



StorageTek™ Storage Management Component (SMC) MVS software

Configuration and Administration Guide

Part Number: 312530902

Revision: C

Version: 6.0



Storage Management Component (SMC)

Configuration and Administration Guide

Release 6.0

312530902

Information contained in this publication is subject to change without notice. Comments concerning the contents of this publication should be directed to:

Global Learning Solutions
Storage Technology Corporation
One StorageTek Drive
Louisville, CO 80028-3526
USA

glsfs@stortek.com

Export Destination Control Statement

These commodities, technology or software were exported from the United States in accordance with the Export Administration Regulations. Diversion contrary to U.S. law is prohibited.

Restricted Rights

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) and (2) of the Commercial Computer Software - Restricted Rights at FAR 52.227-19 (June 1987), as applicable.

Limitations on Warranties and Liability

Storage Technology Corporation cannot accept any responsibility for your use of the information in this document or for your use in any associated software program. You are responsible for backing up your data. You should be careful to ensure that your use of the information complies with all applicable laws, rules, and regulations of the jurisdictions in which it is used.

Warning: No part or portion of this document may be reproduced in any manner or in any form without the written permission of Storage Technology Corporation.

Proprietary Information Statement

The information in this document, including any associated software program, may not be reproduced, disclosed or distributed in any manner without the written consent of Storage Technology Corporation.

Should this publication be found, please return it to StorageTek, One StorageTek Drive, Louisville, CO 80028-5214, USA. Postage is guaranteed.

Revision C (July, 2006)

StorageTek and the StorageTek logo are trademarks or registered trademarks of Storage Technology Corporation. Other products and names mentioned herein are for identification purposes only and may be trademarks of their respective companies.

© 2006 Storage Technology Corporation. All rights reserved.

Document Effectivity

EC Number	Date	Doc Kit Number	Type	Effectivity
128668	February, 2004	---	Revision A (First Edition)	This document applies to SMC, Release 6.0.
128668	June, 2004	---	Revision B (Second Edition)	
132713	July, 2006	---	Revision C	

Contents

What's New With This Release?	xv
About this Guide	xvii
Intended Audience	xvii
How this Guide is Organized	xvii
Conventions Used in this Guide	xviii
Product Names	xviii
Typographic	xviii
Symbols	xviii
Syntax Flow Diagrams	xix
Control Statements	xxiv
Related Publications	xxv
Technical Support	xxvi
Reader's Comments	xxvi
Chapter 1. Introduction	1
What is SMC?	1
Terminology	1
Chapter 2. SMC and StorageTek Library Management	3
Overview	3
SMC LIBRARY and SERVER Commands	3
StorageTek HTTP Server for OS/390 and z/OS	3
SMC Configuration Scenarios	5
Scenario 1: SMC and a Single Local HSC Library Configuration	5
Scenario 2: SMC and a Single Remote HSC Library Configuration	7
Scenario 3: SMC and a Single HSC Library Configuration with One Local and One Remote (alternate) Path	9
Scenario 4: SMC and a Single HSC Library Configuration with Multiple Remote Server Paths	11
Scenario 5: SMC with Two Libraries; One Local and One Remote	13
Scenario 6: SMC with Two Libraries; One Local and One Remote with Multiple Paths	15
SMC Policy And Drive Type Information Synchronization	16
SMC Library Subsystem Selection	16
MVS/CSC Library Issues	17

Chapter 3. Starting the SMC	19
Overview	19
Creating the SMC START Procedure	20
SMC EXEC Statement	21
Executing the SMC START Procedure	22
Chapter 4. SMC Allocation in a JES2 Environment	23
Overview	23
Jobs Selected for SMC Allocation Management	24
SSI55 Interpreter/Dynamic Allocation Exit (IDAX)	24
SSI24 Common Allocation	24
SSI78 Tape Allocation	24
Exceptions	25
DFSMS Esoteric Substitution	26
Drive Exclusion	26
Minimum Exclusion Level	27
Subsequent Levels	27
Intentionally Failing a Job	28
Specific Volume Request Drive Exclusion Criteria	29
Scratch Volume Request Drive Exclusion Criteria	34
Affinity Separation	40
Affinity Head-Of-Chain	40
User Policy Influence on Affinity Separation	40
Drive Prioritization	44
Deferring Mounts	45
Library Subsystem Selection	45
SMC DFSMS Processing	46
Enabling/Disabling the SMC DFSMS Interface	46
Tailoring the SMC DFSMS Interface	46
DATACLAS, STORCLAS, and MGMTCLAS Specification	46
Invoking ACS Routines	47
DFSMS Automatic Class Selection (ACS) Routine Environment for SMC	47
Esoteric Substitution During IDAX	48
Retrieving DFSMS Values During Allocation and Volume Mount	51
Interaction with Other Software	55
Automated Operations	55
Open Type J	55
SAMS: DISK (DMS)	56
CA-MIA Tape Sharing	56
CA1-RTS Real Time Stacking	56
MVS Security Packages	56

