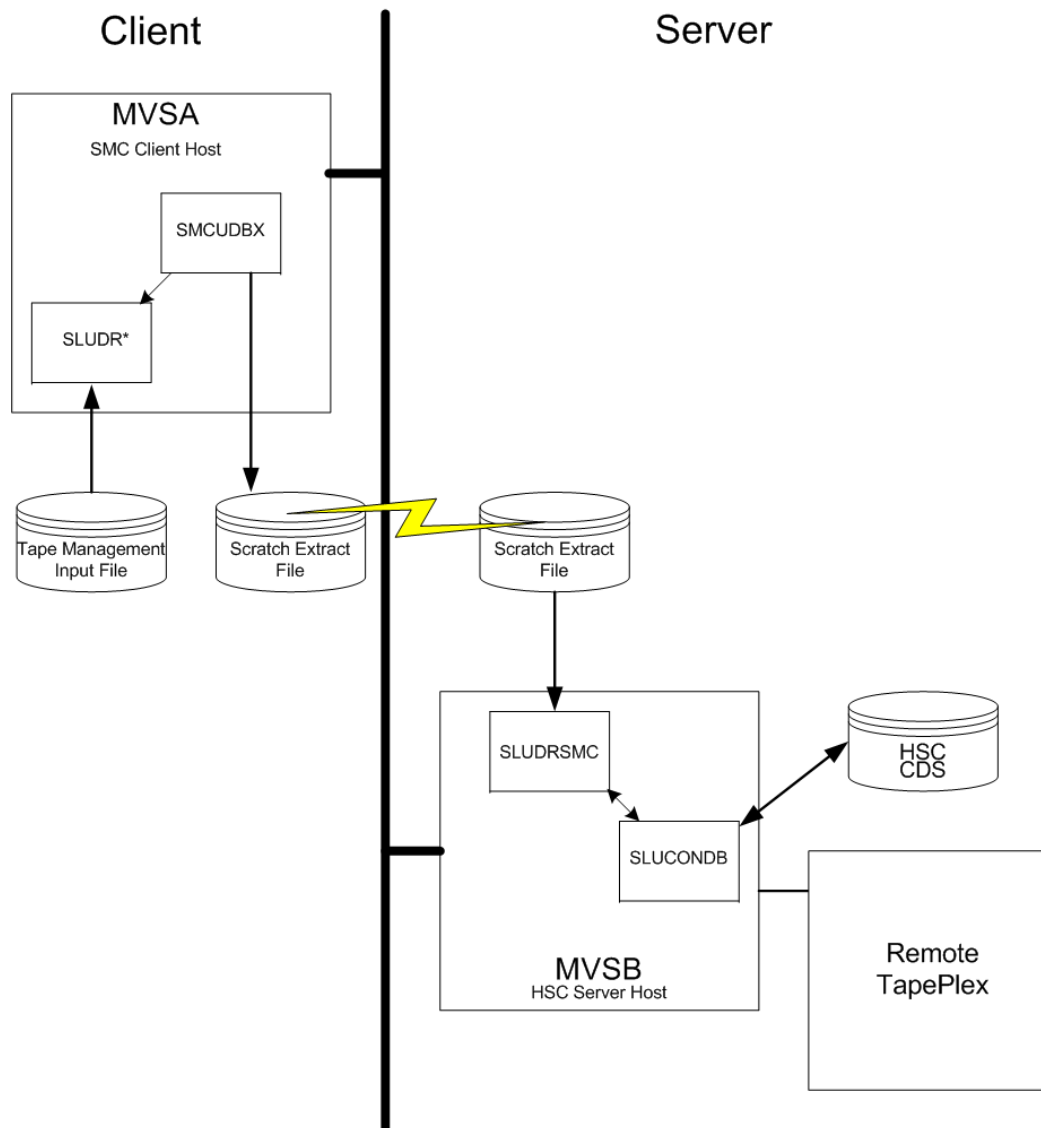
---

**Note:** See [Appendix C, "Tape Management System Interface \(SLUHDR\\*\) Routines"](#) for more information about these routines.

---

---

The following figure shows The SMCUBDX process:

**Figure 4–2 Scratch Synchronization Between SMC Client Host and HSC CDS**

figure

If the client and server share a tape management system, the HSC utility `SLUCONDB` may be run directly using the common tape management system database as input.

If, however, the client host has a tape management system that is not accessible to the HSC host, the SMC utility `SMCUBX` can be used to extract scratch volumes from the tape management system using label and date parameters. `SMCUBX` output can then be used as input on the HSC host to the `SLUCONDB` utility. The SMC process supports timing synchronization by comparing the GMT time of the `SMCUBX` run with the last select time of the HSC CDS volume record. If the volume was selected after the SMC scratch run, it is not eligible for scratch by `SLUCONDB`.

## Running the SMC Scratch Extract Utility

You can run the SMC scratch extract utility regardless of whether the SMC is executing on the host.



SMCUBDX processes the CA-1 Tape Management Catalog (TMC), the CA-DYNAM/TLMS Volume Master File (VMF), the DFSMSrmm report extract file, or the Zara database, and selects the volume serial numbers for processing based on information specified on the PARM parameter of the JCL EXEC statement.

For CA-1, CA-DYNAM/TLMS, and DFSMSrmm, SMCUBDX calls the appropriate Tape Management System interface routine. For Zara, SMCUBDX invokes the SLUDRZAR routine, which in turn invokes an application programming interface (API) to read Zara's database. The data returned from these routines is used to format the SMC scratch extract file. See [Appendix C, "Tape Management System Interface \(SLUDR\\*\) Routines"](#) for more information about these routines.

If your installation uses a different tape management system, you can develop a program modeled on one of the supplied examples SLUDRCA1, SLUDRRMM, SLUDRTLM, SLUDRSMC, SLUDRTLM, or SLUDRZAR, or you can write your own program to produce an extract file in the following format:

**Table 4-5 Tape Management Subsystem Extract File Format**

Position	Field	Format
1	Scratch volume	CL6
7	GMT of run	XL8
15	Unused	CL2

The time filed in the SMC scratch extract file is a character representation of the first four bytes of the output of the MVS STCK macro. For example, if the STCK macro produces a result of X'B0912345 6789ABCD', the GMT field of the SMC extract contains the character value C'B0912345'.

SMCUBDX, SLUDRCA1, SLUDRTLM, SLUDRRMM, and SLUDRZAR are provided as load modules. You can use the current versions of these load modules with CA-DYNAM/TLMS Release 5.4 and above, CA-1 Release 5.0 and above, DFSMSrmm 1.1, and Zara 1.2. The ELS MACLIB file on the ELS installation tape contains the sample source code.

---



---

**Note:**

- If there are local modifications to your tape management system, you must reassemble the associated TMS interface routine.
  - If your tape management system is CA-1 Release 4.9 or below, you must modify and reassemble the SLUDRCA1 routine. See ["Using SMCUBDX with CA-1 Release 4.9 or Below"](#) on page 4-24 for information about modifying and reassembling this routine.
  - If your tape management system is Zara, the SMCUBDX utility must be able to access the data set that contains the Zara API module (ZARAAPI1). You can either store this module in the LINKLIST, or include a JOBLIB or STEPLIB DD statement in the SMCUBDX JCL that points to the library containing the module.
  - Any routine that you modify and reassemble must be re-linked as non-reentrant (NORENT parameter) with AMODE=24 and RMODE=24.
- 
-

## Using SMCUBX with CA-1 Release 4.9 or Below

To use SMCUBX with CA-1 Release 4.9 or below, you must modify and reassemble the SLUDRCA1 routine. The CA-1 MACLIB containing the TMRECORD MACRO must be accessible during the assembly.

Perform the following steps for CA-1 Release 4.9:

1. Delete the asterisk in column 1 of the following line:

```
TMRECORD
```

2. Add an asterisk in column 1 of the following line:

```
TMMTMREC
```

3. Reassemble the SLUDRCA1 routine.

## Syntax

SMCUBX uses program parameters to determine the Tape Management System interface (SLUDR\*) routine to invoke and method of operation. See [Appendix C, "Tape Management System Interface \(SLUDR\\*\) Routines"](#) for more information about these routines.

You can specify the following values for the PARM parameter on the JCL EXEC statement when running the SMCUBX utility:

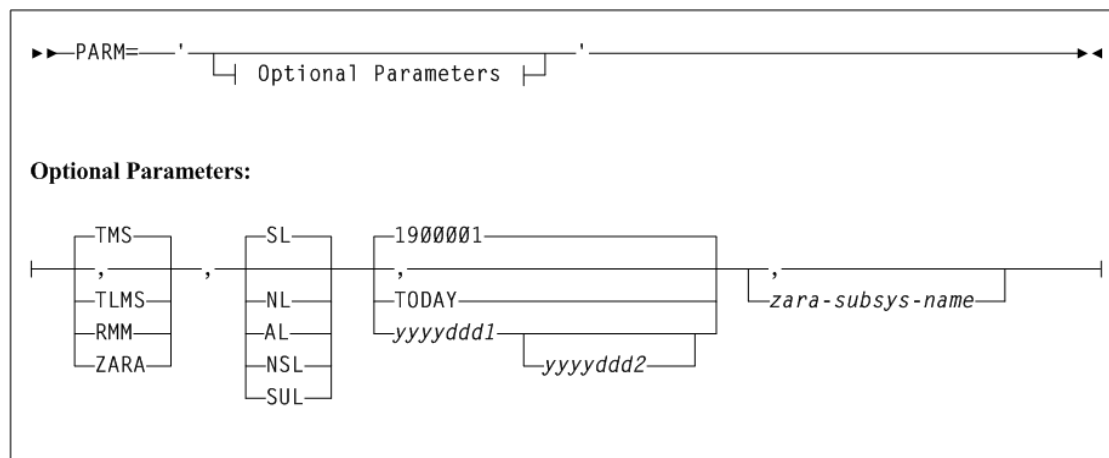
---

**Note:** You must enter parameters in the specified order and include commas for omitted parameters.

---

The following figure shows syntax for the SMCUBX utility:

**Figure 4-3 SMCUBX PARM parameter syntax**



## Parameters

### PARM=

specifies the parameters that follow can be accessed by the program indicated in the PGM parameter in the JCL.

**TMS**

optionally, specifies the CA-1 tape management system (TMS). This is the default.

**TLMS**

optionally, specifies the TLMS: CA-DYNAM/TLMS tape management system.

**RMM**

optionally, specifies the DFSMSrmm tape management system.

**ZARA**

optionally, specifies the ZARA tape management system.

**SL**

optionally, specifies standard label type for the scratch volume selection. This is the default.

**NL**

optionally, specifies nonlabeled type for the scratch volume selection.

**AL**

optionally, specifies ANSI label type for the scratch volume selection.

**NSL**

optionally, specifies nonstandard label type for the scratch volume selection.

**SUL**

optionally, specifies standard user label type for the scratch volume selection.

---



---

**Note:** The following date-related parameters specify the date or date range when a volume became scratch in the TMC.

---



---

## Date Related Parameters

The following date-related parameters specify the date or date range when a volume became scratch in the TMC.

**1900001**

optionally, specifies January 1, 1900 as the date. This is the default.

**TODAY**

optionally, indicates the current system date. Only volumes placed in scratch status on this date are processed.

**yyyyddd1**

optionally, the "from" date. The "to" date defaults to the current system date. All specified dates are in Julian notation.

All volumes placed in scratch status starting on day *yyyyddd1* through today's date are processed.

**yyyyddd1-yyyyddd2**

optionally, the date range in which the volumes became scratch. Neither of these dates can be greater than the current system date.

All volumes placed in scratch status starting on day *yyyyddd1* through day *yyyyddd2* are processed.

**zara-subsys-name**

Specifies the subsystem name for the Zara tape management system, where *zara-subsys-name* is the four-character name. The default subsystem name is ZARA.

**JCL Requirements**

The following definitions apply to the SMCUDBX JCL:

**DBTMS**

Input to the Scratch Extract utility in the form of the CA-1 TMS, the CA-DYNAM/TLMS VMS, or DFSMSrmm report extract file. This DD statement is not required for Zara.

**STDOUT**

Output messages from the Scratch Extraction utility. This DD statement is required.

**DBEXTFIL**

Output file from the Scratch Extract Utility. If a DCB LRECL is specified, it must specify a 16-byte value.

If Zara is the tape management system, the Zara API module must either reside in the MVS LINKLIST, or you must include one of the following JCL statements to access the module.

```
//STEPLIB DD DISP=SHR, DSN=your.zaraapi1.library
           DISP=SHR, DSN=your.tmsextract.library
```

```
//JOB LIB DD DISP=SHR, DSN=your.zaraapi1.library
```

The following example shows sample JCL that invokes the Scratch Extraction utility to process the CA-DYNAM/TLMS VMF for all tapes scratched in 2003.

**Example 4-7 Sample JCL for Scratch Extraction Utility**

```
//yourstd JOB card
//STEP0 EXEC PGM=SMCUDBX, PARM='TLMS, ,2003001-2003365'
//STEPLIB DD DISP=SHR, DSN=your.sea.linklib
// DD DISP=SHR, DSN=your.SLUDRxxx.linklib
//DBTMS DD DISP=SHR, DSN=your.tlms.database
//DBEXTFIL DD DISP=(NEW,KEEP), DSN=scratch.extract.file,
// SPACE=(TRK,(5,1)), UNIT=SYSDA
//STDOUT DD SYSOUT=*
/*
```

**Output Description**

SMCUDBX output includes the following:

- Messages associated with error conditions resulting from an unsuccessful attempt to execute scratch extraction processing
- Messages indicating errors or actions occurring during processing
- Output data set containing Scratch Update utility control statements

The following example shows sample output from the SMCUDBX utility.

**Example 4-8 Scratch Extraction Utility Sample Output Messages**

```
SMCUDBX 7.3.0      SMC TMS DB SCRATCH EXTRACT
```

Page 0001

TIME 15:26:41

Date 2015-08-14

Total records extracted for scratch volumes = 977

## SMC UI Simulation Utility (SMCUSIM)

The `SMCUSIM` utility is used to issue SMC UI commands, including the `SMC SIMulate` command, in a test environment. Changes to SMC command settings, tape policies, and allocation user exits can be tested before they are put into production without affecting existing SMC clients and HSC servers.

`SMCUSIM` code executes similarly to the SMC subsystem, with the following exceptions:

- It does not initialize the MVS SSI allocation interfaces.
- It does not set up any command or message handling intercepts.
- It does not utilize any CSA storage for processing.

The `SMCUSIM` utility runs as a batch program, and can execute on the same host as the existing SMC subsystem without interfering with SMC client operations. `SMCUSIM` processing occurs in two phases.

### Phase 1 Processing

In this phase, the utility operates similarly to the SMC subsystem startup routine. SMC modules are loaded and the `SMCPARMS` and `SMCCMDS` data sets are read.

Similarly to SMC subsystem startup, the `SMCPARMS` and `SMCCMDS` data sets are optional. However, as `SMCUSIM` does not include the SMC subsystem command intercept, you must specify any commands required to set up your test environment during this phase.

---

---

**Note:**

- Follow the guidelines in ["Specifying SMC Commands in the SMCCMDS or SMCPARMS Data Sets"](#) on page 1-3 for commands to include in the `SMCPARMS` and `SMCCMDS` data sets.
  - See [Chapter 2, "SMC Commands, Utilities, and Control Statements"](#) for more information about each individual command.
- 
- 

The `SMCUSIM` utility processes `TAPEPLEX` and `SERVER` commands as if it were an active SMC subsystem, and TCP/IP services may be required. Therefore, the security administration considerations for communication described in Oracle's ELS publication *Configuring and Managing SMC* do apply.

The following SMC commands are disabled by the `SMCUSIM` because they may interfere with the actual SMC subsystem or HSC server:

- `CMDDef`
- `HTTP`
- `LOG`
- `Route`

These commands can be included in the `SMCPARMS` or `SMCCMDS` data sets, but are ignored by the utility.

## Phase 2 Processing

In this phase, the utility reads and processes the `UUIIN` data set similarly to the `SMCUUUI` utility, except that only SMC commands are permitted. These commands are targeted to the `SMCUSIM` simulated environment, and have no effect on the actual SMC subsystem.

You can include SMC `SIMulate` commands, or any other SMC command in the `UUIIN` data set to test how a job with the specified parameters would be processed using the `SMCPARMS` and `SMCCMDS` data sets configured in phase 1. HSC UUI commands are not valid, and are flagged in error. Several types of output can be received, including:

- Plain text output
 

This type of output is written to `TEXTOUT`, and corresponds to the normal output produced by the SMC command.
- Structured XML
 

This type of output is written to `XMLOUT`. Structured XML is a well formed XML data stream with XML header tags and XML data tag elements, and XML data element content. Only specific SMC commands produced structured XML. Refer to the *ELS Programming Reference* for a complete list of SMC XML tag names and types.
- Comma Separated Values (CSV)
 

This type of output is written to `CSVOUT`. An input template of XML data tag names is used to define the format of the `CSVOUT` corresponding to a particular request. The output data for each request is a CSV header line that contains a comma-separated list of the XML data tags requested, followed by CSV detail lines. Each CSV detail line contains one comma-separated entry for each requested XML data tag. This type of output is readily adaptable as input into spreadsheets or customized report writers. Refer to the *ELS Programming Reference* for more information about the format for specifying CSV.

The type of output received is controlled by the data set definition statements (DD) defined in the JCL to execute the `SMCUSIM` utility.

## Sample JCL

The following example shows sample JCL that executes the `SMCUSIM` utility:

### Example 4-9 JCL to Execute the SMCUSIM Utility

```
//yourstd JOB card
//STEP EXEC PGM=SMCUSIM, PARM='pgmparms'
//STEPLIB DD DISP=SHR, DSN=your.sea.linklib
//          DD DISP=SHR, DSN=your.sea.exitlib
//*****
/* The following data sets are input to phase 1 and are optional
//*****
//SMCPARMS DD DISP=SHR, DSN=parmlib_name(parm_member_name)
//SMCCMDS DD DISP=SHR, DSN=cmdlib_name(cmd_member_name)
//*****
/* The following data set is required
//*****
//STDOUT DD DISP=SHR, DSN=your.stdout.dataset
```

```

//*****
/* The following data set is input to phase 2 and is required
//*****
//UUIIN DD *
CSV and UUI commands
//*****
/* At least 1 of the following data sets must be specified
//*****
//CSVOUT DD DSN=yourcsv.output.dataset,DISP=SHR
//XMLOUT DD DSN=yourxml.output.dataset,DISP=SHR
//TEXTOUT DD DSN=yourtext.output.dataset,DISP=SHR

```

## Parameters

The following SMCUSIM utility parameters may be specified for *pgmparms* in the sample JCL:

### XMLCASE

Specifies the case of alphabetic data fields in XML and CSV output. Valid case formats include the following:

- M indicates mixed uppercase and lowercase. This is the default.
- U indicates uppercase.

### XMLDate(format)

Specifies the format of date fields in XML and CSV output. Valid date formats include the following:

- YYYYMONDD
- YYYY-MM-DD
- YYYY-MON-DD
- HEXSTCK (hex character STCK format)

The default format is *yyyymondd*. For example, 2005Oct14.

### NOHDR

optionally, specifies that *STDOUT* and *TEXTOUT* report headings and pagination carriage control are not produced. *NOHDR* and *LINECNT* are mutually exclusive.

### LINECNT

optionally, specifies *STDOUT* and *TEXTOUT* report pagination line count setting. *LINECNT* and *NOHDR* are mutually exclusive.

*nn* indicates the number of lines per page. Valid values are 10-99.

---



---

**Note:** Settings for *STDOUT* report headings and pagination can only be specified in the EXEC statement program parameters. Settings for *TEXTOUT* report headings, while initially specified by the EXEC statement program parameters, can later be modified by specifying *TEXTOUT UUI* control statements in the UUIIN command stream.

---



---

For example:

In the following figure, the EXEC statement program PARM='LINECNT(55)' sets the report pagination to 55 lines per page for both STDOUT and TEXTOUT output.

However, the UUI command TEXTOUT NOHDR in the UUIIN command stream alters the TEXTOUT report parameters to disable report headers and pagination.

```
//yourstd JOB card
//STEP EXEC PGM=SMCUSIM,PARM='LINECNT(55)'
//STEPLIB DD DISP=SHR,DSN=your.sea.linklib
// DD DISP=SHR,DSN=your.sea.exitlib
//SMCPARMS DD DISP=SHR,DSN=parmlib_name(parm_member_name)
//SMCCMDS DD DISP=SHR,DSN=cmdlib_name(cmd_member_name)
//STDOUT DD DISP=SHR,DSN=your.stdout.dataset
//UUIIN DD *
TEXTOUT NOHDR
UUI commands
//CSVOUT DD DSN=yourcsv.output.dataset,DISP=SHR
//XMLOUT DD DSN=yourxml.output.dataset,DISP=SHR
//TEXTOUT DD DSN=yourtext.output.dataset,DISP=SHR
```

### MAXRC

specifies whether SMCUSIM initialization is to be terminated when the specified command return code is exceeded. If MAXRC is not specified, then the SMCUSIM always attempts to complete its initialization regardless of any SMCPARMS or SMCCMDS command failure(s). This is the default behavior.

*nn* specifies the highest allowed return code. If an SMC command executed from the SMCPARMS or SMCCMDS data set exceeds this value, then SMC0236 and SMC0237 messages are produced, and the SMCUSIM utility terminates before any commands are read from the UUIIN DD. Allowable values are 0, 4, 8, and 12.

---

---

#### Note:

Certain SMC commands, such as CMDDef, HTTP, LOG, and Route, that are allowed in the SMC subsystem environment, but are not allowed in the SMCUSIM environment are merely ignored under SMCUSIM and do not affect MAXRC processing, if specified.

Only commands executed as part of the simulated SMC subsystem startup (that is, those read from the SMCPARMS and SMCCMDS data sets) are evaluated as part of MAXRC processing. Commands read from the UUIIN DD which result in a return code greater than the specified MAXRC value do not automatically terminate the SMCUSIM utility

---

---

### PLEXRC

specifies whether SMC subsystem initialization is to be terminated based on the status of TapePlexes returned from the automatically issued RESYNC command.

If PLEXRC is not specified, then the SMC subsystem will complete its initialization regardless of the outcome of the RESYNC command. This is the default behavior.

*n* specifies the highest allowed return code from the RESYNC command. Valid values are 0 and 4.

The SMC RESYNC command sets a return code of 8 if SMC is unable to communicate with any defined TapePlex, and a return code of 4 if SMC is able to communicate with one or more, but not all, defined TapePlexes.



**NOCC**

optionally, indicates that no printer carriage control is to be included. When this option is selected, any non-filled pages will be appended with blank lines for the specified `LINECNT`.

**Required Data Set Definition (DD) Statements**

You must specify the `UUIIN` and `STDOUT` data set definitions.

**UUIIN Data Set**

The `UUIIN` data set can include:

- SMC UUI commands
- HSC UUI commands
- VTCS PGMI requests

Native SMC UUI commands include:

- Display Drive
- Display Volume
- POLicy
- SIMulate

Refer to the *ELS Programming Reference* for a list of supported HSC UUI commands and VTCS PGMI requests.

**Syntax Rules**

The following syntax rules apply:

- Input must be 80 byte card images.
- Only columns 1-72 of the 80 byte card image may contain actual command input. Columns 73-80 may be used for sequence numbers if desired.
- If the last non-blank character in columns 1-72 is a "+", the logical command is continued on the next physical card.
- Comments begin with " /\*" (in column 2 and beyond) and end with "\*/".
- The maximum logical command length is 32000 bytes.

**Control Statements**

Utility control statements allow the report pagination for `TEXTOUT` output and the target TapePlex to be altered for subsequent SMC, HSC, and VTCS UUI and PGMI commands.

**TEXTOUT**

optionally, sets the report pagination parameters for `TEXTOUT` output. If any errors are encountered on the `TEXTOUT` command, all subsequent UUI commands are bypassed until a correct `TEXTOUT` command is encountered.

- `NOHDR` optionally, specifies that `TEXTOUT` report headings and pagination carriage control are not produced. `NOHDR` and `LINECNT` are mutually exclusive.
- `LINECNT(nn)` optionally, specifies `TEXTOUT` report pagination line count setting. `nn` is the number of lines per page. Valid values are 10-99. `LINECNT` and `NOHDR` are mutually exclusive.

---

**Note:**

- Settings for `STDOUT` report headings and pagination can only be specified in the `EXEC` statement program parameters. Settings for `TEXTOUT` report headings, while initially specified by the `EXEC` statement program parameters, can later be modified by specifying `TEXTOUT` `UII` control statements in the `UIIIN` command stream.
  - If multiple `TEXTOUT` statements are input before a single `UII` command, the last `TEXTOUT` statement controls the `TEXTOUT` report pagination.
  - `PLEXDext` and `TEXTOUT` commands can be specified along with `CSV` commands to alter the destination TapePlex, and the `CSV` and `TEXTOUT` output characteristics of subsequent `UII` commands.
- 

**STDOUT Data Set**

The `STDOUT` data set is where the `SMCUSIM` utility reports the input and final completion code for each `UIIIN` request.

**SMCUSIM Utility Sample CSV**

The following CSV sample displays Y/N values for major sources of tape policy for the specified simulated job step. Refer to the *ELS Programming Reference* for a complete list of XML tags and their structure.

```
csv field=(jobname,dsname,idax_matched_policy_flag,      +
idax_matched_tapereq_flag,idax_policy_esoteric_flag,    +
idax_storgrp_esoteric_flag,sms_info_flag,              +
policy_info_flag,tapereq_info_flag,                   +
userexit_info_flag,owning_tapeplex),                  +
break(simulate_request) notitle fixed
```

**Trace Format Utility (SMCUGTF)**

The `SMCUGTF` utility converts `SMC` `GTF` output created with the "compact" format (the default format) into the original `SMC` trace format.

**Sample JCL**

The following example shows sample `JCL` that executes the `SMCUGTF` utility.

**Example 4–10 JCL to Execute the SMCUGTF Utility**

```
// yourstd jobcard
//STEP      EXEC PGM=SMCUGTF
//STEPLIB DD DSN=your.sea.linklib,DISP=SHR
//GTFIN DD DSN=your.compact.gtftrace.dsn,DISP=SHR
//GTFOUT DD DSN=your.formatted.gtftrace.dsn,
// DISP=(NEW,CATLG,DELETE),UNIT=SYSDA,
// SPACE=(space parameters)
```

---



---

**Note:** Space required for the output GTF trace is approximately 6 times the size of the COMPACT input file.

---



---

## VSM console SMF Offload utility (SMCUSMF)

The SMC SMCUSMF utility enables you to offload SMF records from a VSM console server. The SMF records are placed in a z/OS data set with the following DCB attributes:

- recfm - VB
- lrecl - 27990
- blksize - 27994

Refer to the ELS publication *ELS Programming Reference* for information about these HSC/VTCS SMF records.

## Sample JCL

The following JCL sample executes the SMCUSMF utility:

### **Example 4–11 JCL to Execute the SMCUSMF Utility**

```
//jobname JOB (account),REGION=0M
//S1 EXEC PGM=SMCUSMF,PARM='pgmparms'
//STDOUT DD SYSOUT=*
//SMCSMF DD DISP=SHR,DSN=yoursmf.output.dataset
```

## Parameters

The following SMCUSMF utility parameters may be specified for *pgmparms* in the sample JCL:

### **NOHDR**

optionally, specifies that the *STDOUT* report headings and pagination carriage control are not produced.

### **LINECNT(nn)**

optionally, specifies *STDOUT* report pagination line count setting.

*nn* indicates the number of lines per page. Valid values are 10-99.

### **SERVer(ssss)**

specifies the name of the VSM console server from which SMF records are to offloaded.

*ssss* indicates the server name. The specified server must be the active server for the VSM console TapePlex. *SERVer* is a required parameter.

### **BEGIN(begin-date:begin-time)**

specifies the beginning of the period of SMF record creation. This is the server date and time. All dates and times are the local time as defined on the system providing the SMF records. *BEGIN* is a required parameter.

*begin-date* indicates the starting date, expressed in *yyyymmdd* format.

**yyyymmdd**

the beginning date.

**TODAY**

specifies TODAY as the beginning date.

**YESTERDAY**

specifies YESTERDAY as the beginning date.

*begin-time* indicates the starting time-of-day (24-hour value, expressed in *hhmmss* format). The allowable range for *begin-time* is 000000 to 235959. The default value is 000000.

**END(end-date:end-time)**

optionally, specifies the end of the period of SMF record creation. This is the server date and time. All dates and times are the local time as defined on the system providing the SMF records. When END is not entered, it defaults to the BEGIN date with a time of 235959.

*end-date* indicates the ending date, expressed in *yyyymmdd* format.

**yyyymmdd**

the ending date.

**TODAY**

specifies TODAY as the ending date.

**YESTERDAY**

specifies YESTERDAY as the ending date.

*end-time* indicates the ending time-of-day (24-hour value, expressed in *hhmmss* format). The allowable range for the *end-time* parameter is 000000 to 235959. The default value is 235959.

**SMFTYPE(nnn)**

optionally, specifies the SMF record type for the output records.

*nnn* indicates the record type. Valid values are 128 to 255. The default is 255.

## Required Data Set Definition (DD) Statements

The following DD statements are required:

**STDOUT**

The SMCUSMF utility reports final completion code and any error messages in the STDOUT data set.

**SMCSMF**

The SMCUSMF utility places SMF records in the SMCSMF data set. This is a variable blocked file. The DCB attributes must be:

```
DCB=(RECFM=VB,LRECL=27990,BLKSIZE=27994)
```

## SMCUSMF Usage

If the VSM console TapePlex is defined with two servers and each server has functioned as the active server for that TapePlex since the last run of SMCUSMF, you must run the following procedure to collect all SMF records from both servers.

With SERVER1 active and SERVER2 inactive:

1. Run SMCUSMF with SERVER (SERVER1) .

2. Disable SERVER1:

```
SERVER NAME (SERVER1) DISABLE
```

This command makes SERVER2 active.

3. Run SMCUSMF with SERVER (SERVER2) .

4. Re-enable SERVER1:

```
SERVER NAME (SERVER1) ENABLE
```

SERVER2 remains active but SERVER1 is eligible to become the active server if SERVER2 becomes inaccessible. To make SERVER1 the active server once again (if desired), disable SERVER2 and then enable it once SERVER1 has become the active server for the TapePlex.



---

## HSC Stand-Alone Utilities

This chapter describes HSC stand-alone utilities. These utilities run as independent programs.

The following utilities are included:

- Utility Administrator (SLUADMIN)
- SLICREAT
- Scratch Conversion Utility (SLUCONDB)

### Utility Administrator (SLUADMIN)

The SLUADMIN program operates as a batch job and controls initialization processing for most utility functions. The utility administrator (SLUADMIN) can accept multiple commands which are handled as serial requests for multiple utilities. If all requirements needed to execute the utility function are met, control is passed to the appropriate program which handles the actual processing.

Utility functions represented by additional control statements are not started until the prior utility function stops and control is returned to the SLUADMIN program.

---

**Caution:** The maximum length of a control statement is 32,767 characters. Exceeding this limit causes message SLS0241I to be displayed, which indicates an invalid utility control statement.

---

### Invoking SLUADMIN

For convenience, most utility programs are packaged to appear as a single, user-invoked program. An exception is Scratch Conversion. All other utility functions are invoked as a normal batch job using the SLUADMIN program and user-specified commands needed to invoke the required utility function.

Utility functions represented by additional commands are not started until the prior utility function stops and control is returned to the SLUADMIN program.

The following is a generic example of JCL that can be modified to execute SLUADMIN. Where “utility and CSV commands” appears in this sample JCL, enter the utility or CSV commands you wish to execute.

For example:

```
SET HOSTID(HSC2) FORHOST(HSCB)
OFFLoad LOGF
SCREdist ACS(01)
```

The following example shows JCL that invokes SLUADMIN:

**Example 5-1 JCL to Invoke SLUADMIN**

```
//jobname      job (account),programmer,REGION=0M
//S1 EXEC     PGM=SLUADMIN,PARM=MIXED
//STEPLIB    DD DSN=your.sea.linklib,DISP=SHR
//SLSPRINT   DD SYSOUT=A
//SLSIN DD    *
              utility and CSV commands
              .
              .
              .
//*****
/* Optional UUI Output Definition Statements *
//*****
//SLSCSV DD DSN=yourcsv.output.dataset,DISP=SHR
//SLSCSVX DD SYSOUT=*
//SLSXML DD DSN=yourxml.output.dataset,DISP=SHR
/*
//
```

---

---

**Note:** Specify a region size of 0M on the JOB statement for all utility functions.

---

---

### SLUADMIN Execution Parameters

The following optional parameters may be specified for PARM= on the SLUADMIN EXEC statement:

#### NOHDR

Suppresses printing of report headings. This parameter only affects programs that produce report output., such as ACTivities, VOLRpt, and VTVRpt, etc.

---

---

**Note:** NOHDR and LINECNT are mutually exclusive.

---

---

If NOHDR is not specified, the default is to print headings that include the following information:

- date/time of the report
- page numbers
- host software version number
- name of the utility function executed

#### NOPHDR

Suppresses printing of page headings, but produces control headers in the output, such as:

SLS0030I Message ID NNNN Help Info:

or

SLS0041I Command CCCCCC Help Info:



---

---

**Note:** NOPHDR is mutually exclusive with LINECNT and NOHDR.

---

---

This parameter is useful when you issue the `Display Cmd` or `Display Message` command to create a file that includes help text for commands or messages.

**MIXED**

Prints all report headings and messages in mixed case. The default is uppercase.

**LINECNT=nn**

Specifies the number of lines per page for the SLUADMIN report. Allowable values are 10 through 99; the default is 60 lines per page.

---

---

**Note:** LINECNT and NOHDR are mutually exclusive.

---

---

**SSYS(ssss)**

Optionally, specifies the name of the HSC subsystem for which the SLUADMIN utility is being run. This parameter is required when the HSC subsystem(s) running on the LPAR have specified the MULT parameter.

**XMLDate(date)**

Specifies the format of date fields in XML and CSV output. Valid date formats include the following:

- *yyyymondd*
- *yyy-mm-dd*
- *yyy-mon-dd*
- HEXSTCK (hex character STCK format)

The default format is *yyyymondd*. For example, 2005Oct14.

**XMLCase(case)**

Specifies the case of alphabetic data fields in XML and CSV output. Valid case formats include the following:

- M indicates mixed uppercase and lowercase. This is the default.
- U indicates uppercase

**SLUADMIN Definition Statements**

Additional statements may be required for certain HSC and VTCS commands. In this publication, these statements are described with the applicable command.

**General Definition Statements**

The following definition statements are always required:

- STEPLIB  
specifies the link library that contains the ELS modules.
- SLSPRINT  
specifies the destination for the utility output (plain text).
- SLSIN

specifies the input to the SLUADMIN (utility command names and parameters).

---

**Note:** SLSIN can also include the SDD command, See "[SDD Command](#)" on page 4-8 for more information.

---

- SLSOUT

specifies the destination for SLUADMIN processing messages.

### CDS Definition Statements

One or more of the following CDS definition statements are required for utility commands executed on a host where the HSC is inactive. These statements specify a primary CDS, and optionally, a secondary and standby CDS.

---

**Note:** These statements can be omitted for commands that execute on a host with HSC active. If these statements are omitted, the data sets used by the active HSC on the host are accessed.

---

#### SLSCNTL

specifies the primary CDS. This statement is required for utility commands executed on a host where the HSC is inactive.

#### SLSCNTL2

optionally, specifies the secondary CDS.

If a secondary CDS exists, include this statement so that if a switch occurs and the secondary CDS becomes active, the CDSs can be reordered to maintain database integrity.

#### SLSSTBY

optionally, specifies the standby CDS.

If a standby CDS exists, include this statement so that if a switch occurs and the standby CDS becomes active, the CDSs can be reordered to maintain database integrity.

When you specify CDS definition statements, the following conditions apply:

- The number of DD statements specified must equal the number defined by the SET TECHNIQUE command. When you run the SET TECHNIQUE command, the number of DD statements defined must equal the number implied by the new technique.
- Rules for selecting primary, secondary, and standby CDS versions are the same as those used by the HSC subsystem. Refer to the publication *Managing HSC and VTCS* for more information about CDS configuration.
- For utilities that are "read only" (for example, VOLRpt), you can specify a single DD (SLSCNTL) and no internal or external name validation is performed.

### UUI Output Definition Statements

One or more of the following definition statements are required when you wish to generate output types added through Unified User Interface (UUI) requests. See "[Requesting SLUADMIN Output](#)" on page 5-5 for more information about requesting these output types.

**SLSCSV**

specifies the destination for the utility output in Comma Separated Value (CSV) format.

**SLSCSVX**

specifies the destination for CSV exceptions.

**SLSXML**

specifies the destination for the utility output in structured XML format.

## Requesting SLUADMIN Output

In addition to plain text, SLUADMIN supports output types added through Unified User Interface (UUI) requests. Refer to the *ELS Programming Reference* for a list of commands and utilities supported by the UUI.

The output type is controlled by DD statements defined in the SLUADMIN JCL. Output types include the following:

- Plain text

Plain text output is always produced and is unchanged from previous releases. Output is written to the data set specified by the SLSPRINT definition statement.

- Structured XML

This output type is optional and is generated only if the SLSXML definition statement is specified. Output is written to the data set specified by this statement.

Structured XML is a well-formed XML data stream containing XML header tags, data tags, and data element content. Refer to the *ELS Programming Reference* for XML tag names and structure for all UUI requests.

- Comma Separated Values (CSV)

This output type is optional and is generated only if the CSV command and SLSCSV definition statement are specified.

CSV output can be customized. A template of XML data tag names can be input by the user to specify which data values will be output. The output data is a single CSV header line followed by CSV detail lines.

A CSV header line is a list of the XML data tag names requested; each name is separated by a comma. Each CSV detail line is a specified fixed number of data values, separated by a comma. This kind of output is readily adaptable as input into spreadsheets or customizable report writers.

### Requesting XML Output

If the SLSXML DD statement is present, it indicates that structured XML output should be provided for all SLUADMIN commands that support the UUI interface.

When XML is generated, exceptions are generated under their own header tag, <exceptions>, at the end of each request. Each exception is reported using the <reason> tag. Refer to the *ELS Programming Reference* for more information.

### Requesting CSV Output

If the SLSCSV DD statement is present, comma separated value (CSV) output is provided for all SLUADMIN commands that support the UUI interface.

If the `SLSCSVX` DD statement is present, CSV exceptions are reported. The `SLSCSVX` DD statement is valid only if the `SLSCSV` DD statement is provided. CSV exceptions are generated only if the CSV command is in effect for a request.

The default DCB information for this DD is the same as `SLSPRINT`, `LRECL=121`, `RECFM=FB`).

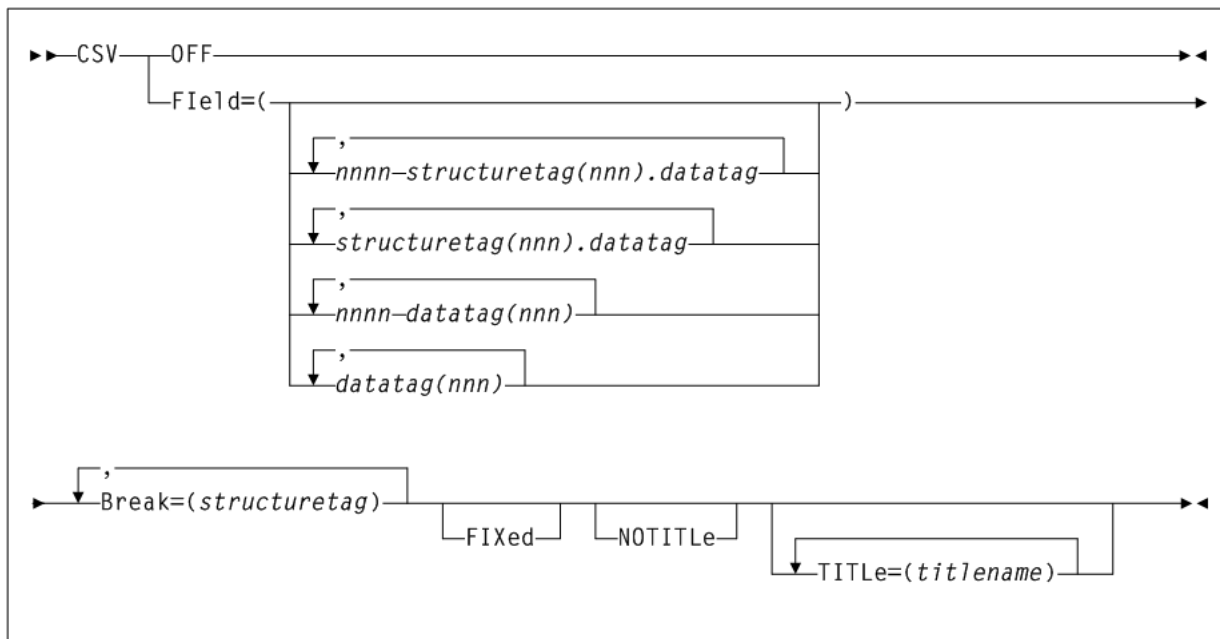
### CSV Command

The CSV command controls CSV output processing.

Each CSV command contains a list of CSV element definitions separated by commas. Each CSV element definition specifies an XML data tag name with optional XML structure tag and subscript values. Refer to the *ELS Programming Reference* for more information about ELS XML tags.

#### Syntax:

**Figure 5-1 CSV command syntax**



#### Parameters:

##### OFF

Disables CSV processing for subsequent requests.

---

**Note:** `OFF` is mutually exclusive with all other CSV parameters.

---

##### Field

Required if `OFF` is not specified.

- nnnn-** is an optional field name prefix containing 1 to 4 numeric characters followed by a dash. When a field name is prefixed by "*n* (*nnn*) -", the CSV output for the field always contains exactly the number of characters specified. Numeric fields are right-justified with the most left-most digits truncated; other fields are left-justified with the right-most characters truncated. If the value specified is

longer than the actual value, it is padded with blanks on the left for numeric fields and blanks on the right for other fields.

For example:

`8-acs_status` truncates the `acs_status` value to 8 characters.

`12-acs_status` blank-pads the `acs_status` value to 12 characters.

The maximum value of `nnnn` is 1024. The specification of "`nnnn-`" is valid only when the parameter `FIXed` is also specified.

- `structuretag(nnn).datatag` is an optional XML structure tag and data tag pair. An XML structure tag is an element that contains other elements but does not have content value. Refer to the *ELS Programming Reference* for more information about ELS XML tags.

Specify an XML structure tag/data tag pair only when the data tag occurs within multiple structure tags. For example, the `DISPLAY LSM` request data tag `free_cell_count` is used under the structure `lsm_data` to contain the total LSM free cell count and under the structure `panel_data` to contain the panel free cell count. Specifying `lsm_data.free_cell_count` requests the free cell count from the `lsm_data` structure; specifying `panel_data.free_cell_count` requests the free cell count from the `panel_data` structure. If no structure tag is coded for a data tag that occurs under multiple structure tags, the first occurrence of the data tag in the generated XML is used.

`nnn` is an optional subscript that determines how many of the multiple XML data values are included in the CSV output.

Specify a subscript only when the structured XML data for a single request contains multiple XML structure elements with the same name. This is indicated in the requestXML description whenever the column labelled "Maximum Occurrences" is greater than 1. A subscript value can be specified in BOTH the structure tag name AND the data tag name, for example:

```
CSV FIELD=(mvc_instance(4).volser,+
mvc_instance(4).partition_id(2))
```

creates CSV output with 12 fields, four `mvc_instance volser` occurrences, followed by two fields representing the `partition_id` values associated with the first MVC `volser`, then two fields representing the `partition_id` values associated with the second MVC, etc.

- `datatag(nnn)` is an optional XML data tag. If a data tag does not under more than one structure tag in a request, it is not necessary to specify a structure tag qualifier.

`nnn` is an optional subscript that determines how many of the multiple XML data values are included in the CSV output.

Specify a subscript only when the structured XML data for a single request contains multiple XML structure elements with the same name. This is indicated in the requestXML description whenever the column labelled "Maximum Occurrences" is greater than 1.

---



---

**Note:** The XML tag names are NOT case-sensitive and can be specified in any combination of upper and lower case characters.

---



---

**Break**

Required if `OFF` is not specified. The break tag must be a valid structure tag defined for the input request. Whenever a trailer tag matching the break tag is found in the output process, a new CSV record is created. Examples of valid break tags are:

- For `DISPLAY CDS`, `cds_data` or `host_data`
- For `VOLRPT` or `Display Volume`, `volume_data`

**Fixed**

Generates fixed format CSV output. Each field in the CSV output is blank-padded so that it occupies a fixed number of characters. This option can be used to create a customer-specified format flat file for reporting.

**NOTITLE**

Specifies that no CSV title line is produced. If this keyword is omitted, the first line of each CSV request is a title line containing comma separated tag names.

**TITLE**

If `TITLE` is specified, the title names specified are used in the title output line in place of the XML tag names. For any omitted title name, the specified XML tag name from the `FIELD` parameter is used in the title.

The following `SLUADMIN JCL` executes the `MVCRPT` command to produce a report of the percentage of space available for all MVCs in CSV format. After the job executes, you can FTP the CSV output file to your spreadsheet application.

Refer to the *ELS Programming Reference* for information about the structure and data tags used in this example.

**Example 5-2 CSV Output for the MVCRPT command**

```
//MVCREPT JOB x,CHRIS,CLASS=A,MSGCLASS=E
//*
//*
//*
//MVCRPT EXEC PGM=SLUADMIN,
// PARM='MIXED,XMLDATE=YYYY-MM-DD'
//STEPLIB DD DSN=your.sea.LINKLIB,DISP=SHR
//SLSCNTL DD DSN=your.sea.CLUSTER.CDS,DISP=SHR
//SLSPRINT DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SLSCSV DD DISP=(,CATLG),DSN=hlq.MVCREPT.CSV,
// DCB=(RECFM=VB,LRECL=4096,BLKSIZE=32000),
// UNIT=SYSDA,SPACE=(CYL,(x,y))
//SLSIN DD *
CSV FIELD=(mvc_data.volser,mvc_data.percent_available)+
BREAK=(mvc_data)+
NOTITLE
MVCRPT
/* /*
//
```

## SLUADMIN Program Return Codes

The `SLUADMIN` program sets a return code for the execution of the job step as described below.

The return code for each utility function is listed in a message. For return codes other than 0, additional messages are listed providing more information and description of the error condition. Refer to the publication *ELS Messages and Codes* for any messages encountered.

The SLUADMIN utility includes the following return codes:

- 0 indicates that all utility functions requested via control statements completed normally.
- 4 indicates that at least one utility function encountered an abnormal situation, but it did not cause termination of that utility function or any subsequent utility functions.
- 8 indicates that at least one utility function encountered an error condition that prevented continuation of that utility function, but any subsequent utility functions represented by control statements were attempted.
- 12 indicates that an error condition was detected that either prevented the start of any utility functions, or terminated the active utility function and prevented processing of subsequent utility functions.

---

---

**Note:** If the return code occurred because of a problem with the HSC CDS, the problem must be resolved before the HSC can be initialized safely.

---

---

## Reports Created by Utility Commands

Several utility commands produce reports when executed. These reports are in addition to the utility function. The following utility commands produce reports:

- LIBGen
- ACTivities
- AUDit
- BACKup
- MOVE
- VOLRpt

### Report Headings

Report headings include the following information:

- date/time of the report
- page number(s)
- host software version number
- name of the utility function executed

## SLICREAT

The SLICREAT program creates library control data sets and displays library volume capacity in a confirmation message.

These library control data sets are created based on the data set definitions you specify in the SLICREAT data set initialization JCL.

Using SLICREAT involves the following steps:

1. Create data set definitions in the data set initialization JCL using the SLICREAT program.
2. Execute the SLICREAT program to create the library control data sets.
3. Verify successful completion of the SLICREAT program and note the library volume capacity displayed in the confirming message.

These steps are described in the following sections. They must be performed before the library is operational.

---



---

**Note:**

- Before using SLICREAT, you must first create, assemble and link the LIBGen macros. Refer to the publication *Configuring HSC and VTCS* for more information.
  - You can use SLICREAT to reformat your HSC SLSJRNnn journal file(s). Refer to the *ELS Legacy Interfaces Reference* for more information.
- 
- 

## Creating Control Data Set Initialization JCL

Creating the definitions for the library control data set is done by coding JCL in a job file named SLICREAT.

The following is an example of JCL for the SLICREAT program. This sample is also included in the HSC SAMPLIB as member JCLCRT.

### Example 5-3 SLICREAT JCL

```
//SLICREAT JOB (account), 'programmer', CLASS=A
//CREATE EXEC PGM=SLICREAT, CDS CREATE MODULE
// PARM='libgen-load-module-name',
// REGION=0M
//*
//STEPLIB DD DSN=your.sea.linklib, DISP=SHR
//SYSPRINT DD SYSOUT=* MESSAGES
//*
//*****
//* LIBRARY PRIMARY CONTROL DATASET (CDS)
//*****
//SLSCNTL DD DSN=SLS.SLSCNTL, P RIMARY CDS
// SPACE=(4096,S,,CONTIG,ROUND), REPLACE 'S' WITH YOUR
// DISP=(NEW,CATLG,DELETE), SPACE CALCULATIONS
// UNIT=SYSDA
//*****
//* LIBRARY SECONDARY CONTROL DATASET (CDS)
//*****
//SLSCNTL2 DD DSN=SLS.SLSCNTL2, SECONDARY CDS
// SPACE=(4096,S,,CONTIG,ROUND), REPLACE 'S' WITH YOUR
// DISP=(NEW,CATLG,DELETE), SPACE CALCULATIONS
// UNIT=SYSDA
//*****
//* LIBRARY STANDBY CONTROL DATASET (CDS)
//***** /
//SLSSTBY DD DSN=SLS.SLSCNTL3, STANDBY CDS
// SPACE=(4096,S,,CONTIG,ROUND), REPLACE 'S' WITH YOUR
// DISP=(NEW,CATLG,DELETE), SPACE CALCULATIONS
```



```

//          UNIT=SYSDA
//*****
//*          LIBRARY JOURNAL DATASET (INITIAL)
//*****
//SLSJRN01 DD  DSN=SLS.SLSJRN01,          INITIAL JOURNAL
//              SPACE=(4096,u, ,CONTIG,ROUND),  REPLACE 'u' WITH YOUR
//              DISP=(NEW,CATLG,DELETE),        SPACE CALCULATIONS
//              UNIT=SYSDA
//***** /
//*          LIBRARY JOURNAL DATASET (ALTERNATE)
//*****
//SLSJRN02 DD  DSN=SLS.SLSJRN02,          ALTERNATE JOURNAL
//              SPACE=(4096,u, ,CONTIG,ROUND),  REPLACE 'u' WITH YOUR
//              DISP=(NEW,CATLG,DELETE),        SPACE CALCULATIONS
//              UNIT=SYSDA
//*****
//*          ONE PAIR OF JOURNALS ARE REQUIRED FOR EACH GENED HOST
//*****
//*

```

## DD Statements

The following JCL DD statements must be used to define the library control data sets:

### SYSPRINT

output messages.

Refer to the following criteria when allocating the control data sets:

- Do not specify the DCB statement. The HSC defines the required block size (4K).
- The data set(s) must be allocated in a single contiguous extent.
- The CDS(s) must not include existing data sets containing records prior to running the SLICREAT process.
- Oracle StorageTek recommends that the CDSs do not reside on the same volume(s) as other CDSs or other data sets that generate high I/O activity.

### SLSCNTL

primary library control data set.

This is the initial data set that is used to control and synchronize all activities within the HSC. It is used by the HSC until the controlling data set is switched to the secondary, either automatically or manually.

SLSCNTL also controls the SLICREAT process. If you do not include an SLSCNTL DD statement when you run SLICREAT, as follows:

- the CDSs will not be formatted, and
- the size requirement (in blocks) of the LIBGEN load module included in the PARM statement will be calculated and reported and journal formatting will be attempted.

### SLSCNTL2

secondary copy of the primary control data set.

This is required only if the SLIRCVRV macro TCHNIQE parameter is SHADOW, STANDBY, BOTH, or ALL.

**SLSSTBY**

standby control data set.

This data set is required only if the SLIRCVRY macro `TCHNIQE` parameter is `STANDBY` or `ALL`. A secondary data set is required if a standby control data set is specified. This data set is also required for `STBYONLY` processing (see "Using SLICREAT to Create Only the Standby CDS" on page 5-13).

## Executing the SLICREAT Program

Execute the SLICREAT program to create the library control data sets.

### SLICREAT EXEC Statement

Use the following EXEC statement to execute the SLICREAT program:

```
EXEC PGM=SLICREAT,PARM=' libgen-load-module-name '
```

The `PARM=` value indicates the LIBGEN load module name. The `STEPLIB` statement must point to the appropriate library in which the LIBGEN load module currently resides.

---

---

**Note:**

- SLICREAT requires a LIBGEN load module assembled with the same level macros as the release LINKLIB. A LIBGEN load module created with earlier HSC releases cannot be used as input to a later release of SLICREAT.
  - The CDS files that are created by SLICREAT (`SLSCNTL`, `SLSCNTL2`, and `SLSSTBY`) are verified to be empty data sets prior to being formatted. If SLICREAT identifies any of these files as containing records, an error message is displayed and the creation process terminates. If this occurs, specify a different data set(s) or delete and redefine the data set(s) and re-execute SLICREAT.
- 
- 

## Verifying Successful Completion of the SLICREAT Program

After the SLICREAT program is successfully processed, a message is generated indicating the status and the capacity of the library. An example is:

```
SLS0557I DATABASE SUCCESSFULLY INITIALIZED; TOTAL CARTRIDGE  
CAPACITY: DDDDDDDDDDD VOLUMES
```

### Calculating Cartridge Capacity - SL8500 and SL3000

Message SLS0557I displays the total cartridge capacity for the library. For the SL8500 and SL3000 libraries, you must first vary the library online (`Vary ACS` command) to see the actual capacity of the library rather than the maximum capacity of the library. Before you vary the library, the maximum capacity is based on the highest possible number of panels that the HSC allows to be defined, not the number you defined in LIBGEN.

After you vary the library online, issue the `Display Acs` or `Display Lsm` command to show the actual library capacity.

## Using SLICREAT to Create Only the Standby CDS

SLICREAT can be used to create an additional CDS. If the HSC was previously installed without a standby CDS, you can run the SLICREAT program to create one.

This method does not make a backup of the primary CDS. It formats the additional CDS and enables you to activate it with the CDS `Enable` command.

### SLICREAT EXEC Statement to Create Standby CDS

Specify `PARM='STBYONLY'` on the SLICREAT EXEC statement to create a standby CDS:

```
EXEC PGM=SLICREAT, PARM='STBYONLY'
```

---

---

#### Note:

- You must have a primary CDS before you can run the `STBYONLY` option of SLICREAT.
  - For a standby data set to be recognized by the system, it must be defined by a `CDSDEF` statement in the `PARMLIB`.
- 
- 

## Using SLICREAT to Calculate DASD Space

You can determine the minimum DASD space required for the CDSs by executing SLICREAT without specifying the `SLSCNTL` and `SLSJRNNn DD` statements. The HSC generates a message that indicates the minimum space requirements in 4096-byte blocks. When you execute SLICREAT, you must specify the number of blocks for each of these data sets that is at least as large as this minimum.

---

---

**Note:** If you are specifying multiple CDSs (`SLSCNTL2`, `SLSSTBY`), Oracle StorageTek recommends that you allocate the same amount of space (in blocks) for all your data sets when you define them.

---

---

If the data sets are defined with different space allocations, the HSC uses the size of the smallest data set to determine the number of 4K blocks that it will use for the CDS. The additional space in the other CDS data sets, if any, will not be used by the HSC.

The difference in space between the minimum space required (returned by SLICREAT) and the size of the smallest CDS copy is formatted as CDS free blocks.

## Scratch Conversion Utility (SLUCONDB)

The SLUCONDB program generates scratch transactions from CA-1, CA-DYNAM/TLMS, DFSMSrmm and Zara tape management systems, or from a file extracted from a client system by the SMC `SMCUDBX` utility.

---

---

**Note:** See "[Tape Management Extract Utility \(SMCUDBX\)](#)" on page 4-21 for more information about the `SMCUDBX` utility

---

---

SLUCONDB can execute as a standalone utility, generating scratch transactions for input to SLUADMIN. Alternatively, it can execute on a host with an active HSC system, and, using the `SLUCONDB LIBONLY` parameter, generate UUI scratch transactions sent directly to the active HSC system. In a client/server system, using the `TAPEPLEX` parameter,

SLUCONDB can access an HSC system on another host to acquire CDS data and to send UUI scratch transactions.

---

---

**WARNING:**

- **SLUCONDB should not be executed with any other SCRatch, UNSCRatch, or REPLaceall activity.**
  - **The SLUCONDB module is no longer distributed as source. If you wish to synchronize scratch processing with a tape management system not supported by SLUCONDB, alter one of the distributed SLUCONDB tape management interface modules to use the SLUDRINF interface format.**
- 
- 

## Input Formats

SLUCONDB uses program parameters to determine the Tape Management System interface (SLUDR\*) routine to invoke. See [Appendix C, "Tape Management System Interface \(SLUDR\\*\) Routines"](#) for more information about these routines.

You can supply SLUCONDB input values using the PARM parameter on the JCL EXEC statement when running the SLUCONDB utility. You can use either the following formats:

- Keyword Format
- Positional Format

Consider the following:

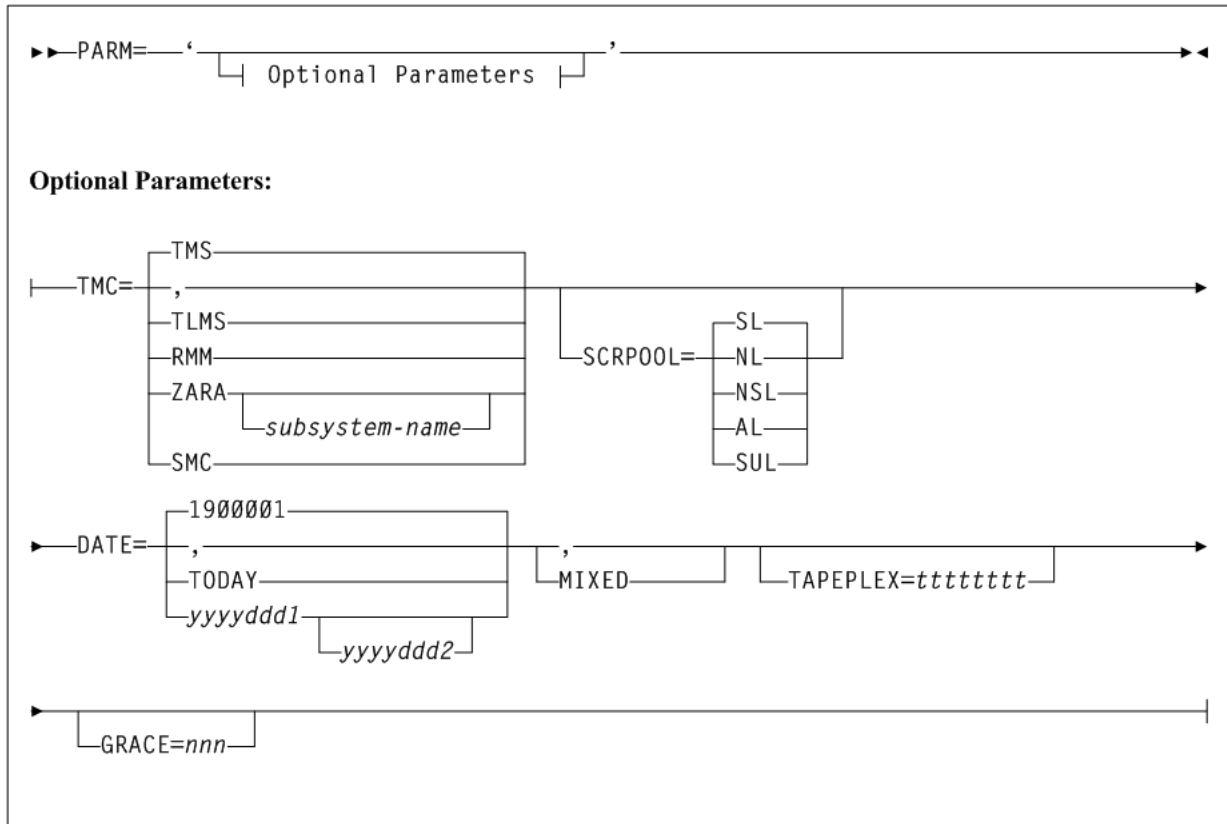
- The GRACE parameter is only permitted using the keyword format.
- The LIBONLY parameter is only permitted using the positional format.

### Keyword Format

The following section describes the syntax and parameters for keyword format of the PARM parameter.

### Syntax

Figure 5-2 SLUCONDB PARM parameter syntax (keyword format)



### Parameters

If you use the keyword format, adhere to the following syntax rules:

- Keywords and values must be separated by "=" (except for the MIXED keyword, which does not have a value).
- Extraneous commas between keywords are not allowed. For example, the following is invalid: `PARM='TMC=TMS, ,MIXED'`.
- Duplicate keywords are not allowed.

Valid parameters include the following:

#### PARM=

specifies that the parameters that follow can be accessed by the program indicated in the PGM parameter in the JCL.

The defaults for the PARM are:

- TMS
- All label types
- Scratch eligibility begins on January 1, 1900 and ends on the current system date
- Uppercase only
- Generate scratch transactions (no direct update).

#### TMC=

optionally, specifies the tape management system. For example:

- TMS specifies the CA-1 tape management system (TMS). This is the default.
- TLMS optionally, specifies the TLMS: CA-DYNAM/TLMS tape management system.
- RMM optionally, specifies the DFSMSrmm tape management system.
- SMC optionally, specifies the extract file from the SMC utility SMCUDBX.  
If SMC is specified, label and date parameters are not permitted. (Label and date selection are performed by the SMC utility when it creates the extract file.)
- ZARA optionally, specifies the ZARA tape management system.  
*subsystem-name* is an optional subsystem name.

**SCRPOOL=**

optionally, specifies the scratch pool label type used by the library. Valid label type values are:

- (SL) indicates standard label (the default).
- (NL) indicates non-labeled.
- (NSL) indicates non-standard label.
- (AL) indicates ANSI label.
- (SUL) indicates standard user label

**DATE=**

optionally, specifies the date or date range when a volume became scratch in the TMC.

- 1900001 optionally, specifies January 1, 1900 as the date. This is the default.
- TODAY optionally, indicates the current system date. Only volumes placed in scratch status on this date are processed.
- *yyyyddd1* optionally, the "from" date. The "to" date defaults to the current system date. All specified dates are in Julian notation.

All volumes placed in scratch status starting on day *yyyyddd1* through today's date are processed.

- *yyyyddd1-yyyddd2* optionally, the date range in which the volumes became scratch. Neither of these dates can be greater than the current system date.

All volumes placed in scratch status starting on day *yyyyddd1* through day *yyyddd2* are processed.

**MIXED**

optionally, specifies to print all the headings and messages in mixed case characters (uppercase and lowercase characters).

---

---

**Note:** If a comma is specified, or if this keyword is not specified, the system defaults to uppercase letters.

---

---

**TAPEPLEX=**

optionally, specifies the name of an SMC TapePlex to be used to extract CDS volumes and run UUI scratch commands.

*tttttttt* is the TapePlex name.

The specified TapePlex may be a local HSC executing on the same host as SLUCONDB, or may be an HSC subsystem on another host accessed via the SMC client server feature. Specifying the TAPEPLEX parameter requires an active SMC system on the SLUCONDB host, an active HSC system for the TapePlex, and requires that SLUCONDB be executed from an APF-authorized library.

#### GRACE=

optionally, specifies a grace period (in hours) for scratch eligibility. A volume accessed after the start of the SLUCONDB run, minus the grace period, is not eligible for scratch.

*nnn* is a 1-3 digit numeric value from 0 to 999, representing the grace period in hours.

#### Positional Format

The following section describes the syntax and parameters for positional format of the PARM parameter.

#### Syntax

---



---

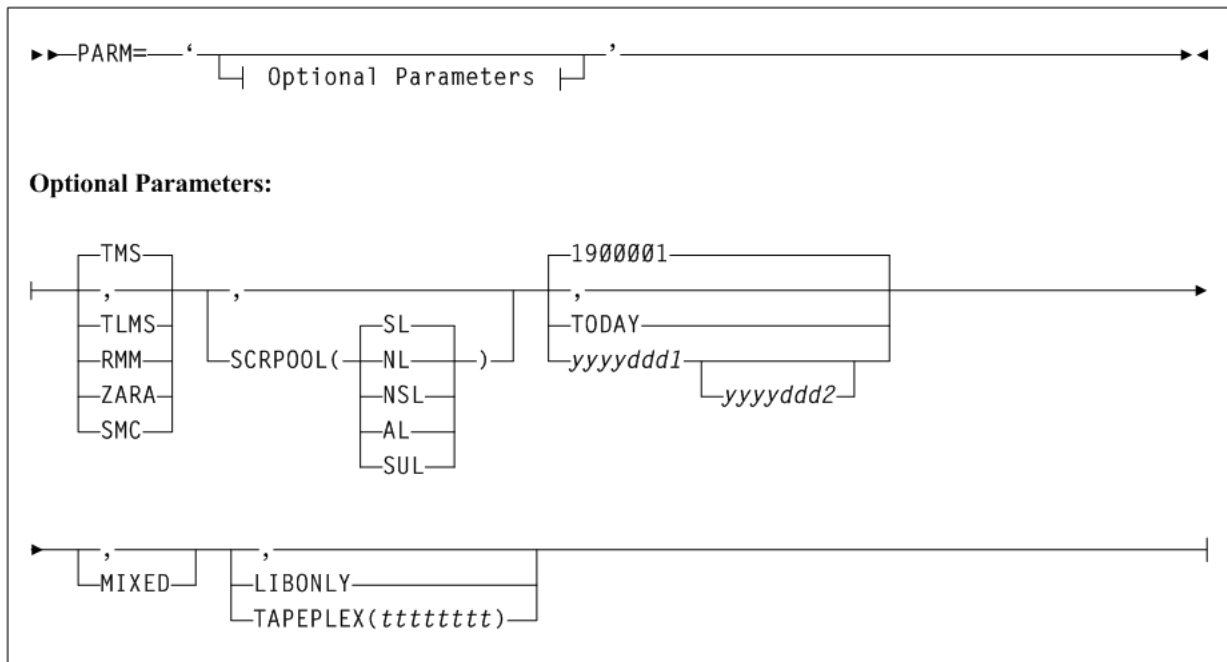
**Note:** You must enter parameters in the specified order and include commas for omitted parameters.

---



---

Figure 5-3 SLUCONDB PARM parameter syntax (positional format)



#### Parameters

##### PARM=

specifies that the parameters that follow can be accessed by the program indicated in the PGM parameter in the JCL.

The defaults for the PARM are:

- TMS
- All label types

- Scratch eligibility begins on January 1, 1900 and ends on the current system date
- Uppercase only
- Generate scratch transactions (no direct update).

The keywords for the `PARM` are positional. If you accept the system defaults, either do not specify `PARM`, or code "`PARM=`".

**TMS**

optionally, specifies the CA-1 tape management system (TMS). This is the default.

**TLMS**

optionally, specifies the TLMS: CA-DYNAM/TLMS tape management system.

**RMM**

optionally, specifies the DFSMSrmm tape management system.

**ZARA**

optionally, specifies the ZARA tape management system.

**SMC**

optionally, specifies the extract file from the SMC utility `SMCUDBX`.

If `SMC` is specified, label and date parameters are not permitted. (Label and date selection are performed by the SMC utility when it creates the extract file.)

**SCRPOOL**

optionally, specifies the scratch pool label type used by the library. Valid label type values are:

- `(SL)` indicates standard label (the default).
- `(NL)` indicates non-labeled.
- `(NSL)` indicates non-standard label.
- `(AL)` indicates ANSI label.
- `(SUL)` indicates standard user label

**Date Related Parameters**

The following date-related parameters specify the date or date range when a volume became scratch in the TMC.

**1900001**

optionally, specifies January 1, 1900 as the date. This is the default.

**TODAY**

optionally, indicates the current system date. Only volumes placed in scratch status on this date are processed.

**yyyyddd1**

optionally, the "from" date. The "to" date defaults to the current system date. All specified dates are in Julian notation.

All volumes placed in scratch status starting on day `yyyyddd1` through today's date are processed.



**yyyyddd1-yyyyddd2**

optionally, the date range in which the volumes became scratch. Neither of these dates can be greater than the current system date.

All volumes placed in scratch status starting on day *yyyyddd1* through day *yyyyddd2* are processed.

**MIXED**

optionally, specifies to print all the headings and messages in mixed case characters (uppercase and lowercase characters).

---



---

**Note:** If a comma is specified, or if this keyword is not specified, the system defaults to uppercase letters.

---



---

**LIBONLY**

optionally, specifies that volumes are to be placed in scratch status through the UUI interface. Specifying **LIBONLY** requires an active HSC, and requires that **SLUCONDB** be executed from an APF-authorized library

**TAPEPLEX**

optionally, specifies the name of an SMC tapeplex to be used to extract CDS volumes and run UUI scratch commands.

*tttttttt* is the TapePlex name.

The specified TapePlex may be a local HSC executing on the same host as **SLUCONDB**, or may be an HSC subsystem on another host accessed via the SMC client server feature. Specifying the **TAPEPLEX** parameter requires an active SMC system on the **SLUCONDB** host, an active HSC system for the TapePlex, and requires that **SLUCONDB** be executed from an APF-authorized library

## JCL Requirements

The following definitions apply to the **SLUCONDB** JCL:

**SLSTMS**

input to the utility in the form of either the CA-1 TMC, CA-DYNAM/TLMS Volume Master File (VMF), or **DFSMSrmm** report extract file. This statement is required.

**SLSSOUT**

output from the utility specifying the Scratch Update utility control statement. This statement is required if **LIBONLY** or **TAPEPLEX** is not specified.

If specifying a temporary or permanent data set for 'DSN=' in the JCL, you must include the appropriate DCB information (that is, **LRECL=72**).

**SLSPRINT**

output messages from the utility program. This statement is required.

---



---

**Note:** The Scratch Conversion utility uses the UUI to extract volume and VTV records from the CDS. The UUI requires read access to the CDS, either from statements in the JCL or by getting the data set information from an active HSC. The **SLSCNTL**, **SLSCNTL2**, and **SLSSTBY DD** statements are optional if an active HSC is running on the same system. The **SLSCNTL**, **SLSCNTL2**, and **SLSSTBY DD** statements are not allowed if either **LIBONLY** or **TAPEPLEX** parameters are specified.

---



---

**SLSCNTL**

the primary control data set. This statement is optional.

**SLSCNTL2**

the secondary control data set. This statement is optional.

**SLSSTBY**

the standby control data set. This statement is optional.

## JCL Examples

The following examples illustrate both generating scratch transactions and directly updating the HSC CDS.

The library referred to as "*your.sea.smp/e.loadlib*" in the following examples refers to the SEALINK library generated during the ELS installation. This library contains the TMS interface (SLUDR\*) modules. If you reassemble these modules, this library is the output of your link edit.

---

---

**Note:** The UNIT, SPACE, and DCB parameter settings are determined by the user.

---

---

### Example 1

The following is an example of SLUCONDB JCL to process the tape management databases, with active HSC, generating scratch update transactions:

```
//JOBSCUP JOB (account),programmer
//STEP1 EXEC PGM=SLUCONDB,
// PARM=('TLMS,,MIXED')
//STEPLIB DD DISP=SHR,DSN=your.sea.smp/e.loadlib
//SLSTMS DD DSN=TMS.MASTER.FILE,DISP=SHR
//SLSSOUT DD DSN=&&SCUPINPT,DISP=(NEW,PASS),
// UNIT=,SPACE=,DCB=
//SLSPRINT DD SYSOUT=*
//*
//STEP2 EXEC PGM=SLUADMIN,COND=(4,LT,STEP1),PARM=MIXED
//SLSIN DD DSN=&&SCUPINPT,DISP=(OLD,DELETE)
//SLSPRINT DD SYSOUT=*
//
```

### Example 2

The following is an example of SLUCONDB JCL to process the TMS database, with HSC CDS DD names, generating scratch update transactions:

```
//JOBSCUP JOB (account),programmer
//STEP1 EXEC PGM=SLUCONDB,
// PARM=('',2009100-2009110,MIXED')
//STEPLIB DD DISP=SHR,DSN=your.sea.smp/e.loadlib
//SLSTMS DD DSN=TMS.MASTER.FILE,DISP=SHR
//SLSSOUT DD DSN=&&SCUPINPT,DISP=(NEW,PASS),
// UNIT=,SPACE=,DCB=
//SLSPRINT DD SYSOUT=*
//SLSCNTL DD DSN=primary.control.dataset,DISP=SHR
//SLSCNTL2 DD DSN=secondary.control.dataset,DISP=SHR
//SLSSTBY DD DSN=standby.control.dataset,DISP=SHR
//*
```

```
//STEP2      EXEC PGM=SLUADMIN,COND=(4,LT,STEP1),PARM=MIXED
//SLSIN      DD DSN=&&SCUPINPT,DISP=(OLD,DELETE)
//SLSPRINT   DD SYSOUT=A
//
```

### Example 3

The following is an example of SLUCONDB JCL to process the DFSMSrmm database, with direct scratch updates (HSC active on host):

```
//JOBSCUP JOB (account),programmer
//STEP1      EXEC PGM=EDGHSKP,PARM='RPTEXT,DATEFORM(J) '
//MESSAGE    DD DSN=rmm.message.dataset,DISP=SHR
//REPTEXT    DD DSN=rmm.report.extract,DISP=SHR
//*
//STEP2      EXEC PGM=SLUONDB,
//            PARM=('RMM,,,MIXED,LIBONLY')
//STEPLIB    DD DISP=SHR,DSN=your.sea.smp/e.loadlib
//SLSTMS     DD DSN=rmm.report.extract,DISP=SHR
//SLSPRINT   DD SYSOUT=*
//
```

### Example 4

The following is an example of SLUONDB JCL to process the SMC extract file:

```
//JOBSCUP JOB (account),programmer
//STEP1      EXEC PGM=SLUONDB,PARM='SMC,,,MIXED,LIBONLY'
//STEPLIB    DD DISP=SHR,DSN=your.sea.smp/e.loadlib
//SLSTMS     DD DISP=SHR,DSN=your.smc.extract.file
//SLSPRINT   DD SYSOUT=*
//
```

### Example 5

The following is an example of SLUONDB JCL to process from SMC to a remote TapePlex:

```
//JOBSCUP JOB (account),programmer
//STEP1      EXEC PGM=SLUONDB,
//            PARM=',,,MIXED,TAPEPLEX(RODPLX) '
//STEPLIB    DD DISP=SHR,DSN=your.sea.smp/e.loadlib
//SLSTMS     DD DSN=TMS.MASTER.FILE,DISP=SHR
//SLSPRINT   DD SYSOUT=*
//
```

## Output Description

SLUONDB output includes the following:

- a listing of input commands with appropriate messages if syntax errors occur
- messages associated with errors in the SLUONDB process
- messages associated with error conditions resulting from an unsuccessful attempt to scratch a volume (using LIBONLY or TAPEPLEX parameters)
- scratch input transactions (if LIBONLY or TAPEPLEX is not specified).

The following example shows Scratch Conversion utility output, without LIBONLY or TAPEPLEX specified:

**Example 5-4 Scratch Conversion Utility output (without LIBONLY or TAPEPLEX)**

```
StorageTek Enterprise Library Software Scratch Conversion Utility Page 00001
Time 12:15:39 Date 2010-03-31
SLS4723I Volume (VTV) 400021 used after SLUCONDB start or TODATE
SLS0286I 000005 volumes have been selected
SLS0155I Condition code for utility function is 0
```

The following example shows Scratch Conversion utility output, with LIBONLY or TAPEPLEX specified:

**Example 5-5 Scratch Conversion Utility output (with LIBONLY or TAPEPLEX)**

```
StorageTek Enterprise Library Software Scratch Conversion Utility Page 00001
Time 12:15:39 Date 2010-03-31

SLS2321I Volume A00020 successfully added to VSM as scratch
SLS2321I Volume A00023 successfully added to VSM as scratch
SLS0167I Volume FB2110 successfully added to library as scratch
SLS0167I Volume TIM021 successfully added to library as scratch
SLS2280I Volume 400021 selected after SCRATCH command USETOKN
SLS1312I 000004 volumes successfully scratched
SLS0155I Condition code for utility function is 4
```

## VTCS Stand-Alone Utilities

This chapter describes VTCS stand-alone utilities. These utilities are not issued using the common utility administrator program (SLUADMIN).

### RTV Utility

The RTV utility converts VTVs contained on MVCs to data sets on volumes (real tape volumes). This utility is a standalone utility executed using the SWSRTV program.

Refer to the publication *Managing HSC and VTCS* for detailed information about when to use this utility.

---

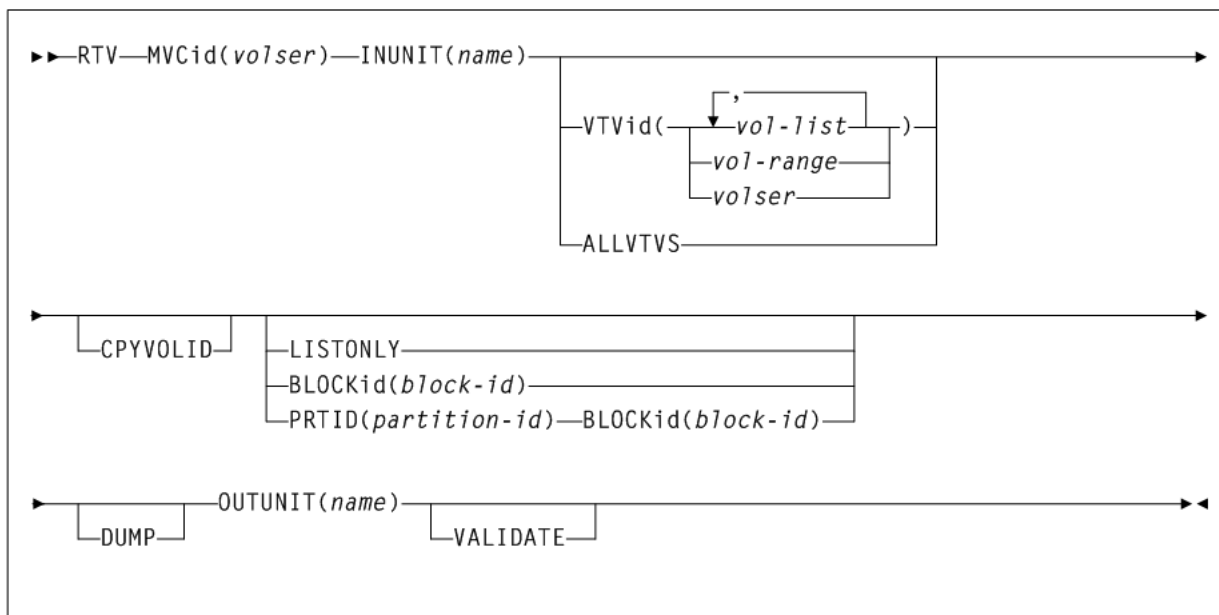
**Note:** The RTV utility **does not** support LTO Generation 6, 7, or 8 media and drives.

---

### Syntax

Figure 6–1 shows RTV utility syntax.

Figure 6–1 RTV utility syntax



## Parameters

As shown in [Figure 6-1](#), the RTV utility includes the following parameters:

### MVCid

specifies the MVC that contains the VTVs that RTV converts to volume(s) which become real tape versions of the VTVs.

*volser* indicates the MVC volser.

### INUNIT

specifies 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* indicates the unit name.

### VTVid

optionally, specifies one or more VTVs to convert.

*volser*, *vol-range*, or *vol-list* indicate the volsers of one or more VTVs.

### ALLVTVS

optionally, 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:** VTVid and ALLVTVS are mutually exclusive.

---



---

### CPYVOLID

optionally, copy the VTV internal volser from the VTV to the output volume VOL1 record. The default is to not copy the internal volser.

---



---

**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.
  - RTV supports VTVs created with standard or ANSI labels. If you do not specify CPYVOLID, RTV processes these label types as described in [Table 6-1](#). Note that this only applies to the VOL1 record. The HDR1/HDR2 labels are always copied from the VTV by RTV for every VTV processed.
- 
- 

The following table lists RTV label types:

**Table 6–1 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

In [Table 6–1](#), the WTOR is as follows:

SWSRTV - Label mismatch - Reply RELABEL, RETRY, or CANCEL

The operator responses produce the following results:

- RELABEL specifies to decompress the RTV and overwrite the volser on the output volume.
- RETRY specifies to mount another output volume and retry the operation.
- CANCEL specifies to not decompress the RTV.

#### **LISTONLY**

optionally, lists (but does not convert) the VTVs on the specified MVC. For more information, see "[RTV LISTONLY Listing](#)" on page 6-7.

#### **BLOCKID**

optionally, specifies the logical block ID where the VTV begins on the MVC.

*block-id* is the logical block ID (8 hexadecimal characters).

---



---

#### **Note:**

- BLOCKID and LISTONLY are mutually exclusive. In addition, if you specify the ALLVTVs parameter, or if a list or range of VTVs is specified, the BLOCKid parameter is 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.
  - The LISTONLY parameter listing on "[RTV LISTONLY Listing](#)" on page 777 supplies a Block ID value that you can use as input to the RTV utility to convert a VTV to a volume.
- 
- 

#### **PRTID**

optionally, specifies the partition id of the VTV being recovered from a partitioned MVC.

*partition-id* is the hexadecimal partition ID.

---



---

#### **Note:**

- If PRTID is specified, BLOCKid must also be specified.
  - PRTID and LISTONLY are mutually exclusive. In addition, if you specify the ALLVTVs parameter, or if a list or range of VTVs is specified, the BLOCKid parameter is ignored.
- 
-

**DUMP**

produce a S0C3 abend dump if RTV cannot decompress a VTV. If you specify `DUMP`, create a `SYSDUMP 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* is the unit name.

**VALIDATE**

Validates that the RTV utility can successfully process the specified VTV(s) without creating an output tape.

## JCL Requirements

The following are required or optional statements for the RTV utility JCL:

**STEPLIB**

specifies the link library (`SEALINK`) 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).

**SYSDUMP**

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 MVS Authorized Assembler Services Guide* for a description of the dynamic allocation reason codes.

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:** 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

**Explanation:** 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 statement, the VTV volser did not match that requested by the VTVid() parameter. The utility terminates with return code 12. Remove the BLOCKid statement and rerun the utility.

## RTV LISTONLY Listing

The following example shows the listing RTV produces when you specify the LISTONLY parameter.

### Example 6-1 RTV LISTONLY Listing

```
SWSRTV          (1.0.0)          StorageTek VTCS RTV Utility          PAGE 0001
TIME 14:28:33          Control Card Image Listing          DATE 12/01/15
RTV MVC(C83107) LISTONLY
```

```
SWSRTV          (1.0.0)          StorageTek VTCS RTV Utility          PAGE 0002
TIME 14:28:33          MVC C83107 Contents Report          DATE 12/01/15
```

VTV	File	Block	<---Created---		<---Migrated---		VTVPAGE	VTV
volser	#	ID	Date	Time	Date	Time	Size	Status
VV6825	1	00000000	2015Nov30	12:07:56	2015Nov30	12:15:56	Large	
VV6863	2	92005F0F	2015Sep27	12:57:54	2015Sep27	12:59:54	Large	
VV6893	3	92005F18	2015Aug18	08:57:26	2015Aug18	08:59:26	Large	
VV0403	4	92005F21	2015Aug18	08:57:26	2015Aug18	08:59:26	Large	

This report lists the following information for VTVs:

- Volser
- Logical file number on the MVC
- Block ID on the MVC
- Time created and migrated
- VTVPAGE size - STANDARD or LARGE
- Status - Not Current, or if blank, the VTV is current

## RTV Decompress Listing

The following example shows the listing RTV produces when you do not specify the LISTONLY parameter (that is, you run RTV to convert VTVs to volumes).

### Example 6-2 RTV Decompress Listing

```
SWSRTV          (1.0.0)          StorageTek VTCS RTV Utility          PAGE 0001
TIME 14:28:33          Control Card Image Listing          DATE 12/01/15
RTV MVC(C8228) VTV(VV6800-VV6900) CPYVOLID
```

```
SWSRTV          (1.0.0)          StorageTek VTCS RTV Utility          PAGE 0002
TIME 14:28:33          MVC C83223 Contents Report          DATE 12/01/15
```

VTV	File	Block	<---Created--->		<---Migrated--->		VTVPAGE	VTV
volser	#	ID	Date	Time	Date	Time	Size	Status
VV6070	1	00000000	2015Nov30	12:07:56	2015Nov30	12:15:56	Large	
VV0874	2	2B001384	2015Sep27	12:57:54	2015Sep27	12:59:54	Large	
VV0772	3	A3002707	2015Aug18	08:57:26	2015Aug18	08:59:26	Large	
VV6828	4	9B002AB9	2015Aug18	08:57:26	2015Aug18	08:59:26	Large	Not current
VV6828	5	9B002AC2	2015Aug18	08:57:26	2015Aug18	08:59:26	Large	
VV6826	6	9B002ACB	2015Aug18	08:57:26	2015Aug18	08:59:26	Large	

SWSRTV

```
SWSRTV          (1.0.0)          StorageTek VTCS RTV Utility          PAGE 0003
TIME 14:28:33          MVC C83223 Contents Report          DATE 12/01/15
```

VTV	Mounted	Final	Decompress
volser	Volser	Volser	Status
VV6826	XX0772	VV6826	Successful
VV6828	XX0773	VV6828	Successful

In addition to the contents fields shown in Figure 67 on page 136, the decompress listing shown in Figure 68 on page 137 lists the VTVs:

- Volser of the output volume as initially mounted
- Final volser of the output volume; if CPYVOLID is specified, the final volser will be identical to the VTV volser, otherwise is final volser is identical to the volser of the output volume as initially mounted
- Decompress status

---

## MEDia, RECTech, and MODel Values

This appendix provides values for MEDia, RECTech, and MODel parameters. These parameters enable you to specify transport and media characteristics. They are specified in various VM Client commands and TMI requests.

---

**Note:**

- The SL8500 library supports only the T9840A/B/C/D, T9940B, LTO, SDLT, and T10000A/B/C/D media types and recording techniques.
  - The SL3000 library supports only the T9840C/D, LTO, SDLT, and T10000A/B/C/D media types and recording techniques.
  - LTO and SDLT drives are not supported in an MVS environment. These drives are recognized by the HSC but are accessible only to open systems clients using LibraryStation.
  - LTO-6, LTO-7, and LTO-8 drives and media are supported in VSM 6 and VSM 7 only.
  - The VTCS RTV utility does not support LTO drives.
- 

### Media Type (MEDia)

The media type, or MEDia, enables you to specify the desired type of media to be used for a data set. It is specified in the following ELS commands and control statements:

SMC:

- POLicy operator command
- TAPEREQ control statement

HSC/VTCS:

- Display Drives command
- Display SCRatch command
- Display THReshld command
- Eject command
- SCREdist command
- Warn command
- VOLPARM control statement

The following table describes valid media types:

**Table A-1 Media Types**

Media Type	Description
LONGitud	Indicates any standard or enhanced (ECART) capacity cartridges.
ZLONGI	Indicates standard, enhanced (ECART), or extended-enhanced (ZCART) capacity cartridges.
Standard	Indicates a standard length, 3480 cartridge. It can be read on any longitudinal drive (4480, 4490, 9490, or 9490EE). Data can be written in 36-track mode on a 4490, 9490, or 9490EE drive but cannot be read on an 18-track (4480) drive. Synonyms include CST, MEDIA1, STD, 1, and 3480.
ECART	Indicates a 3490E, extended capacity cartridge. It can be used only on a 36-track drive (4490, 9490, or 9490EE). Synonyms include E, ECCST, ETAPE, Long, MEDIA2, and 3490E.
ZCART	Indicates a 3490E, extended-enhanced capacity cartridge. It can be used only on a 9490EE drive. ZCART can be abbreviated as Z
Virtual	Indicates a VTV (Virtual Tape Volume) mounted on a VTD (Virtual Tape Drive).
HELical	<p>Indicates a helical cartridge. A helical cartridge can be used only on RedWood drives. The following subtypes and abbreviations specify a helical cartridge:</p> <ul style="list-style-type: none"> <li>■ DD3 indicates any DD3A, DD3B, or DD3C helical cartridge.</li> <li>■ DD3A or A indicates a helical cartridge with a 10 GB media capacity</li> <li>■ DD3B or B indicates a helical cartridge with a 25 GB media capacity.</li> <li>■ DD3C or C indicates a helical cartridge with a 50 GB media capacity</li> </ul> <p>The seventh position in the external label is encoded with the cartridge type (that is, A, B, or C).</p>
STK1	Indicates any T9840 cartridge.
STK1R	<p>Indicates a T9840 data cartridge. The media indicator in the external label is encoded with the cartridge type (R). STK1R can be abbreviated as R.</p> <p>T9840 cartridge media capacities are 20 GB (T9840A and T9840B), 40 GB (T9840C), or 75 GB (T9840D).</p>
STK1U	Indicates a T9840A, T9840B, or T9840C cleaning cartridge. STK1U can be abbreviated as U.
STK1Y	Indicates a T9840D cleaning cartridge. STK1Y can be abbreviated to Y.
STK2	Indicates any T9940 cartridge.
STK2P	<p>Indicates a T9940 data cartridge. STK2P can be abbreviated as P.</p> <p>T9940 cartridge media capacities are 60 GB (T9940A) or 200 GB (T9940B).</p>
STK2W	Indicates a T9940 cleaning cartridge. STK2W can be abbreviated as W.
LTO-10G (CL)	Indicates an LTO 10 GB capacity cartridge.
LTO-35G (BL)	Indicates an LTO 35 GB capacity cartridge.

**Table A-1 (Cont.) Media Types**

<b>Media Type</b>	<b>Description</b>
LTO-50G (AL)	Indicates an LTO 50 GB capacity cartridge.
LTO-100G (1L)	Indicates an LTO 100 GB capacity cartridge.
LTO-200G (2L)	Indicates an LTO 200 GB capacity cartridge.
LTO-400G (3L)	Indicates an LTO 400 GB capacity cartridge.
LTO-400W (TL)	Indicates an LTO Generation 3 WORM cartridge. (Not supported in VTCS)
LTO-800G (4L)	Indicates an LTO 800 GB capacity cartridge.
LTO-800W (UL)	Indicates an LTO Generation 4 WORM cartridge. (Not supported in VTCS)
LTO-1.5T (5L)	Indicates an LTO 1.5 TB capacity cartridge.
LTO-1.5W (VL)	Indicates an LTO Generation 5 WORM cartridge. (Not supported in VTCS)
LTO-2.5T (6L)	Indicates an LTO Generation 6 2.5 TB cartridge.
LTO-2.5W (WL)	Indicates an LTO Generation 6 WORM cartridge. (Not supported in VTCS)
LTO-6.4T (7L)	Indicates an LTO Generation 7 6.4 TB cartridge.
LTO-6.4W (XL)	Indicates an LTO Generation 7 WORM cartridge. (Not supported in VTCS)
LTO-12T (8L)	Indicates an LTO Generation 8 12.8 TB cartridge.
LTO-12W (YL)	Indicates an LTO Generation 8 WORM cartridge.
LTO-CLN1 (1C)	Indicates an LTO type 1 cleaning cartridge.
LTO-CLN2(2C)	Indicates an LTO type 2 cleaning cartridge.
LTO-CLNU (UC)	Indicates an LTO universal cleaning cartridge.
SDLT	Indicates an SDLT 160 GB cartridge.
SDLT-2	Indicates an SDLT 125 GB cartridge.
SDLT-S1	Indicates an SDLT Generation S1 cartridge.
SDLT-S2	Indicates an SDLT Generation S2 cartridge.
SDLT-S3	Indicates an SDLT Generation S3 cartridge.
SDLT-S4	Indicates an SDLT Generation S4 cartridge.
SDLT-4	Indicates an SDLT Generation 4 cartridge.
T10000T1	Indicates a full-capacity 500 GB T10000A or 1 TB T10000B cartridge. T10000T1 can be abbreviated as T1.
T10000TS	Indicates a smaller-capacity 120 GB T10000A or 240 GB T10000B cartridge. T10000TS can be abbreviated as TS.
T10000CT	Indicates a T10000A or T10000B cleaning cartridge. T10000CT can be abbreviated as CT.
T10000T2	Indicates a full-capacity 5 TB T10000C or 8 TB T10000D cartridge. T10000T2 can be abbreviated as T2.
T10000TT	Indicates a smaller-capacity 1 TB T10000C or 1.6 TB T10000D cartridge. T10000TT can be abbreviated as TT.
T10000CL	Indicates a T10000A, T10000B, T10000C, or T10000D cleaning cartridge. T10000CL can be abbreviated as CL.

**Note:**

- T10000C and T10000D drives can read T10000T1 or T10000TS media but cannot write to that media.
- T10000C and T10000D drives can only write to T10000T2 or T10000TT media.

When the MEDia parameter is not specified, a default is chosen based on the value of the RECTech parameter. The following table shows default values used if MEDia is omitted:

**Table A-2 Media Type Defaults**

RECTech Entered	MEDia Default
18track	Standard
36track, 36Atrack, 36Btrack	LONGItud
36Ctrack	ZLONGI
LONGItud	LONGItud
DD3, Helical	DD3A
STK1R, STK1R34, STK1R35, STK1RA, STK1RA34, STK1RA35, STK1RB, STK1RB34, STK1RB35, STK1RAB, STK1RAB34, STK1RAB35, STK1RC, STK1RC34, STK1RC35, STK1RD, STK1RDE, STK1RDN, STK1RD34, STK1RD35, STK1RDE4, STK1RDE5	STK1R
STK2P, STK2P34, STK2P35, STK2PA, STK2PA34, STK2PA35, STK2PB, STK2PB34, STK2PB35	STK2P
T10K, T10KN, T10KE, T10KA, T10KAN, T1A34, T1A35, T10KAE, T1AE34, T1AE35, T10KC, T10KCN, T1C34, T1C35, T10KCE, T1CE34, T1CE35, T10KD, T10KDN, T1D34, T1D35, T10KDE, T1DE34, T1DE35	T10000T1

**Note:**

- T10000C and T10000D drives can read T10000T1 or T10000TS media but cannot write to that media.
- T10000C and T10000D drives can only write to T10000T2 or T10000TT media.



## Recording Technique (RECTech)

The recording technique, or RECTech, enables you to specify the method used to record data tracks on the tape surface for the desired data set. It is specified in the following ELS commands and control statements:

SMC:

- POLicy
- UNITAttr
- TAPEREQ

HSC/VTCS:

- Display Drives command
- Display SCRatch command
- Display THReshld command
- Eject command
- SCREdist command
- Warn command
- VOLPARM control statement

The following table describes valid recording techniques:

**Table A-3** Recording Techniques

Recording Technique	Description
LONGitud	Indicates any device that uses longitudinal recording. Devices include 4480, 4490, 9490, and 9490EE drives.
18track	Indicates a 4480 drive.
36track	Indicates a 4490, 9490, or 9490EE drive (any device that records in 36-track mode).
36Atrack	Indicates a 4490 (Silverton) drive.
36Btrack	Indicates a 9490 (Timberline) drive.
36Ctrack	Indicates a 9490EE drive.
HELical	Indicates a device using helical recording.
DD3	Indicates a device using helical recording.
LT02	Indicates any HP or IBM LTO Generation 2 drive.
LT03	Indicates any HP or IBM LTO Generation 3 drive.
LT04	Indicates any HP or IBM LTO Generation 4 drive.
LT05	Indicates any HP or IBM LTO Generation 5 drive.
LT06	Indicates any HP or IBM LTO Generation 6 drive.
LT07	Indicates any IBM LTO Generation 7 drive.
LT08	Indicates any IBM LTO Generation 8 drive.
STK1R	Indicates any T9840 drive.
STK1R34	Indicates a 3490E-image T9840 drive.
STK1R35	Indicates a 3590-image T9840 drive.

**Table A-3 (Cont.) Recording Techniques**

<b>Recording Technique</b>	<b>Description</b>
STK1RA	Indicates any T9840A drive.
STK1RA34	Indicates a 3490E-image T9840A drive.
STK1RA35	Indicates a 3590-image T9840A drive.
STK1RB	Indicates any T9840B drive.
STK1RB34	Indicates a 3490E-image T9840B drive.
STK1RB35	Indicates a 3590-image T9840B drive.
STK1RAB	Indicates any T9840A or T9840B drive.
STK1RAB4	Indicates a 3490E-image T9840A or T9840B drive.
STK1RAB5	Indicates a 3590-image T9840A or T9840B drive.
STK1RC	Indicates any T9840C drive.
STK1RC34	Indicates a 3490E-image T9840C drive.
STK1RC35	Indicates a 3590-image T9840C drive.
STK1RD	Indicates any T9840D drive.
STK1RDE	Indicates an encryption-enabled T9840D drive.
STK1RDN	Indicates a non-encryption-enabled T9840D drive.
STK1RD34	Indicates a non-encryption-enabled 3490E-image T9840D drive.
STK1RD35	Indicates a non-encryption-enabled 3590-image T9840D drive.
STK1RDE4	Indicates an encryption-enabled 3490E-image T9840D drive.
STK1RDE5	Indicates an encryption-enabled 3590-image T9840D drive.
STK2P	Indicates any T9940 drive.
STK2P34	Indicates a 3490E-image T9940 drive.
STK2P35	Indicates a 3590-image T9940 drive.
STK2PA	Indicates any T9940A drive.
STK2PA34	Indicates a 3490E-image T9940A drive.
STK2PA35	Indicates a 3590-image T9940A drive.
STK2PB	Indicates any T9940B drive.
STK2PB34	Indicates a 3490E-image T9940B drive.
STK2PB35	Indicates a 3590-image T9940B drive.
T10K	Indicates any T10000 drives.
T10KN	Indicates all non-encrypted T10000 drives.
T10KE	Indicates all encrypted T10000 drives.
T10KA	Indicates any T10000A drive.
T10KAN	Indicates a non-encryption enabled 3490E- or 3590-image T10000A drive.
T1A34	Indicates a non-encryption enabled 3490E-image T10000A drive.
T1A35	Indicates a non-encryption enabled 3590-image T10000A drive.
T10KAE	Indicates an encryption-enabled 3490E- or 3590-image T10000A drive.

**Table A–3 (Cont.) Recording Techniques**

<b>Recording Technique</b>	<b>Description</b>
T1AE34	Indicates an encryption-enabled 3490E-image T10000A drive.
T1AE35	Indicates an encryption-enabled 3590-image T10000A drive.
T10KB	Indicates any T10000B drive.
T10KBN	Indicates a non-encryption enabled 3490E- or 3590-image T10000B drive.
T1B34	Indicates a non-encryption enabled 3490E-image T10000B drive.
T1B35	Indicates a non-encryption enabled 3590-image T10000B drive.
T10KBE	Indicates an encryption-enabled 3490E- or 3590-image T10000B drive.
T1BE34	Indicates an encryption-enabled 3490E-image T10000B drive.
T1BE35	Indicates an encryption-enabled 3590-image T10000B drive.
T10KC	Indicates any T10000C drive.
T10KCN	Indicates a non-encrypted 3490E- or 3590-image T10000C drive.
T1C34	Indicates a non-encrypted 3490E-image T10000C drive.
T1C35	Indicates a non-encrypted 3590-image T10000C drive.
T10KCE	Indicates an encryption-enabled 3490E- or 3590-image T10000C drive.
TICE34	Indicates an encryption-enabled 3490E-image T10000C drive.
TICE35	Indicates an encryption-enabled 3590-image T10000C drive.
T10KD	Indicates any T10000D drive.
T10KDN	Indicates a non-encrypted 3490E- or 3590-image T10000D drive.
TID34	Indicates a non-encrypted 3490E-image T10000D drive.
TID35	Indicates a non-encrypted 3590-image T10000D drive.
T10KDE	Indicates an encryption-enabled 3490E- or 3590-image T10000D drive.
TIDE34	Indicates an encryption-enabled 3490E-image T10000D drive.
TIDE35	Indicates an encryption-enabled 3590-image T10000D drive.
Virtual	Indicates a VTV (Virtual Tape Volume) mounted on a VTD (Virtual Tape Drive).

When the RECTech parameter is not specified, a default is chosen based on the value of the MEDIA parameter. The following table shows default values used if RECTech is omitted.

**Table A–4 Recording Technique Defaults**

<b>MEDIA Entered</b>	<b>RECTech Default</b>
LONGitud	LONGitud
ZLONGI	LONGitud
Standard	LONGitud
ECART	36track

**Table A–4 (Cont.) Recording Technique Defaults**

<b>MEDia Entered</b>	<b>RECtech Default</b>
ZCART	36Ctrack
DD3A, DD3B, DD3C, DD3D	DD3
STKR, STK1U, STKY	STK1R
STK2P, STK2W	STK2P
T10000T1, T10000TS, T10000CL	T10K
T10000CT	T10KA and T10KB
T10000T2, T10000TT	T10KC and T10KD
Virtual	Virtual

## Model Type (MODE1)

The model type, or MODE1, enables you to specify the model number of a transport (drive), or drive. MODE1 provides the same type of information as RECtech, however, a user may find it more convenient to specify a transport model rather than a recording technique.

MODE1 is specified in the following ELS commands and control statements:

SMC:

- POLicy command
- UNITAttr command
- TAPEREQ control statement

HSC/VTCS:

- Display Drives command

---



---

**Note:**

- MODE1 and RECtech are mutually exclusive.
  - The SL8500 library supports only model types associated with T9840, T9940, and T10000 series drives.
  - You can specify multiple values for this parameter; separate each value with a comma.
- 
- 

The following table describes valid model types:

**Table A–5 Model Types**

<b>Model Type</b>	<b>Description</b>
4480	Indicates a 4480 (18-track) drive.
4490	Indicates a 4490 (36-track Silverton) drive.
9490	Indicates a 9490 (36-track Timberline) drive.
9490EE	Indicates a 9490EE (36-track Timberline EE) drive.
SD3	Indicates an SD-3 (RedWood) drive.

**Table A-5 (Cont.) Model Types**

<b>Model Type</b>	<b>Description</b>
9840	Indicates a 3490E-image T9840A drive.
984035	Indicates a 3590-image T9840A drive.
T9840B	Indicates a 3490E-image T9840B drive.
T9840B35	Indicates a 3590-image T9840B drive.
T9840C	Indicates a 3490E-image T9840C drive.
T9840C35	Indicates a 3590-image T9840C drive.
T9840D	Indicates a non encryption-enabled 3490E-image T9840D drive.
T9840D35	Indicates a non encryption-enabled 3590E-image T9840D drive.
T9840DE	Indicates an encryption-enabled 3490E-image T9840D drive.
T9840DE5	Indicates an encryption-enabled 3590E-image T9840D drive.
T9940A	Indicates a 3490E-image T9940A drive.
T9940A35	Indicates a 3590-image T9940A drive.
T9940B	Indicates a 3490E-image T9940B drive.
T9940B35	Indicates a 3590-image T9940B drive.
HP-LTO2	Indicates an HP LTO Generation 2 drive.
HP-LTO3	Indicates an HP LTO Generation 3 drive.
HP-LTO4	Indicates an HP LTO Generation 4 drive.
HP-LTO5	Indicates an HP LTO Generation 5 drive.
HP-LTO6	Indicates an HP LTO Generation 6 drive.
IBM-LTO2	Indicates an IBM LTO Generation 2 drive.
IBM-LTO3	Indicates an IBM LTO Generation 3 drive.
IBM-LTO4	Indicates an IBM LTO Generation 4 drive.
IBM-LTO5	Indicates an IBM LTO Generation 5 drive.
IBM-LTO6	Indicates an IBM LTO Generation 6 drive.
IBM-LTO7	Indicates an IBM LTO Generation 7 drive.
IBM-LTO8	Indicates an IBM LTO Generation 8 drive.
SDLT-600	Indicates a SuperDLT model 600 drive.
DLT-S4	Indicates a Quantum DLT-S4 drive.
T1A34	Indicates a non-encryption enabled 3490E-image T10000A drive.
T1A35	Indicates a non-encryption enabled 3590-image T10000A drive.
T1AE34	Indicates an encryption-enabled 3490E-image T10000A drive.
T1AE35	Indicates an encryption-enabled 3590-image T10000A drive.
T1B34	Indicates a non-encryption enabled 3490E-image T10000B drive.
T1B35	Indicates a non-encryption enabled 3590E-image T10000B drive.
T1BE34	Indicates an encryption-enabled 3490E-image T10000B drive.
T1BE35	Indicates an encryption-enabled 3590-image T10000B drive.
T1C34	Indicates a non-encryption enabled 3490E-image T10000C drive.

**Table A-5 (Cont.) Model Types**

<b>Model Type</b>	<b>Description</b>
T1C35	Indicates a non-encryption enabled 3590-image T10000C drive.
T1CE34	Indicates an encryption-enabled 3490E-image T10000C drive.
T1CE35	Indicates an encryption-enabled 3590-image T10000C drive.
T1D34	Indicates a non-encryption enabled 3490E-image T10000D drive.
T1D35	Indicates a non-encryption enabled 3590-image T10000D drive.
T1DE34	Indicates an encryption-enabled 3490E-image T10000D drive.
T1DE35	Indicates an encryption-enabled 3590-image T10000D drive.
Virtual	Indicates a VTV (Virtual Tape Volume) mounted on a VTD (Virtual Tape Drive).

---

## Library Identification

Each ACS, LSM, and CAP is assigned a unique identification number during LIBGEN. Use this number in ELS commands and utilities to identify a specific ACSid, LSMid, or CAPid.

### ACSid

Each ACS is identified by an ACSid (*acs-id*), a two digit decimal identifier from 00-99

The first ACS acquires an identifier of 00, the second ACS acquires an identifier of 01, and so forth until all ACSs are identified.

### LSMid

Each LSM is identified by an LSMid (*lsm-id*), consisting of an ACSid and LSM number separated by a colon. The LSMid differentiates each LSM from every other LSM in a library.

The format is AA:LL, where:

- AA is the ACSid, a two digit decimal identifier from 00-99.
- LL is the LSM number, a two digit decimal identifier from 00-99.

### CAPid

Each library CAP is identified by a CAPid (*cap-id*), consisting of an LSMid and CAP number separated by a colon. The format is AA:LL:CC, where:

- AA:LL is the LSMid.
- CC is the CAP number, a two digit decimal identifier from 00-11.

### CAP Numbers

The following are valid CAP numbers:

- 00
  - For 4410 and 9310 LSMs, standard 21-cell CAP or the right-hand 40-cell enhanced CAP
  - For 9740 LSMs, fixed rack 14-cell or 10-cell removable magazine CAP
  - For SL3000 LSMs, this is a 234-cell AEM CAP
  - For SL8500 LSMs, the CAP consists of 3, 13-cell removable magazines.

- For SL8500 LSMs, the Bulk CAP consists of 3, 12-cell removable magazines.
- **01**
  - For 4410 and 9310 LSMs, left-hand 40-cell enhanced CAP
  - For SL3000 LSMs, this is an optional CEM CAP consisting of 2, 13-cell removable magazines.
  - For SL8500 LSMs, this is an optional CAP consisting of 3, 13-cell removable magazines.
  - For SL8500 LSMs, the optional Bulk CAP consists of 3, 12-cell removable magazines.
- **02**
  - Priority CAP (PCAP) for a 4410 or 9310 LSM enhanced CAP
  - For SL3000 LSMs, this is an optional CEM CAP consisting of 2, 13-cell removable magazines.
  - For SL8500 LSMs, this is an optional CAP consisting of 3, 13-cell removable magazines.
- **03, 04, 05**
  - For SL3000 LSMs, these are optional CEM/DEM CAPs consisting of 2, 13-cell removable magazines.
- **06**
  - For SL3000 LSMs, this is the only required BDM CAP consisting of 2, 13-cell removable magazines.
- **07, 08, 09, 10**
  - For SL3000 LSMs, these are an optional CEM CAPs consisting of 2, 13-cell removable magazines.
- **11**
  - For SL3000 LSMs, this is a right-side AEM CAP, consisting of 18, 13-cell removable magazines.
  - For SL3000 LSMs, this is a 234-cell AEM CAP

## Panels, Rows and Columns

Certain ELS commands and utilities allow you to specify panels, rows, and columns within a library.

### Panels

Each *panel-list* element is a one or two digit decimal number, specific to the LSM type:

- LSM Models 4410 and 9310 PowderHorn
  - 0-11 for outer wall panels
  - 12-19 for inner wall panels
- LSM Model 9360 WolfCreek
  - 0-2 for Model 9360-050



- 0-3 for Model 9360-075
- 0-4 for Model 9360-100
- LSM Model 9740 TimberWolf
  - 0-2 (panel 3 is included if the optional cells are present).
- LSM Model SL3000 Library
  - Access Expansion Module (AEM optional) - panels 0-1 and 22-23 are located on both ends of the library and in this release serve only as library end caps.
  - Base Drive Module (BDM required) - panels 12-13
  - Drive Expansion Module (DEM optional) - panels 10-11 are located left of the BDM.
  - Cartridge Expansion Module (CEM optional) - panels 2-9 are located left of the BDM or optional DEM.
  - Cartridge Expansion Module (CEM optional) - panels 14-21 are located right of the BDM.
  - Parking Expansion Module (PEM) - Dual Robots only - panel numbers equate to the same panel numbers of the CEM it replaces, depending on its location right or left of the BDM.

---

**Note:** The SL3000 library has static panels numbers starting with the left-most back panel (panel 0) and ending with the right-most front panel (panel 23) on a fully-configured library. The panel numbering starts with the BDM, the only required module (panels 12 and 13), and works outward to the left and to the right.

---

- LSM Model SL8500 Library
  - Base library: 2-10
  - With one expansion panel: 2-23 (expansion panel is 8-15)
  - With two expansion panels: 2-26 (expansion panels are 8-23)
  - With three expansion panels: 2-34 (expansion panels are 8-31)
  - With four expansion panels: 2-42 (expansion panels are 8-39)
  - With five expansion panels: 2-50 (expansion panels are 8-47)

## Rows

Each row-list element is a one or two digit decimal number, specific to the LSM type:

- LSM Models 4410 and 9310 PowderHorn
  - 0-14 for outer wall panels
  - 0-5 and 8-14 for inner wall panels.
  - The maximum list is 4 rows (approximately 100 cells).
- LSM Model 9360 WolfCreek
  - 0-41 (all models). The maximum list is 20 rows (approximately 100 cells).
- LSM Model 9740 TimberWolf

- 0-41 on panels 0, 2, and 3
- 36-41 on panel 1

---

---

**Note:**

- Column 3 on panel 2 allows row entries only on rows 28-41.
  - The cells on panel 3 are optional.
- 
- 

- LSM Model SL3000 Library
  - Access Expansion Modules - panels 0-1 and 22-23 (no cells in this initial release)
  - Base Drive Module (BDM) - panel 12 (back)
    - \* 12-47 with 8 drives installed
    - \* 23-47 with 16 drives installed
    - \* 35-47 with 24 drives installed
  - Base Drive Module (BDM) - panel 13 (front)
    - \* 0-51 on the front panel
  - Drive Expansion Module (DEM) - panel 10 (back)
    - \* 12-47 available with 8 drives installed
    - \* 23-47 available with 16 drives installed
    - \* 35-47 available with 24 drives installed
    - \* No rows available with 32 drives installed
  - Drive Expansion Module (DEM) - panel 11 (front)
    - \* 0-51 available on the front panel
  - Cartridge Expansion Modules (CEM) - all panels
    - \* 0-51 available on both front and back panels
  - Parking Expansion Module (PEM) - all panels
    - \* 0-51 available on the back and front panels
- LSM Model SL8500 Library
  - 0-26 on a standard panel
    - \* 0-12 on a short panel (panels 4, 6-7)
    - \* 6-12 on a PTP panel (panel 5)

## Columns

Each column-list element is a one or two digit decimal number, specific to the LSM type:

- LSM Models 4410 and 9310 PowderHorn
  - 0-23 for outer wall panels
  - 0-19 for inner wall panels
- LSM Model 9360 WolfCreek

- 0-5 (all models).
- LSM Model 9740 TimberWolf
  - 0-3 for panels 0, 2, and 3 (if the optional cells are present)
  - 0-2 for panel 1
- LSM Model SL3000 Library
  - Access Expansion Module (AEM) panels 0-1 and 22-23
    - \* No columns available in the initial release
  - Base Drive Module (BDM) - panel 12 (back)
    - \* 0 available, if a CEM or DEM is added to the left of the BDM
    - \* 1-5 available for rows 12-47 with 8 drives installed
    - \* 1-5 available for rows 23-47 with 16 drives installed
    - \* 1-5 available for rows 35-47 with 24 drives installed
  - Base Drive Module (BDM) - panel 13 (front)
    - \* 0 available with expansion to the left of the BDM
    - \* 5 available for rows 39-51 with expansion to the right of the BDM
    - \* 1-3 available for rows 0-38 with a CAP installed
    - \* 1-4 available for rows 39-51 with a CAP installed
    - \* 1-3 not available for rows 9-16 with optional window /Operator panel installed
  - Drive Expansion Module (DEM) - panel 10 (back)
    - \* 0 is available with expansion to the left of the DEM
    - \* 1-5 available for rows 12-47 with 8 drives installed
    - \* 1-5 available for rows 23-47 with 16 drives installed
    - \* 1-5 available for rows 35-47 with 24 drives installed
    - \* No columns available with 32 drives installed.
  - Drive Expansion Module (DEM) - panel 11 (front)
    - \* 0 available with expansion to the left DEM
    - \* 1-3 available for rows 0-38 with a CAP installed
    - \* 1-5 available for rows 39-51, with a CAP installed
    - \* 1-3 not available for rows 9-16 with a window installed
    - \* 1-5 available for rows 0-51 on a standard DEM panel
  - Cartridge Expansion Module (CEM) - panels 2, 4, 6, 8 (back)
    - \* 0 available with expansion to the left
    - \* 1- 5 available on the standard CEM back panel
  - Cartridge Expansion Module (CEM) - panels 3, 5, 7, 9 (front)
    - \* 0 available with expansion to the left
    - \* 1-3 available for rows 0-38 with a CAP installed

- \* 1-4 available for rows 39-51, with a CAP installed
- \* 1-4 available on the standard CEM front panel
- Cartridge Expansion Module (CEM) - panels 14, 16, 18, 20 (back)
  - \* 0-5 available on the standard CEM back panel
- Cartridge Expansion Module (CEM) - panels 15, 17, 19, 21 (front)
  - \* 0-3 available for rows 0-38 with a CAP installed
  - \* 0-5 available for rows 39-51 with a CAP installed
  - \* 0-5 available on the standard CEM front panel
- Parking Expansion Module (PEM) - left side PEM
  - \* 3-5 available on the back and front panels with a second robot installed
- Parking Expansion Module (PEM) - right side PEM
  - \* 0-2 available on the back and front panels with a second robot installed
- LSM Model SL8500 Library
  - 0-1 for all panel types

---

---

**Note:** All SL3000 columns are numbered 0-5 (left to right).

---

---

---

## Tape Management System Interface (SLUDR\*) Routines

This appendix describes TMS (Tape Management System) interface routines called by the HSC `SLUCONDB` utility and SMC `SMCUDBX` utility. These utilities call the appropriate routine based on the tape management system being used.

Routines include the following:

- `SLUDRCA1`
- `SLUDRRMM`
- `SLUDRTL`
- `SLUDRSMC`
- `SLUDRZAR`

These routines are included in the `SEALINK` library generated during ELS installation. The data returned from these routines is used to format the SMC scratch extract file.

### SLUDRCA1

The `SLUDRCA1` routine is called to read the CA-1 Tape Management Catalog (TMC) when CA-1 is defined as the tape management system. `SLUDRCA1` utilizes CA-1 macros to map the layout of the CA-1 TMC.

The macro names which map the CA-1 TMC have changed from Release 4.9 to Release 5.0. Both macros are located at the end of the `SLUDRCA1` routine with the CA-1 Release 4.9 macro as a comment. The version of `SLUDRCA1` supplied in `SAMPLIB` utilizes CA-1 Release 5.0 macro and was assembled using this macro.

- If you are using CA-1 Release 4.9 or earlier, you must comment out the Release 5.0 macro, remove the asterisk in column 1 of the Release 4.9 macro, and reassemble this module. Make sure the appropriate CA-1 macro library is used in the assemble `SYSLIB` concatenation.
- If you are running CA-1 Release 5.0 without any local modifications, this module does not need to be reassembled and nothing further must be done to this module. If you have local modifications, you must reassemble this module.

### SLUDRRMM

The `SLUDRRMM` routine is called to read the `DFSMSrmm` report extract file when `DFSMSrmm` is defined as the tape management system. `SLUDRRMM` uses the `DFSMSrmm` `EDGRVEXT` macro to map the layout of the report extract file.

Fields read in the extract file include:

- RVTYPE (record type)
- RRVOLSER (volser number)
- RVLCDATE (last change date)
- RVNAME (unit type)
- RVSTATUS (volume status)
- RVEXPDT (volume expiration date)
- RVEXPDTO (volume original expiration date)
- RVLABEL (volume label type).

SLUDRRMM processes the extract and passes the volume record information to SLUCONDB, which builds scratch card images to be input to SLUADMIN. Volume scratch status (scratch/nonscratch) is then updated in the CDS for each volume record in the extract file.

Dates on the RMM report must be in Julian date format (EDGHSKP run with PARM 'DATEFORM(J) '). Tapes listed on the RMM report without expiration dates are skipped by SLUDRRMM.

## SLUDRTLM

The SLUDRTLM routine is called to read the CA-DYNAM/TLMS Volume Master File (VMF) when CA-DYNAM/TLMS is defined as the tape management system. SLUDRTLM utilizes a CA-DYNAM/TLMS COPY member to map the layout of the VMF.

SLUDRTLM must be re-assembled if you are running TLMS Release 5.4 or if you have local modifications. If you re-assemble this module, make sure the appropriate CA-DYNAM/TLMS source library is used in the assemble SYSLIB concatenation.

## SLUDRSMC

The SLUDRSMC routine is called to read the extract file created by the SMCUBX utility from a client tape management system. This file includes one record for every scratch volume extracted that contains the volume serial number and the time of the extract run.

During execution, correct scratch synchronization is provided by matching the time (in GMT format) of the scratch extract run with the volume's "last select time." Any volume selected after the scratch selection run is skipped in SLUCONDB processing.

## SLUDRZAR

The SLUDRZAR routine is called to invoke an application programming interface (API) to read Zara's database when Zara is defined as the tape management system.

---

---

# Glossary

**abnormal end of task (abend)**

A software or hardware problem that terminates a computer processing task.

**access method**

A technique for moving data between processor storage and input/output devices.

**ACSid**

A method used in the LIBGEN process to identify ACSs by using decimal digits, 00 to 99.

**ACS**

Automated Cartridge System, a fully-automated, cartridge storage and retrieval library subsystem consisting of one or more Library Storage Modules (LSMs) connected by pass-thru ports.

**ACS library**

A library is composed of one or more Automated Cartridge Systems (ACSs), attached cartridge drives, and cartridges residing in the ACSs.

**ACS routine**

An SMS term referring to automatic class selection routine. Not to be confused with the HSC term, ACS, referring to automatic cartridge system.

**address**

Coded representation of hardware id, or the destination or origination of data.

**allocation**

The assignment of resources to a specific task.

**AMT**

Automatic Migration Threshold, values that determine when virtual tape volume migration begins and ends. VTV migration begins when the VTSS buffer reaches the high AMT and ends when the buffer reaches or falls below the low AMT. These thresholds apply to all VTSSs.

**APF**

Authorized Program Facility.

**APPL**

VTAM APPLID definition for the HSC.

**archiving**

The storage of backup files and associated journals, usually for a given period of time.

**asynchronous transmission**

Character-oriented data transmission (as distinct from IBM's block-mode transmission).

**audit**

A VSM audit (which is not the same as an HSC audit) reconstructs VTV and MVC information.

**Automated Cartridge System (ACS)**

See [ACS](#).

**Automated Cartridge System Library Software (ACSL)**

The library control software, which runs in the UNIX®-based Library Control System.

**automatic mode**

A relationship between an LSM and all attached hosts. LSMs operating in automatic mode handle cartridges without operator intervention. This is the normal operating mode of an LSM that has been modified online. The opposite situation is "manual mode."

See also [manual mode](#).

**automatic migration**

Migrating VTVs to MVCs that is automatically initiated and controlled by VSM.

**Automatic Migration Threshold (AMT)**

See [AMT](#).

**automatic recall**

Recalling VTVs to the VTSS that is automatically initiated and controlled by VSM.

**automatic reclaim**

Reclaiming MVC space that is automatically initiated and controlled by VSM.

**bar code**

A code consisting of a series of bars of varying widths. This code appears on the external label attached to the spine of a cartridge and is equivalent to the volume serial number (volser). This code is read by the robot's machine vision system.

**BISYNC**

Binary Synchronous Communications, an early low-level protocol developed by IBM and used to transmit data on a synchronous communications link. It is a form of data transmission in which synchronization of characters is controlled by timing signals generated at the sending and receiving stations.

**block**

A collection of contiguous records recorded as a unit. Blocks are separated by interblock gaps, and each block may contain one or more records.



**buffer**

A routine or storage used to compensate for a difference in rate of data flow, or time of occurrence of events, when transferring data from one device to another.

**CAP**

Cartridge Access Port, an assembly that allows several cartridges to be inserted into or ejected from an LSM without human entry into the LSM.

**CAPid**

A CAPid uniquely defines the location of a CAP by the LSM on which it resides. A CAPid is of the form "AA:LL:CC" where AA:LL is the LSMid and CC is the two digit CAP number.

**cartridge**

The plastic housing around the tape. It is approximately 4 inches (100 mm) by 5 inches (125 mm) by 1 inch (25 mm). The tape is threaded automatically when loaded in a transport. A plastic leader block is attached to the tape for automatic threading. The spine of the cartridge contains an OCR/Bar Code label listing the VOLSER (tape volume identifier).

**Cartridge Access Port (CAP)**

See [CAP](#).

**Cartridge Drive (CD)**

See [CD](#).

**cartridge tape I/O driver**

Operating system software that issues commands (for example, read, write, and rewind) to cartridge subsystems. It is the software focal point for attaching a particular type of control unit. (An example is Oracle's StorageTek CARTLIB product.)

**CAW**

Channel Address Word, an area in storage that specifies the location in main storage at which a channel program begins.

**CD**

Cartridge Drive, a hardware device containing two or four cartridge transports and associated power and pneumatic supplies.

**CDRM**

Cross Domain Resource Manager definition (if not using existing CDRMs).

**CDRSC**

Cross Domain Resource definition.

**CDS**

Control Data Set, the data set used by the host software to control the functions of the automated library. Also called a library database.

**CE**

Channel End.

**cell**

A receptacle in the LSM in which a single cartridge is stored.

**CFT**

Customer Field Test.

**CGI**

Common Gateway Interface

**channel**

A device that connects the host and main storage with the input and output devices' control units. A full-duplex channel has two paths (that is, 2 wires, or one wire with signals at two frequencies). A half-duplex channel requires that one port receives while the other transmits.

**Channel Address Word (CAW)**

See [CAW](#).

**Channel-to-Channel (CTC)**

See [CTC](#).

**check**

Detection of an error condition.

**CI**

Converter/Interpreter (JES3).

**client**

The ultimate user of the ACS services as provided by the Library Control System.

**client link**

The communications link between the LCS and a client.

**client/server**

A model of interaction in a distributed system in which a program at one site serves a request to a program at another site and awaits a response. The requesting program is called a client; the program satisfying the request is called a server.

**client system**

The system to which the LCS provides an interface to a StorageTek Automated Cartridge System.

**Client System Component (CSC)**

See [CSC](#).

**CLINK**

Cluster link, the path between a primary VTSS and secondary VTSS in a cluster. The Clink path is used to copy replicate VTVs from the primary to the secondary.

**cluster**

Two VTSSs which are physically cabled together by Clink paths and are defined in CONFIG as a cluster. A cluster consists of a primary and a secondary VTSS. VTVs with

the replicate attribute attached will be copied from the primary to the secondary as soon as possible after dismount time.

**cluster link (CLINK)**

See [CLINK](#).

**coaxial cable**

A transmission medium used in data transmissions for networks using synchronous communications, as opposed to twisted-pair, the primary medium for asynchronous RS-232 communications.

**complex**

A system composed of other systems, specifically the ACS server system and the client system.

**connected mode**

A relationship between a host and an ACS. In this mode, the host and an ACS are capable of communicating (in the sense that at least one station to this ACS is online).

**connection number**

The unique identifier on the server for a communications path. The number is assigned by TCP/IP to identify the unique connection between the server node and a specific port on the server, and the client node and a specific port on the client. The connection number exists only as long as the connection exists.

**console**

The primary I/O device to control a session on a system.

**Control Data Set (CDS)**

See [CDS](#).

**control data set allocation map**

A CDS subfile that marks individual blocks as used or free.

**control data set data blocks**

CDS blocks that contain information about the library and its configuration or environment.

**control data set directory**

A part of the CDS that maps its subdivision into subfiles.

**control data set pointer blocks**

CDS blocks that contain pointers to map data blocks belonging to a subfile.

**control data set recovery area**

A portion of the CDS reserved for maintaining integrity for updates that affect multiple CDS blocks.

**control data set subfile**

A portion of the CDS consisting of Data Blocks and Pointer Blocks containing related information.

**Control Path Adaptor (CPA)**

See [CPA](#).

**Control Unit (CU)**

See [CU](#).

**conventional transport**

An HSC-controlled transport that is not defined to VSM as an RTD.

**coupling facility**

A special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex.(I)

**coupling facility channel**

A high bandwidth fiber optic channel that provides the high-speed connectivity required for data sharing between a coupling facility and the central processor complexes directly attached to it.(I)

**coupling services**

In a sysplex, the functions of XCF that transfer data and status between members of a group residing on one or more MVS systems in the sysplex.(I)

**cross-host recovery**

The ability for one host to perform recovery for another host that has failed..

**Cross-system Coupling Facility (XCF)**

See [XCF](#).

**CPA**

Control Path Adapter, a Bus-Tech, Inc. hardware device that allows communications between a host processor's block multiplexer channel and a local area network.

**CSC**

Client System Component, software that provides an interface between the Client Computing System's operating system and the Oracle StorageTek Library Control System (LCS).

**CSE**

Customer Service Engineer.

**CSI**

Consolidated System Inventory.

**CSL**

Cartridge Scratch Loader.

**CTC**

Channel-to-Channel, refers to the communication (transfer of data) between programs on opposite sides of a channel-to-channel adapter.(I)

**CU**

Control Unit, a microprocessor-based unit situated locally between a channel and an I/O device. It translates channel commands into device commands and sends device status to the channel.

**DAE**

Dump Analysis Elimination.

**DASD**

Direct Access Storage Device, IBM's term for a disk drive storage device.

**data**

Any representations such as characters or analog quantities to which meaning is, or might be, assigned.

**Data Buffer Utilization (DBU)**

See [DBU](#).

**data class**

A collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set.

**data compaction**

An algorithmic data-reduction technique that encodes data from the host and stores it in less space than unencoded data. The original data is recovered by an inverse process call decompression.

**data-compaction ratio**

The number of host data bytes divided by the number of encoded bytes. It is variable depending on the characteristics of the data being processed. The more random the data stream, the lower the opportunity to achieve compaction.

**Data Control Block (DCB)**

See [DCB](#).

**Data Path Adapter**

A hardware device which translates from a client computing system's data protocol to the data protocol of the StorageTek Control Unit or IMU. An example is DEC's TC44-AA/BA STI-to-4400 ACS Interconnect.

**data set**

A set of records treated as a unit.

**data sharing**

The ability of concurrent subsystems or application programs to directly access and change the same data while maintaining data integrity.(I)

**data streaming**

A continuous stream of data being transmitted in character or binary-digit form, using a specified format.

**DBU**

Disk Buffer Utilization, the ratio of used to total VTSS buffer capacity.

**DCB**

Data Control Block, a control block used by access routines in storing and retrieving data.

**DDR**

Dynamic Device Reconfiguration, a facility that allows a demountable volume to be moved, and repositioned if necessary, without abnormally terminating the job or repeating the initial program load procedure.

**demand allocation**

An MVS term meaning that a user has requested a specific unit.

**demand migration**

Migrating VTVs to MVCs that an administrator does with the MIGRATE command or utility.

**demand recall**

Recalling VTVs to the VTSS that an administrator does with the RECALL command or utility.

**demand reclaim**

Reclaiming MVC space that an administrator does with the RECLAIM command or utility.

**device number**

A four-digit hexadecimal number that uniquely identifies a device attached to a processor.

**device preferencing**

The process of preferring one 36-track transport type over another 36-track transport type.

**device separation**

See [drive exclusion](#).

**DFSMS**

Data Facility Storage Management Subsystem.

**direct access storage device (DASD)**

See [DASD](#).

**directed allocation**

See [drive prioritization](#).

**disconnected mode**

A relationship between a host and an ACS. In this mode, the host and the ACS are not capable of communicating (there are no online stations to this ACS).

**disk buffer utilization (DBU)**

See [DBU](#).

**DOME<sub>d</sub>**

Pertaining to a console message that was previously highlighted during execution, but is now at normal intensity.

**dotted-decimal notation**

The syntactic representation of a 32-bit integer that consists of four 8-bit numbers written in base ten with periods (dots) separating them. In TCP/IP descriptions, dotted-decimal notation is used for Internet addresses.

**drain**

The deletion of data from an MVC. May be accompanied by a “virtual” eject to prevent the MVC from being reused.

**drive exclusion**

(previously known as device separation) refers to the SMC function of excluding drives for an allocation request based on SMC exclusion criteria.

**drive loaded**

A condition of a tape drive in which a tape cartridge has been inserted in the drive, and the tape has been threaded to the beginning-of-tape position.

**drive panel**

An LSM wall containing tape transports. The drive panel for a T9840 transport can contain either 10 or 20 transports. The drive panel for a non-T9840 transport can contain a maximum of 4 transports.

**drive prioritization**

(previously known as directed allocation) refers to the SMC function of influencing selection of a particular drive based on allocation criteria, including volume location.

**Dual LMU**

A hardware/microcode feature that provides a redundant LMU capability.

**dump**

A printed representation of the contents of main storage at time t. This representation is used for debugging purposes.

**Dynamic Device Reconfiguration (DDR)**

See [DDR](#).

**dynamic server switching**

The capability of switching server processors when a system failure occurs on the active server.

**EC**

Error Code, a numeric code displayed by messages indicating the type of problem that caused an error.

**ECART**

Enhanced Capacity Cartridge.

**ECB**

Event Control Block, provides an area for a completion code to be stored when an operation has completed.

**EDL**

Eligible Device List, a group of tape drives that are available to satisfy an allocation request.

**eligible device list (EDL)**

See [EDL](#).

**enhanced CAP**

An enhanced CAP contains two forty-cell magazine-style CAPs and a one-cell priority CAP (PCAP). Each forty-cell CAP holds four removable magazines of ten cells each. An LSM access door with an enhanced CAP contains no cell locations for storing cartridges. An enhanced CAP is ordered as Feature Number CC80.

See also [Cartridge Access Port \(CAP\)](#) and [standard capacity cartridge](#).

**effective recording density**

The number of user bytes per unit of length of the recording medium

**eject**

The LSM robot places a cartridge in a Cartridge Access Port (CAP) so the operator can remove it from the LSM.

**enhanced capacity cartridge**

A cartridge that has a length of 1100 feet and can be used only on 36-track transports (that is, 4490, 9490, and 9490EE).

**Enterprise Systems Connection (ESCON)**

See [ESCON](#).

**EOT**

End-of-Tape marker.

**EPO**

Emergency Power Off.

**ERDS**

Error Recording Data Set.

**EREP**

Environmental Recording, Editing, Printing.

**ERP**

Error Recovery Procedure, a procedure designed to help isolate and, where possible, to recover from errors in equipment.

**error codes (EC)**

See [EC](#).



**Error Recovery Procedures (ERP)**

See [ERP](#).

**ESCON**

Enterprise Systems Connection, a set of products and services that provides a dynamically connected environment using optical cables as a transmission medium.(I)

**esoteric name**

The name assigned to transports that have the same device type.

**Ethernet**

One LAN architecture using a bus topology that allows a variety of computers to be connected to a common shielded coaxial spine. The Ethernet architecture is similar to the IEEE 802.3 standard.

**event control block (ECB)**

See [ECB](#).

**ExPR**

Expert Performance Reporter, collects performance data and generates reports about Oracle StorageTek Nearline ACSs and VTSS status and performance. It has an MVS component and a PC component.

**Expert Performance Reporter (ExPR)**

See [ExPR](#).

**file**

A set of related records treated as a unit.

**file protected**

Pertaining to a tape volume from which data can be read only. Data cannot be written on or erased from the tape.

**File Transfer Protocol (FTP)**

See [FTP](#).

**foreign socket**

One of two end-points in a TCP/IP connection-oriented protocol. Specifies the address of a foreign host that can connect to the server.

**format**

The arrangement or layout of data on a data medium.

**FTP**

File transfer protocol, a TCP/IP command that provides a way to transfer files between machines connected through TCP/IP.

**GB**

1,073,741,834 bytes of storage

**GDG**

Generation Data Group, an MVS data set naming convention. Sequence numbers are appended to the basic data set name to track the generations created for that data set.

**generalized trace facility (GTF)**

See [GTF](#).

**generation data group (GDG)**

See [GDG](#).

**GTF**

Generalized Trace Facility, an MVS facility used to trace software functions and events.

**handshake**

A flow-of-control signal sent by one process to another.

**HDA**

Head/Disk Assembly.

**helical cartridge**

A high capacity, helical scan cartridge that can hold up to 50GB of uncompressed data. This cartridge can be used only on RedWood (SD-3) transports.

**High Watermark Setup (HWS)**

See [HWS](#).

**host computer**

A computer that controls a network of computers.

**Host Software Component (HSC)**

See [HSC](#).

**HSC**

Host Software Component, software running on the Library Control System processor that controls the functions of the ACS.

**HSM**

Hierarchical Storage Manager.

**HTTP**

Hypertext Transfer Protocol.

**HWS**

High Watermark Setup, relates to chains set up for tape transport allocation in JES3.

**IEEE 802.3**

A standard produced by the IEEE and accepted worldwide for local area networks using CSMA/CD (Carrier Sense Multiple Access with Collision Detection).

**ICRC**

Improved Cartridge Recording Capacity, a compression and compaction feature that increases the amount of data that can be stored on a 1/2-inch cartridge.

**Improved Cartridge Recording Capability (ICRC)**

See [ICRC](#).

**ID**

Identifier or identification.

**IDAX**

Interpreter Dynamic Allocation exit, a subfunction of the DFSMS/MVS subsystem request (SSREQ 55) that the MVS JCL Interpreter and dynamic allocation functions issue for calling DFSMS ACS routines for management of the data set requested.

**IML**

Initial Microprogram Load, a process that activates a machine reset.

**Index**

A function performed by the cartridge loader that moves cartridges down the input or output stack one cartridge position. A loader can perform multiple consecutive indexes.

**Initial Microprogram Load (IML)**

See [IML](#).

**Initial Program Load (IPL)**

A process that activates a machine reset.

**Interactive Storage Management Facility**

A series of applications for defining DFSMS/MVS storage groups and classes.

**Internet**

A collection of networks using TCP/IP that functions as a virtual network.

**Internet address**

The numbering system used to specify a network or host on that network for TCP/IP communications. Standard Internet address notation is dotted-decimal format.

**Internet Protocol (IP)**

See [IP](#).

**Interpreter dynamic allocation exit (IDAX)**

See [IDAX](#).

**IP**

Internet Protocol, formal description of messages and rules two networks use to exchange messages.

**ISMF**

Interactive Storage Management Facility

**IVP Installation Verification Programs**

A package of programs that is run by a user after the library is installed in order to verify that the library is functioning properly

**JCL**

Job Control Language, a problem oriented language designed to describe a job's processing requirements to an operating system.

**JES**

Job Entry Subsystem.(I)

**JES2**

An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

See also [JES3](#).

**JES3**

An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them through a common job queue.

See also [JES2](#).

**Job Control Language (JCL)**

See [JCL](#).

**journal**

The log associated with journaling. The log (stored in a data set) contains a record of completed work and changes to the control data set since the last backup was created.

**journaling**

A technique for recovery that involves creating a backup control data set and maintaining a log of all changes (transactions) to that data set.

**LAN**

Local Area Network, a network in a small (local) geographic area

**LCS**

Library Control System, the library control platform and the Library Control Software.

**LCS processor console**

The Library Control System processor console is used to control the VM operating system (for the VM-based LCS).

**LCU**

Library Control Unit, the portion of an LSM that controls the movements of the robot.

**LIBGEN**

The process of defining the configuration of a library to the VM/HSC.

**library**

See [TapePlex](#).

**library cartridge transport**

See [transport](#).

**library complex**

A library complex consists of one HSC Control Data Set (CDS) and may contain up to 256 Automatic Cartridge Systems (ACSs), each of which may contain up to 24 Library Storage Modules (LSMs).

**library control component**

Software that controls the mounting and dismounting of cartridges in an ACS.

**library control platform**

The hardware and software that provides the proper environment for the Library Control System.

**library control processor**

Properly configured computer hardware that supports the operation of the Library Control System.

**Library Control Software**

A library control component, the client system interface, and library utilities.

**Library Control System (LCS)**

See [LCS](#).

**Library Control Unit (LCU)**

See [LCU](#).

**library database**

A file or data set containing information about the location and status of the removable media volumes, such as cell location, scratch status. Also called a control data set (CDS).

**library drive**

A cartridge drive in the ACS, as distinct from a standalone cartridge drive.

**Library Management Unit (LMU)**

See [LMU](#).

**library mode**

The operation of a 4480 Cartridge Subsystem as part of a 4400 Automated Cartridge System, as opposed to manual mode, in which the operator inserts cartridges into the transports.

See also [manual mode](#).

**LibraryStation**

Software that allows MVS hosts to share ACS facilities with client systems.

**Library Storage Module (LSM)**

See [LSM](#).

**LMU**

Library Management Unit, a hardware and software product that coordinates the activities of one or more LSMs/LCUs.

**Local Area Network (LAN)**

See [LAN](#).

**local port**

The designation of a given application or process among many that are available for a TCP/IP-capable host processor.

**local socket**

The address combination of a TCP/IP-capable host's network address and a specific port for an application process.

**logical port (LP)**

See [LP](#).

**LP**

Logical Port, CLS software that interfaces with the client system. The CLSLP is one of the software components used to pass data between the client system and the VM/HSC.

**LSM**

Library Storage Module, the standard LSM (4410) a twelve-sided structure with storage space for up to around 6000 cartridges. It also contains a free-standing, vision-assisted robot that moves the cartridges between their storage cells and attached transports. See also [PowderHorn](#), [SL3000](#), [SL8500](#), and [WolfCreek](#).

See also [PowderHorn \(9310\)](#), [SL3000](#), [SL8500](#), and [WolfCreek \(9360\)](#).

**LSMid**

An LSMid is composed of the ACSid joined to (concatenated with) the LSM number.

**LSM number**

A method used to identify an LSM. An LSM number is the result of defining the SLIACS macro LSM parameter during a LIBGEN. The first LSM listed in this parameter acquires the LSM number of 00 (decimal) the second LSM listed acquires a number of 01, and so forth, until all LSMs are identified (up to a maximum of 99).

**management class**

A collection of management attributes, assigned by the storage administrator, that are used to control the allocation and use of space by a data set. Note that SMS Management Classes are different from VSM Management Classes.

**manual mode**

Operation of a cartridge drive apart from an ACS.

See also [library mode](#).

**master LMU**

The LMU currently controlling the functional work of the ACS in a dual LMU configuration.

**media capacity**

The amount of data that can be contained on storage media and expressed in bytes of data.

**migration**

The movement of VTVs from the VTSS to the RTD where the VTVs are stacked onto MVCs. See automatic migration and demand migration.

See also [automatic migration](#) and [demand migration](#).

**mixed configuration**

A configuration that contains different types of cartridge drives in both manual and library modes.

**modem**

A device that enables digital data to be transmitted over an analog transmission facility.

**multi-client**

The environment where more than one (homogenous or heterogeneous) client system is connected to one LCS.

**Multi-Volume Cartridge (MVC)**

See [MVC](#).

**MVC**

multi-volume cartridge, a physical tape cartridge residing in an LSM that either contains migrated virtual tape volumes (VTVs) or is identified as a volume that can be selected for VTV stacking.

**MVS system console**

The MVS/CSC provides an operator interface through the MVS system console.

**OCR label**

Optical Character Recognition Label, an external label attached to the spine of a cartridge that is both human and machine readable.

**operator console**

In this document, the operator console refers to the MVS client system console.

**Operating System (OS)**

See [OS](#).

**OS**

Operating System, software that controls the execution of programs that facilitate overall system operation.

**PARMLIB control statements**

Parameter library (PARMLIB) control statements allow you statically specify various operation parameters which take effect at HSC initialization. Identifying your system requirements and then specifying the appropriate control statements permits you to customize the HSC to your data center.

**Pass-thru Port (PTP)**

See [PTP](#).

**PCR**

Product Change Request, a request for enhancement to a product. Normally, this request comes from a client, but may come from Oracle.

**physical port**

The communications hardware required to support a server/client link.

**physical volume**

A physically bound unit of data file media.

See [cartridge](#).

**PowderHorn (9310)**

The high-performance version of the standard LSM.

**preconfigured package**

A storage server package including all hardware, software, and configuration parameter settings delivered by the vendor.

**primary**

One of two VTSSs in a cluster which is designated in CONFIG as the primary. During normal operations the primary services the host workload and copies replicate VTVs to the secondary.

**Product Change Request (PCR)**

See [PCR](#).

**Program Temporary Fix (PTF)**

See [PTF](#).

**Program Update Tape (PUT)**

See [PUT](#).

**protocol**

A formal description of message formats and the rules two or more machines must follow to exchange these messages.

**PTF**

Program Temporary Fix, a software release designed to remedy one or a series of defects.

**PTP**

Pass-Thru Port, a mechanism that allows a cartridge to be passed from one LSM to another in a multiple LSM ACS.

**PUT**

Program Update Tape, one or more tapes containing updates to, or new versions of, the MVS/CSC system software.

**RACF**

Resource Access Control Facility, security software controlling access to data sets.



**Real Tape Drive (RTD)**

See [RTD](#).

**recall**

The movement of VTVs from the MVC back to the VTSS. May be automatic or on demand.

**reclaim**

Refers to MVC space reclamation. For automatic and demand reclamation, VTCS uses the amount of fragmented free space on the MVC and the amount of VTV data that would have to be moved to determine if space reclamation is justified.

**reconciliation**

An automatic process initiated when a cluster is reestablished after the primary or secondary has been offline. Reconciliation ensures that the contents of the primary and secondary are identical with respect to replicate VTVs.

**recovery**

Automatic or manual procedures to resolve problems in the server system.

**reel-id**

Identifier of a specific tape volume. Equivalent to volume serial number (VOLSER).

**replication**

Copying a replicate VTV from the primary VTSS to the secondary VTSS in a cluster. When replication completes, there are two copies of the VTV, one in the primary and one in the secondary.

**replicate VTV**

A VTV which has had the replicate attribute attached to it by a management class statement.

**request**

Term used to refer to commands issued to the 4400 ACS to perform a tape-related function.

**Resource Access Control Facility (RACF)**

See [RACF](#).

**RTD**

Real Tape Drive, the physical transport attached to the LSM. The transport has a data path to a VTSS and may optionally have a data path to MVS or to another VTSS.

**scratch tape**

A tape that is available to any user because it is not owned.

**scratch tape subpool**

A defined subset of all scratch tapes. Subpools are composed of one or more ranges of volsers with similar physical characteristics (type of volume—reel or cartridge, reel size, length, physical location, and so on). Some installations may also subdivide their scratch pools by other characteristics such as label type.

**SD-3**

Oracle's StorageTek helical cartridge transport. Also known as RedWood.

**SL3000**

Oracle's StorageTek SL3000 modular library offers mixed media, logical and physical partitioning capabilities, advanced management, and high availability. It supports mixed environments, including mainframe and open systems, and is scalable from 200 to just under 6,000 cartridge slots.

**SL8500**

Oracle's StorageTek SL8500 modular library offers mixed media, logical and physical partitioning capabilities, advanced management, and high capacity and availability. It supports mixed environments, including mainframe and open systems, and is scalable from the standard 1,450 cartridge slots to 100,880 cartridge slots in a complex configuration.

**secondary**

One of two VTSSs in a cluster which is designated in CONFIG as the secondary. During normal operations the secondary receives copies of replicate VTVs, stores them, and makes a migration copy on an MVC as soon as possible.

**secondary recording**

A technique for recovery involving maintaining both a control data set and a copy (secondary) of the control data set.

**server**

An ELS library control system such as HSC. In SMC a server is represented by a named SERVER path to a named TAPEPLEX. While the SMC HTTP server component is required as the middle ware on the remote host, the server, as far as SMC is concerned is the ELS library control system operating on the remote host.

**Small Programming Enhancement (SPE)**

See [SPE](#).

**SMC**

Storage Management Component, software interface between IBM's z/OS operating system and StorageTek automated library control systems, HSC and MVS/CSC. SMC performs the allocation processing, message handling, and SMS processing for the ELS solution.

**SMP**

System Modification Program.

**SMP/E**

System Modification Program Extended.

**SMS**

System Managed Storage.

**SNA**

Systems Network Architecture, a description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

**socket**

A unique address on a network plus a node address plus the id of one specific application on a specific network. An abstraction used by TCP/IP.

**SPE**

Small Programming Enhancement, a supplement to a released program that can affect several products or components.

**standard capacity cartridge**

A cartridge that can be used on any longitudinal transport (that is, 4480, 4490, 9490, or 9490EE).

**standby**

The status of a station that has been varied online but is connected to the standby LMU of a dual LMU ACS.

**standby LMU**

The redundant LMU in a dual LMU configuration that is ready to take over in a Master LMU failure or when the operator issues a `SWITCH` command.

**station**

A hardware path between the host computer and an LMU over which the VM/HSC and LMU send control information.

**storage class**

A named list of storage attributes that identify performance goals and availability requirements for a data set. Note that SMS Storage Classes are different from VSM Storage Classes.

**storage group**

A collection of storage volumes and attributes defined by the storage administrator. Note that this is an SMS concept, not a VSM concept.

**storage server**

A set of hardware and software products designed to enable heterogeneous computer systems to use automated tape cartridge library services.

**Storage Management Component (SMC)**

See [SMC](#).

**switchover**

The assumption of master LMU function by the standby LMU.

**synchronous**

See [BISYNC](#).

**synchronous LAN**

Local area network built on synchronous communications.

**sysplex**

A set of MVS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads.(I)

**Systems Network Architecture (SNA)**

See [SNA](#).

**tape drive**

A tape processing device consisting of up to four transports in a cabinet. A drive can refer to an individual transport.

**Tape Library Management System (TLMS)**

See [TLMS](#).

**TapePlex**

(formerly “library”), a single StorageTek hardware configuration, normally represented by a single HSC Control Data Set (CDS). A TapePlex may contain multiple Automated Cartridge Systems (ACSs) and Virtual Tape Storage Subsystems (VTSSs).

**TCP**

Transmission Control Protocol, an inter-network standard protocol that provides a full-duplex stream service.

**TCP/IP**

Transmission Control Protocol/Internet Protocol.

**TLMS**

Tape Library Management System, refers to any tape library management system, not to CA-1.

**trace event type**

Types of event traced through the system when tracing is enabled.

**trace file**

A file that contains information useful for debugging the system.

**transaction**

A specific set of input that triggers the execution of a specific process.

**Transmission Control Protocol (TCP)**

See [TCP](#).

**transport**

An electro-mechanical device used to thread, position, and read or write from a tape.

**UCB**

Unit Control Block.

**userid**

Sometimes referred to as the VM userid, the userid is the name that identifies a specific “virtual machine” user or client.

**utility**

Program that performs a function ancillary to the chief function(s) of a computer system.

**Virtual Machine (VM)**

See [VM](#).

**virtual storage**

A feature of the OS where main storage requirements are allocated by segments (or pages) as needed by programs, thus creating the apparent existence of unlimited or virtual storage.

**Virtual Storage Manager (VSM)**

See [VSM](#).

**Virtual Tape Control System (VTCS)**

See [VTCS](#).

**Virtual tape drive (VTD)**

See [VTD](#).

**Virtual Tape Storage Subsystem (VTSS)**

See [VTSS](#).

**Virtual tape volume (VTV)**

See [VTV](#).

**Virtual Telecommunications Access Method (VTAM)**

See [VTAM](#).

**virtual thumbwheel**

An HSC feature that allows read-only access to a volume that is not physically write-protected.

**VM**

Virtual Machine, a functional simulation of a computer and its associated devices. Each virtual machine is controlled by a suitable operating system.

**volume**

A tape cartridge (data carrier) that is mounted or dismounted as a unit.

**VOLSER**

Volume Serial Number, an identifier of a physical volume.

**volume serial number (VOLSER)**

See [VOLSER](#).

**VSM**

Virtual Storage Manager, a storage solution that virtualizes volumes and transports in a VTSS buffer to improve media and transport use.

**VTAM**

Virtual Telecommunications Access Method, IBM host-resident communications software that serves as a common interface for communications.

**VTCS**

Virtual Tape Control System, the primary host code for the Virtual Storage Manager (VSM) solution. This code operates in a separate address space, but communicates closely with HSC.

**VTD**

Virtual Tape Drive, an emulation of a physical transport in the VTSS that looks like a physical tape transport to MVS. The data written to a VTD is really being written to DASD. The VTSS has 64 VTDs that do virtual mounts of VTVs.

**VTSS**

Virtual Tape Storage Subsystem, the DASD buffer containing virtual volumes (VTVs) and virtual drives (VTDs). The VTSS is a StorageTek RAID 6 hardware device with microcode that enables transport emulation. The RAID device can read and write "tape" data from/to disk, and can read and write the data from/to a real tape drive (RTD).

**VTV**

Virtual Tape Volume, a portion of the DASD buffer that appears to the operating system as a real tape volume. Data is written to and read from the VTV, and the VTV can be migrated to and recalled from real tape.

**WolfCreek (9360)**

The high-performance LSM with a smaller capacity than the standard LSM.

**Write Tape Mark (WTM)**

See [WTM](#).

**WTM**

Write Tape Mark, the operation performed to record a special magnetic mark on tape. The mark identifies a specific location on the tape.

**WTO**

Write-to-Operator.

**WTOR**

Write-to-Operator with reply.

**XCF**

Cross-system Coupling Facility, a component of MVS that provides functions to support cooperation between authorized programs running within a sysplex.(I)

## A

---

ACSid identification, B-1  
ACTIvities command  
  description, 3-1  
  JCL requirements, 3-3  
  parameters, 3-2  
  syntax, 3-2  
ACTMVCgn command  
  description, 3-4  
  JCL requirements, 3-5  
  parameters, 3-5  
  syntax, 3-4  
ALLOCDef command  
  description, 2-1  
  parameters, 2-2  
  syntax, 2-1  
ALLOcJob command  
  description, 2-7  
  parameters, 2-8  
  syntax, 2-7  
ARCHive command  
  description, 3-6  
  parameters, 3-6  
  syntax, 3-6  
AUDit command  
  description, 3-10  
  parameters, 3-11  
  syntax, 3-10

## B

---

BACKup command  
  description, 3-19  
  JCL requirements, 3-19  
  parameters, 3-19  
  syntax, 3-19

## C

---

CANcel command  
  description, 3-20  
  parameters, 3-20  
  syntax, 3-20  
CAPid identification, B-1  
CAPPref command

  description, 3-21  
  parameters, 3-21  
  syntax, 3-21  
CDs command  
  description, 3-23  
  parameters, 3-24  
  syntax, 3-24  
CDSCREat command  
  description, 3-25  
  JCL requirements, 3-27  
  parameters, 3-26  
  syntax, 3-25  
CDSData command  
  description, 3-28  
  parameters, 3-28  
  syntax, 3-28  
CDSDEF command  
  description, 3-29  
  parameters, 3-30  
  syntax, 3-29  
CLEan command  
  description, 3-31  
  parameters, 3-32  
  syntax, 3-32  
CMDDef command  
  description, 2-10  
  parameters, 2-10  
  syntax, 2-10  
COLlector command  
  description, 2-11  
  parameters, 2-12  
  syntax, 2-11  
Column values, B-4  
command interfaces  
  HSC, 1-5  
  SMC, 1-1  
  VTCS, 1-5  
COMMPath command  
  description, 3-32  
  parameters, 3-34  
  syntax, 3-33  
COMMtest command  
  description, 2-13  
  parameters, 2-14  
  syntax, 2-13  
CONFIg CLINK statement

- description, 3-39
  - parameters, 3-40
  - syntax, 3-40
  - CONFIg CLUSTER statement
    - description, 3-41
    - parameters, 3-41
    - syntax, 3-41
  - CONFIg command
    - description, 3-37
    - parameters, 3-37
    - syntax, 3-37
  - CONFIg GLOBAL statement
    - description, 3-42
    - parameters, 3-43
    - syntax, 3-42
  - CONFIg HOST statement
    - description, 3-50
    - parameters, 3-50
    - syntax, 3-50
  - CONFIg MCVOL statement
    - description, 3-51
  - CONFIg RECLAIM command
    - parameters, 3-51
  - CONFIg RECLAIM statement
    - description, 3-51
    - syntax, 3-51
  - CONFIg RTDpath statement
    - description, 3-53
    - parameters, 3-54
    - syntax, 3-53
  - CONFIg STORMNGR statement
    - description, 3-55
    - parameters, 3-56
    - syntax, 3-55
  - CONFIg TAPEPLEX statement
    - description, 3-56
    - parameters, 3-57
    - syntax, 3-56
  - CONFIg VTD statement
    - description, 3-57
    - parameters, 3-58
    - syntax, 3-58
  - CONFIg VTSS statement
    - description, 3-59
    - parameters, 3-60
    - syntax, 3-59
  - CONFIg VTVVOL statement
    - description, 3-62
  - CONSolid command
    - description, 3-62
    - parameters, 3-63
    - syntax, 3-63
- D**
- 
- DBSERVer command
    - description, 3-65
    - parameters, 3-65
    - syntax, 3-65
  - DEComp command
    - description, 3-66
    - parameters, 3-67
    - syntax, 3-66
  - DELETSCR command
    - description, 3-68
    - parameters, 3-69
    - syntax, 3-68
  - DIRBLD command
    - description, 3-71
    - parameters, 3-71
    - syntax, 3-71
  - DISMount command
    - description, 2-15, 3-72
    - parameters, 2-16, 3-72
    - syntax, 2-15, 3-72
  - Display Acs command
    - description, 3-75
    - parameters, 3-75
    - syntax, 3-75
  - Display ACTive command
    - description, 3-76
    - parameters, 3-76
    - syntax, 3-76
  - Display ALI command
    - description, 3-83
    - syntax, 3-83
  - Display Cap command
    - description, 3-84
    - parameters, 3-84
    - syntax, 3-84
  - Display CDS command
    - description, 3-86
    - parameters, 3-86
    - syntax, 3-86
  - Display CLink command
    - description, 3-87
    - parameters, 3-88
    - syntax, 3-88
  - Display CLUster command
    - description, 3-90
    - parameters, 3-90
    - syntax, 3-90
  - Display CMD command
    - description, 3-93
    - parameters, 3-93
    - syntax, 3-93
  - Display command
    - description, 3-73
    - options, 3-73
  - Display COMMPath command
    - description, 3-93
    - parameters, 3-94
    - syntax, 3-94
  - Display CONFIG command
    - description, 3-94
    - parameters, 3-95
    - syntax, 3-94
  - Display DRive command
    - description, 2-16
    - parameters, 2-17



- syntax, 2-16
- Display DRIVE\_INFO command
  - description, 3-100
  - parameters, 3-101
  - syntax, 3-100
- Display DRives command
  - description, 3-98
  - parameters, 3-99
  - syntax, 3-98
- Display EXceptns command
  - description, 3-102
  - parameters, 3-103
  - syntax, 3-103
- Display LINKSto command
  - description, 3-104
  - parameters, 3-105
  - syntax, 3-104
- Display LMUPDEF command
  - description, 3-107
  - parameters, 3-107
  - syntax, 3-107
- Display LOCKs command
  - description, 3-107
  - parameters, 3-108
  - syntax, 3-107
- Display Lsm command
  - description, 3-109
  - parameters, 3-109
  - syntax, 3-109
- Display Message command
  - description, 3-111
  - parameters, 3-111
  - syntax, 3-111
- Display MGMTDEF command
  - description, 3-111
  - parameters, 3-112
  - syntax, 3-112
- Display MIGrate command
  - description, 3-112
  - parameters, 3-112
  - syntax, 3-112
- Display MNTD command
  - description, 3-115
  - parameters, 3-116
  - syntax, 3-115
- Display MONitor command
  - description, 3-116
  - parameters, 3-116
  - syntax, 3-116
- Display MVC command
  - description, 3-117
  - parameters, 3-117
  - syntax, 3-117
- Display MVCPool command
  - description, 3-122
  - parameters, 3-123
  - syntax, 3-123
- Display OPTion command
  - description, 3-127
  - parameters, 3-127
- syntax, 3-127
- Display PATH command
  - description, 3-127
  - parameters, 3-128
  - syntax, 3-127
- Display POLicy command
  - description, 2-18
  - parameters, 2-19
  - syntax, 2-18
- Display Queue command
  - description, 3-130
  - parameters, 3-130
  - syntax, 3-130
- Display RC command
  - description, 2-19
  - parameters, 2-19
  - syntax, 2-19
- Display REPlicat command
  - description, 3-130
  - parameters, 3-131
  - syntax, 3-130
- Display Requests command
  - description, 3-132
  - parameters, 3-133
  - syntax, 3-133
- Display RTD command
  - description, 3-133
  - parameters, 3-133
  - syntax, 3-133
- Display SCRAtch command
  - description, 3-137
  - parameters, 3-137
  - syntax, 3-137
- Display SEN command
  - description, 3-141
  - parameters, 3-141
  - syntax, 3-141
- Display SERVER command
  - description, 3-142
  - parameters, 3-142
  - syntax, 3-142
- Display SERVer command
  - description, 2-20
  - parameters, 2-20
  - syntax, 2-20
- Display SMC command
  - description, 2-21
  - parameters, 2-21
  - syntax, 2-21
- Display SRVlev command
  - description, 3-143
  - parameters, 3-143
  - syntax, 3-143
- Display Status command
  - description, 3-144
  - parameters, 3-144
  - syntax, 3-144
- Display STORCLas command
  - description, 3-145
  - parameters, 3-145

- syntax, 3-145
- Display STORMNGR command
  - description, 2-21
  - parameters, 2-22
  - syntax, 2-22
- Display STORMNgr command
  - description, 3-146
  - parameters, 3-146
  - syntax, 3-146
- Display TAPEPlex command
  - description, 2-22
  - parameters, 2-23
  - syntax, 2-22
- Display TASKs command
  - description, 3-148
  - parameters, 3-148
  - syntax, 3-148
- Display THReshld command
  - description, 3-149
  - parameters, 3-150
  - syntax, 3-149
- Display Volser command
  - description, 3-152
  - parameters, 3-152
  - syntax, 3-152
- Display Volume command
  - description, 2-23
  - parameters, 2-24
  - syntax, 2-23
- Display VOLume\_Info command
  - description, 3-154
  - parameters, 3-154
  - syntax, 3-154
- Display VSCRatch command
  - description, 3-155
  - parameters, 3-155
  - syntax, 3-155
- Display VTD command
  - description, 3-156
  - parameters, 3-157
  - syntax, 3-156
- Display VTSS command
  - description, 3-159
  - parameters, 3-159
  - syntax, 3-159
- Display VTV command
  - description, 3-163
  - parameters, 3-164
  - syntax, 3-163
- DRAin command
  - description, 3-167
  - parameters, 3-168
  - syntax, 3-167
- DRCHKPT command
  - description, 3-168
  - parameters, 3-169
  - syntax, 3-168
- DRIVemap command
  - description, 2-24
  - parameters, 2-25

- syntax, 2-24
- DRMONitr command
  - description, 3-169, 3-170
  - parameters, 3-169, 3-171
  - syntax, 3-169, 3-170
- DRTEST CREATE command
  - description, 3-173
  - JCL requirements, 3-175
  - parameters, 3-174
  - syntax, 3-173
- DRTEST PRIMEprd command
  - description, 3-176
  - parameters, 3-177
  - syntax, 3-177
- DRTEST RESET command
  - description, 3-178
  - parameters, 3-179
  - syntax, 3-179
- DRTEST START command
  - description, 3-179
  - parameters, 3-180
  - syntax, 3-179
- DRTEST STOP command
  - description, 3-180
  - parameters, 3-180
  - syntax, 3-180

## E

---

- EEXPORT command
  - description, 3-181
  - parameters, 3-181
  - syntax, 3-181
- Eject command
  - description, 3-183
  - parameters, 3-184
  - syntax, 3-183
- ENter command
  - description, 3-187
  - parameters, 3-188
  - syntax, 3-187
- EXECParm command
  - description, 3-189
  - parameters, 3-189
  - syntax, 3-189
- EXPORT command
  - description, 3-190
  - JCL requirements, 3-192
  - parameters, 3-191
  - syntax, 3-190

## F

---

- FMTLOG command
  - description, 3-193
  - JCL requirements, 3-193
  - parameters, 3-193
  - syntax, 3-193

## H

---

### Help command

- description, 2-26
- parameters, 2-26
- syntax, 2-26

### HSC

- command interfaces, 1-5
- service levels, 1-8

### HTTP command

- description, 2-27
- parameters, 2-28
- syntax, 2-27

## I

---

### IDAX command

- description, 2-29
- parameters, 2-30
- syntax, 2-30

### identification

- ACSid, B-1
- CAPid, B-1
- LSMid, B-1

### IMPORT command

- description, 3-193
- JCL requirements, 3-196
- parameters, 3-194
- syntax, 3-194

### INITialize command

- description, 3-196
- JCL requirements, 3-198
- parameters, 3-197
- syntax, 3-196

### INVENTORY command

- description, 3-199
- parameters, 3-199
- return codes, 3-200
- syntax, 3-199

## L

---

### LIBGen command

- description, 3-201
- JCL requirements, 3-202
- parameters, 3-202
- syntax, 3-201

### LIMIT command

- description, 2-31
- parameters, 2-33
- syntax, 2-32

### List command

- description, 2-35
- parameters, 2-36
- syntax, 2-35

### LMUPATH control statement

- description, 3-205
- parameters, 3-205
- syntax, 3-205

### LMUPDEF command

- description, 3-203

- parameters, 3-204

- syntax, 3-204

### LOG command

- description, 2-37
- parameters, 2-37
- syntax, 2-37

### LOGUTIL command

- description, 3-207
- parameters, 3-207
- syntax, 3-207

### LOGUTIL FOR\_LOSTMVC statement

- description, 3-208
- parameters, 3-208
- syntax, 3-208

### LOGUTIL GENAUDIT statement

- description, 3-209
- parameters, 3-209
- syntax, 3-209

### LOGUTIL LOCATE\_VTV statement

- description, 3-210
- parameters, 3-210
- syntax, 3-210

### LOGUTIL UNDELETE statement

- description, 3-211
- parameters, 3-211
- syntax, 3-211

### LSMid identification, B-1

## M

---

### Media Type (MEDIA) values, A-1

### MEDVERfy command

- description, 3-212
- parameters, 3-212
- syntax, 3-212

### MERGEcds command

- description, 3-214
- JCL requirements, 3-216
- parameters, 3-215
- syntax, 3-215

### MERGMFST command

- description, 3-219
- JCL requirements, 3-219
- parameters, 3-219
- syntax, 3-219

### messages

- Audit report, 3-16
- CONSolid report, 3-63

### METAdata command

- description, 2-39, 3-220
- parameters, 3-220
- syntax, 2-40, 3-220

### MGMTclas control statement

- description, 3-222
- parameters, 3-224
- syntax, 3-223

### MGMTDEF command

- description, 3-221
- parameters, 3-222
- syntax, 3-222

MIGrate command  
description, 3-250

MIGrate command (Format 1)  
parameters, 3-250  
syntax, 3-250

MIGrate command (Format 2)  
parameters, 3-252  
syntax, 3-251

MIGRSEL control statement  
description, 3-233  
parameters, 3-233  
syntax, 3-233

MIGRVTV control statement  
description, 3-235  
parameters, 3-235  
syntax, 3-235

MNTD command  
description, 3-252  
parameters, 3-253  
syntax, 3-252

MODeL Type (MODeL) values, A-8

MODify command  
description, 3-257  
parameters, 3-258  
syntax, 3-258

Modify command, MVS, 1-2, 1-5

MONitor command  
description, 2-41  
parameters, 2-42  
syntax, 2-41

MOUNT command  
description, 3-261  
parameters, 3-262  
syntax, 3-261

MOunt command  
description, 2-44  
parameters, 2-44  
syntax, 2-44

MOUNTDef command  
description, 2-45  
parameters, 2-46  
syntax, 2-45

MOVE command  
description, 3-264  
parameters, 3-265  
syntax, 3-264

MSGDef command  
description, 2-50  
parameters, 2-50  
syntax, 2-50

MSGJob command  
description, 2-52  
parameters, 2-52  
syntax, 2-52

MVCATTR control statement  
description, 3-237, 3-238  
parameters, 3-237  
syntax, 3-237

MVCDRain command  
description, 3-266

parameters, 3-267  
syntax, 3-266

MVCMAINT command  
description, 3-268  
JCL requirements, 3-271  
parameters, 3-269  
return codes, 3-272  
syntax, 3-269

MVCPLRPT command  
description, 3-273  
parameters, 3-273  
syntax, 3-273

MVCRPt command  
description, 3-278  
parameters, 3-279  
syntax, 3-278

## O

---

OFFload command  
description, 3-286  
JCL requirements, 3-287  
parameters, 3-287  
syntax, 3-286

OPTion command  
description, 3-288  
parameters, 3-289  
syntax, 3-289

OPTION TITLE control statement  
description, 3-287  
parameters, 3-288  
syntax, 3-288

options  
Display command, 3-73

## P

---

Panel values, B-2

PARMLIB, issuing commands, 1-7

PITCOPY command  
description, 3-292  
JCL requirements, 3-292  
parameters, 3-292  
syntax, 3-292

POLicy and TAPEREQ Batch Test Utility (SMCUTRQ), 4-17

POLicy command  
description, 2-54  
parameters, 2-57  
syntax, 2-55

POOLPARM Change control statement  
description, 3-351  
parameters, 3-352  
syntax, 3-351

POOLPARM control statement  
description, 3-343  
parameters, 3-344  
syntax, 3-344

procedure  
SET CLNPRFX, 3-316

SET SLIDRIVS, 3-334  
SET TCHNIQE, 3-338

## R

---

READ command  
description, 2-62  
parameters, 2-63  
syntax, 2-62

RECall command  
description, 3-293  
parameters, 3-294  
syntax, 3-293

RECLaim command  
description, 3-295  
parameters, 3-296  
syntax, 3-295

RECONcil command  
description, 3-299  
parameters, 3-300  
syntax, 3-300

Recording Technique (RECtech) values, A-5

RECOVer command  
description, 3-305  
parameters, 3-305  
syntax, 3-305

RELease command  
description, 3-305  
parameters, 3-306  
syntax, 3-306

remote UUI support, 4-8

REPLaceall command  
description, 3-306  
parameters, 3-307  
syntax, 3-306

reports  
ARCHive (MOVEVTV not specified), 3-7  
ARCHive (MOVEVTV specified), 3-9  
Audit, 3-15  
DELETSCR, 3-70  
INVENTORY, 3-201  
MEDVERfy, 3-214  
MVCMAINT, 3-272  
MVCPLRPT, 3-274  
MVCRPt, 3-280  
RECONcil, 3-301  
SCRPT, 3-312  
VOLRpt, 3-384  
VTVMaint, 3-389  
VTVRPt (Basic), 3-392  
VTVRPt (COPIES), 3-396

REStore command  
description, 3-307  
JCL requirements, 3-308  
parameters, 3-308  
syntax, 3-307

RESYNChronize command  
description, 2-64  
parameters, 2-65  
syntax, 2-65

return codes  
INVENTORY command, 3-200  
MVCMAINT command, 3-272  
SLUADMIN utility, 5-8  
SMC utilities, 4-1  
VTVMaint command, 3-389

Route command  
description, 2-65  
parameters, 2-66  
syntax, 2-66

Row values, B-3

RTV Utility  
description, 6-1  
JCL requirements, 6-4  
parameters, 6-2  
report messages, 6-4  
RTV Decompress listing, 6-8  
RTV LISTONLY listing, 6-7  
syntax, 6-1

## S

---

SCRATch command  
description, 3-308  
parameters, 3-309  
syntax, 3-308

Scratch Conversion Utility (SLUCONDB), 5-13

SCREdist command  
description, 3-309  
parameters, 3-310  
syntax, 3-310

SCRPT command  
description, 3-312  
parameters, 3-312  
syntax, 3-312

SDD utility command  
CONFig example, 4-10  
description, 4-8  
EXPORT example, 4-12  
IMPORT example, 4-11  
MERGMST example, 4-12  
parameters, 4-9  
syntax, 4-9

SENter command  
description, 3-314  
parameters, 3-314  
syntax, 3-314

SERVer command  
description, 2-67  
parameters, 2-68  
syntax, 2-67

service levels, 1-8

SET CLNPRFX command  
description, 3-316  
parameters, 3-316  
syntax, 3-316

SET command  
description, 3-314

SET COMPRFX command  
description, 3-317

- parameters, 3-317
- syntax, 3-317
- SET DRVHOST command
  - description, 3-319
  - parameters, 3-319
  - syntax, 3-319
- SET EJCTPAS command
  - description, 3-319
  - parameters, 3-320
  - syntax, 3-320
- SET EJCTSKP command
  - description, 3-320
  - parameters, 3-321
  - syntax, 3-320
- SET FREEZE command
  - description, 3-321
  - parameters, 3-321
  - syntax, 3-321
- SET HOSTID command
  - description, 3-323
  - parameters, 3-323
  - syntax, 3-323
- SET HSCLEVEL command
  - description, 3-324
  - parameters, 3-324
  - syntax, 3-324
- SET LOGFILE command
  - description, 3-325
  - parameters, 3-325
  - syntax, 3-325
- SET MAJNAME command
  - description, 3-326
  - parameters, 3-326
  - syntax, 3-326
- SET MIGOPT command
  - description, 3-327
  - parameters, 3-327
  - syntax, 3-327
- SET NEWHOST command
  - description, 3-328
  - parameters, 3-328
  - syntax, 3-328
- SET RMM command
  - description, 3-329
  - parameters, 3-329
  - syntax, 3-329
- SET SCRLABL command
  - description, 3-330
  - parameters, 3-330
  - syntax, 3-330
- SET SLIDRIVS command
  - description, 3-331
  - parameters, 3-332
  - syntax, 3-331
- SET SLISTATN command
  - description, 3-335
  - parameters, 3-335
  - syntax, 3-335
- SET SMF command
  - description, 3-336
- parameters, 3-336
- syntax, 3-336
- SET TAPEplex command
  - description, 3-336
  - parameters, 3-336
  - syntax, 3-336
- SET TCHNIQE command
  - description, 3-337
  - parameters, 3-337
  - syntax, 3-337
- SET VAULT command
  - description, 3-338
  - parameters, 3-339
  - syntax, 3-338
- SET VAULTVOL command
  - description, 3-340
  - parameters, 3-341
  - syntax, 3-341
- SET VOLPARM command
  - description, 3-341
  - JCL requirements, 3-343
  - parameters, 3-342
  - syntax, 3-342
- SET VOLPARM JOIN command
  - description, 3-353
  - JCL requirements, 3-354
  - parameters, 3-354
  - syntax, 3-353
- SET VOLPARM UPDATE command
  - description, 3-350
  - JCL requirements, 3-351
  - parameters, 3-351
  - syntax, 3-350
- SIMulate command
  - description, 2-71
  - example, 2-73
  - parameters, 2-71
  - syntax, 2-71
- SLICREAT Utility
  - calculating DASD space, 5-13
  - creating only stanby CDS, 5-13
  - DD statements, 5-11
  - executing, 5-12
  - JCL, 5-10
  - verifying, 5-12
- SLSMERGE control statement
  - description, 3-216
  - parameters, 3-217
  - syntax, 3-217
- SLUADMIN utility
  - definition statements, 5-3
  - description, 1-6, 5-1
  - execution parameters, 5-2
  - invoking, 5-1
  - output, 5-5
  - return codes, 5-8, 5-9
- SLUCONDB Utility, 5-13
- SLUDR\* routines, C-1
- SLUDRCA1 routine, C-1
- SLUDRRMM routine, C-1

SLUDRSMC routine, C-2  
 SLUDRTLm routine, C-2  
 SMCCMDS data set, 1-3  
 SMCPARMS data set, 1-3  
 SMCUDBX Utility, 4-21  
 SMCUGTF Utility, 4-32  
 SMCUSIM Utility, 4-27  
 SMCUSMF Utility, 4-33  
 SMCUTRQ Utility, 4-17  
 SMCUUUI Utility, 4-2  
 SMSDef command  
   description, 2-74  
   parameters, 2-75  
   syntax, 2-74  
 SRVlev command  
   description, 3-354  
   parameters, 3-354  
   syntax, 3-354  
 STOPMN command  
   description, 3-355  
   parameters, 3-355  
   syntax, 3-355  
 STORclas control statement  
   parameters, 3-238  
   syntax, 3-238  
 STORLST control statement  
   description, 3-243  
   parameters, 3-244  
   syntax, 3-243  
 STORMNGR command  
   description, 2-77  
   parameters, 2-78  
   syntax, 2-77  
 STORSEL control statement  
   description, 3-244  
   parameters, 3-245  
   syntax, 3-244  
 SWitch command  
   description, 3-356  
   parameters, 3-357  
   syntax, 3-357  
 syntax  
   ACTIvities command, 3-2  
   ACTMVCgn command, 3-4  
   ALLOCDef command, 2-1, 3-10  
   ALLOCJob command, 2-7  
   ARCHive command, 3-6  
   BACKup command, 3-19  
   CANcel command, 3-20  
   CAPPref command, 3-21  
   CDs command, 3-24  
   CDSCREAt command, 3-25  
   CDSData command, 3-28  
   CDSDEF command, 3-29  
   CLEan command, 3-32  
   CMDDef command, 2-10  
   COLlector command, 2-11  
   COMMPath command, 3-33  
   COMMtest command, 2-13  
   CONFIg CLINK statement, 3-40  
   CONFIg CLUSTER statement, 3-41  
   CONFIg command, 3-37  
   CONFIg GLOBAL statement, 3-42  
   CONFIg HOST statement, 3-50  
   CONFIg RECLAIM statement, 3-51  
   CONFIg RTDpath statement, 3-53  
   CONFIg STORMNGR statement, 3-55  
   CONFIg TAPEPLEX statement, 3-56  
   CONFIg VTD statement, 3-58  
   CONFIg VTSS statement, 3-59  
   CONSolid command, 3-63  
   DBSERVer command, 3-65  
   DEComp command, 3-66  
   DELETSCR command, 3-68  
   DIRBLD command, 3-71  
   DISMOUNT command, 2-15, 3-72  
   Display AcS command, 3-75  
   Display ACTive command, 3-76  
   Display ALL command, 3-83  
   Display Cap command, 3-84  
   Display CDS command, 3-86  
   Display CLink command, 3-88  
   Display CLUster command, 3-90  
   Display CMD command, 3-93  
   Display COMMPath command, 3-94  
   Display CONFIG command, 3-94  
   Display DRive command, 2-16  
   Display DRIVE\_INFO command, 3-100  
   Display DRives command, 3-98  
   Display EXceptns command, 3-103  
   Display LINKSto command, 3-104  
   Display LMUPDEF command, 3-107  
   Display LOCKs command, 3-107  
   Display Lsm command, 3-109  
   Display Message command, 3-111  
   Display MGMTDEF command, 3-112  
   Display MIGrate command, 3-112  
   Display MNTD command, 3-115  
   Display MONitor command, 3-116  
   Display MVC command, 3-117  
   Display MVCPool command, 3-123  
   Display OPTion command, 3-127  
   Display PATH command, 3-127  
   Display POLicy command, 2-18  
   Display Queue command, 3-130  
   Display RC command, 2-19  
   Display REPlicat command, 3-130  
   Display Requests command, 3-133  
   Display RTD command, 3-133  
   Display SCRAtch command, 3-137  
   Display SEN command, 3-141  
   Display SERVER command, 3-142  
   Display SERVer command, 2-20  
   Display SMC command, 2-21  
   Display SRVlev command, 3-143  
   Display Status command, 3-144  
   Display STORCLas command, 3-145  
   Display STORMNGR command, 2-22  
   Display STORMNgr command, 3-146  
   Display TAPEPLex command, 2-22

Display TASKs command, 3-148  
 Display THReshld command, 3-149  
 Display Volser command, 3-152  
 Display Volume command, 2-23  
 Display VOLume\_Info command, 3-154  
 Display VSCRatch command, 3-155  
 Display VTD command, 3-156  
 Display VTSS command, 3-159  
 Display VTV command, 3-163  
 DRAIn command, 3-167  
 DRCHKPT command, 3-168  
 DRIVemap command, 2-24  
 DRMONitr command, 3-169,3-170  
 DRTEST CREATE command, 3-173  
 DRTEST PRIMEprd command, 3-177  
 DRTEST RESET command, 3-179  
 DRTEST START command, 3-179  
 DRTEST STOP command, 3-180  
 EEXPORT command, 3-181  
 Eject command, 3-183  
 ENter command, 3-187  
 EXECParM command, 3-189  
 EXPORT command, 3-190  
 FMTLOG command, 3-193  
 Help command, 2-26,2-30  
 HTTP command, 2-27  
 IMPORT command, 3-194  
 INITialize command, 3-196  
 INVENTORY command, 3-199  
 LIBGen command, 3-201  
 LIMIT command, 2-32  
 LIst command, 2-35  
 LMUPATH control statement, 3-205  
 LMUPDEF command, 3-204  
 LOG command, 2-37  
 LOGUTIL command, 3-207  
 LOGUTIL FOR\_LOSTMVC statement, 3-208  
 LOGUTIL GENAUDIT statement, 3-209  
 LOGUTIL LOCATE\_VTV statement, 3-210  
 LOGUTIL UNDELETE statement, 3-211  
 MEDVERfy command, 3-212  
 MERGEcds command, 3-215  
 MERGMFST command, 3-219  
 METAdata command, 2-40,3-220  
 MGMTclas control statement, 3-223  
 MGMTDEF command, 3-222  
 MIGrate command (Format 1), 3-250  
 MIGrate command (Format 2), 3-251  
 MIGRSEL control statement, 3-233  
 MIGRVTV control statement, 3-235  
 MNTD command, 3-252  
 MODify command, 3-258  
 MONitor command, 2-41  
 MOUNT command, 3-261  
 MOUNT command, 2-44  
 MOUNTDef command, 2-45  
 MOVE command, 3-264  
 MSGDef command, 2-50  
 MSGJob command, 2-52  
 MVCATTR control statement, 3-237  
 MVCDRAIn command, 3-266  
 MVCMAINT command, 3-269  
 MVCPLRPT command, 3-273  
 MVCRPt command, 3-278  
 OFFload command, 3-286  
 OPTion command, 3-289  
 OPTION TITLE control statement, 3-288  
 PITCOPY command, 3-292  
 POLicy command, 2-55  
 POOLPARM Change control statement, 3-351  
 POOLPARM control statement, 3-344  
 READ command, 2-62  
 RECall command, 3-293  
 RECLaim command, 3-295  
 RECONcil command, 3-300  
 RECOVER command, 3-305  
 RELease command, 3-306  
 REPLaceall command, 3-306  
 RESTore command, 3-307  
 RESYNChronize command, 2-65  
 Route command, 2-66  
 SCRATCH command, 3-308  
 SCREdist command, 3-310  
 SCRPT command, 3-312  
 SENter command, 3-314  
 SERVer command, 2-67  
 SET CLNPRFX command, 3-316  
 SET COMPRFX command, 3-317  
 SET DRVHOST command, 3-319  
 SET EJCTPAS command, 3-320  
 SET EJCTSKP command, 3-320  
 SET FREEZE command, 3-321  
 SET HOSTID command, 3-323  
 SET HSCLEVel command, 3-324  
 SET LOGFILE command, 3-325  
 SET MAJNAME command, 3-326  
 SET MIGOPT command, 3-327  
 SET NEWHOST command, 3-328  
 SET RMM command, 3-329  
 SET SCRLABL command, 3-330  
 SET SLIDRIVS command, 3-331  
 SET SLISTATN command, 3-335  
 SET SMF command, 3-336  
 SET TAPEPlex command, 3-336  
 SET TCHNIQE command, 3-337  
 SET VAULT command, 3-338  
 SET VAULTVOL command, 3-341  
 SET VOLPARM command, 3-342  
 SET VOLPARM JOIN command, 3-353  
 SET VOLPARM UPDATE command, 3-350  
 SIMulate command, 2-71  
 SLSMERGE control statement, 3-217  
 SMSDef command, 2-74  
 SRVlev command, 3-354  
 STOPMN command, 3-355  
 STORclas control statement, 3-238  
 STORLST control statement, 3-243  
 STORMNGR command, 2-77  
 STORSEL control statement, 3-244  
 SWitch command, 3-357



TAPEPlex command, 2-79  
 TAPEREQ control statement, 2-91  
 TCPip command, 2-82  
 TRace command, 2-85, 3-360  
 TRACELKP command, 3-361  
 TREQDef command, 2-89  
 UEXIT command, 3-363  
 UEXit command, 2-97  
 UNITAttr command, 2-99  
 UNSCratch command, 3-365  
 UNSElect command, 3-366  
 USERMsg command, 2-102  
 Vary command, 3-367  
 Vlew command, 3-370  
 VLEMAINT command, 3-374  
 VMSG command, 2-103  
 VOLPARM Change control statement, 3-352  
 VOLPARM control statement, 3-349  
 VOLPCONV command, 3-376  
 VOLRpt command, 3-378  
 VTSSLST control statement, 3-246  
 VTSSSEL control statement, 3-247  
 VTVMAINT command, 3-387  
 VTVRPt BASIC command, 3-391  
 VTVRPt COPIES command, 3-395  
 VVAUDIT command, 3-397  
 Warn command, 3-398  
 XCLIENT command, 2-105  
 XUDB command, 2-107

## T

Tape Management Extract Utility (SMCUDBX), 4-21  
 TAPEPlex command  
   description, 2-79  
   parameters, 2-80  
   syntax, 2-79  
 TAPEREQ control statement  
   parameters, 2-92  
   syntax, 2-91  
 TCPip command  
   description, 2-82  
   parameters, 2-82  
   syntax, 2-82  
 TMS interface (SLUDR\*) routines, C-1  
 TRace command  
   description, 2-85, 3-360  
   parameters, 2-86, 3-360  
   syntax, 2-85, 3-360  
 Trace Format Utility (SMCUGTF), 4-32  
 TRACELKP command  
   description, 3-361  
   parameters, 3-362  
   syntax, 3-361  
 TREQDef command  
   description, 2-89  
   parameters, 2-89  
   syntax, 2-89

## U

UEXIT command  
   description, 3-363  
   parameters, 3-363  
   syntax, 3-363  
 UEXit command  
   description, 2-96  
   parameters, 2-97  
   syntax, 2-97  
 Unified User Interface Utility (SMCUUUI), 4-2  
 UNITAttr command  
   description, 2-98  
   parameters, 2-100  
   syntax, 2-99  
 UNSCratch command  
   description, 3-365  
   parameters, 3-365  
   syntax, 3-365  
 UNSElect command  
   description, 3-365  
   parameters, 3-366  
   syntax, 3-366  
 USERMsg command  
   description, 2-102  
   parameters, 2-102  
   syntax, 2-102  
 utilities  
   HSC  
     Scratch Conversion Utility (SLUCONDB), 5-13  
     SLUADMIN, 1-6, 5-1  
   SMC  
     POLicy and TAPEREQ Batch Test Utility (SMCUTRQ), 4-17  
     Tape Management Extract Utility (SMCUDBX), 4-21  
     Trace Format Utility (SMCUGTF), 4-32  
     Unified User Interface Utility (SMCUUUI), 4-2  
     UUI Simulation Utility (SMCUSIM), 4-27  
     VSM console SMF Offload Utility (SMCUGTF), 4-33  
   VTCS  
     RTV Utility, 6-1  
 utility return codes and messages, SMC, 4-1  
 UUI (Unified User Interface) support, 1-1  
 UUI Simulation Utility (SMCUSIM), 4-27

## V

Vary command  
   description, 3-367  
   parameters, 3-368  
   syntax, 3-367  
 View command  
   description, 3-370  
   parameters, 3-371  
   syntax, 3-370  
 VLEMAINT command  
   description, 3-374  
   parameters, 3-375  
   syntax, 3-374

VMSG command  
description, 2-102  
parameters, 2-103  
syntax, 2-103

VOLPARM Change control statement  
description, 3-352  
parameters, 3-353  
syntax, 3-352

VOLPARM control statement  
description, 3-348  
parameters, 3-349  
syntax, 3-349

VOLPCONV command  
description, 3-376  
JCL requirements, 3-377  
parameters, 3-377  
syntax, 3-376

VOLRpt command  
description, 3-378  
JCL requirements, 3-384  
parameters, 3-379  
syntax, 3-378

VSM console SMF Offload Utility (SMCUSMF), 4-33

VTSSLST control statement  
description, 3-246  
parameters, 3-246  
syntax, 3-246

VTSSSEL control statement  
description, 3-247  
parameters, 3-248  
syntax, 3-247

VTVMaint command  
description, 3-387  
parameters, 3-387  
return codes, 3-389  
syntax, 3-387

VTVRpt BASIC command  
description, 3-391  
parameters, 3-391  
syntax, 3-391

VTVRpt command  
description, 3-390

VTVRpt COPIES command  
description, 3-395  
parameters, 3-395  
syntax, 3-395

VVAUDIT command  
description, 3-397  
parameters, 3-398  
syntax, 3-397

## W

---

Warn command  
description, 3-398  
parameters, 3-399  
syntax, 3-398

## X

---

XCLIENT command  
description, 2-105  
parameters, 2-105  
syntax, 2-105

XUDB command  
description, 2-107  
parameters, 2-107  
syntax, 2-107

XUDB data set, 2-108