Chapter 5. SMC Allocation in a JES3 Environment	57
Overview	57
SMC Allocation - JES3 Not Managing Drives	58
SMC Allocation - JES3 Managing Drives	58
SSI55 Interpreter/Dynamic Allocation Exit (IDAX)	58
JES3 Converter/Interpreter (C/I)	59
SSI23 JES3 Dynamic Allocation	59
JES3 Main Device Scheduler (MDS)	59
SSI24 Common Allocation	59
Exceptions	60
DFSMS Esoteric Substitution	60
Drive Exclusion	61
Minimum Exclusion Level	62
Subsequent Levels	62
Intentionally Failing a Job	63
Specific Volume Request Drive Exclusion Criteria	64
Scratch Volume Request Drive Exclusion Criteria	69
Affinity Separation	75
Affinity Head-Of-Chain	75
User Policy Influence on Affinity Separation	75
Esoteric Unit Name Replacement	79
Suppressing Fetch Messages	82
Drive Prioritization	83
Deferring Mounts	84
JES3 Initialization Parameter Considerations	85
Device Preferencing Considerations	93
ZEROSCR Considerations	93
Virtual Drive Definition Considerations	93
SMC Normal Operations	94
JES3 Constraints	95
Timing Between C/I and MDS	95
JES3 High Watermark Setup and LSM Pass-Thru Processing	95
Library Subsystem Selection	97
SMC DFSMS Processing	98
Enabling/Disabling the SMC DFSMS Interface	98
Tailoring the SMC DFSMS Interface	98
DATACLAS, STORCLAS, and MGMTCLAS Specification	98
Invoking ACS Routines	99
DFSMS Automatic Class Selection (ACS) Routine Environment for SMC	99
Esoteric Substitution During IDAX	100
Retrieving DFSMS Values During Allocation and Volume Mount	103
GDG ALL Processing	108
Interaction with Other Software	109
Automated Operations	109
Open Type J	109
SAMS: DISK (DMS)	109

MVS Security Packages	109
Chapter 6. SMC Message Handling	111
Overview	111
User Directed Message Handling	111
Message Handling Policies	112
MVS Policies	112
HSC Policies	112
MVS/CSC Policies	112
Tape Management System Support	113
SMC Swap Processing	113
HSC Mount-Related Messages	114
Managing HSC Mounts from the SMC Client	115
Chapter 7. Operator Commands	117
Overview	117
Issuing SMC Commands	118
Issuing SMC Commands Using the MVS Modify Command	118
Issuing SMC Commands Using the SMC Command Prefix	118
Specifying SMC Commands in the SMCPARMS or SMCCMDS Data Set	119
SMC Command Descriptions	120
ALLOCDef Command	120
ALLOCJob Command	123
CMDDef Command	127
LIBrary Command	129
Llist Command	133
MSGDef Command	135
MSGJob Command	137
READ Command	140
RESYNChronize Command	141
SERVer Command	142
SMSDef Command	147
TCPip Command	151
TRace Command	152
UEXit Command	157
USERMsg Command	160
Chapter 8. Recovery Procedures	161
Overview	161
SMC Recovery Procedures (JES2)	161
Inactive SMC - Active Library Subsystem	161
Active SMC - Inactive Library Subsystem	162
Automating Mount Requests for Inactive Libraries	162
Lost MVS Mount Requests for Active Libraries	162
SMC Recovery Procedures (JES3)	163
Inactive SMC - Active Library Subsystem	163

Active SMC - Inactive Library Subsystem	163
Inactive JES3 on a Local Processor	164
Inactive JES3 on a Global Processor	164
Automating Mount Requests for Inactive Libraries	164
Lost JES3 Mount Requests for Active Libraries	165
Lost MVS Mount Requests for Active Libraries	165
Chapter 9. SMC Utilities	167
Overview	167
SMC Utility Return Codes and Messages	167
Tape Management Extract Utility	168
Running the SMC Scratch Extract Utility	169
Using SMCUDBX with CA-1 Release 4.9 or Below	171
Using SMCUDBX with CA-DYNAM/TLMS Release 5.3 or Below	171
Specifying Values for the PARM Parameter	171
Parameter Descriptions	172
Sample JCL	174
Sample Output	175
JES3 Configuration Report Utility	176
Sample JCL	176
Sample Reports	177
Chapter 10. SMC Messages	181
Overview	181
Message Format	181
Message Descriptions	182
Variable Definitions	182
SMC System Message Listing	183
SMC Utility Message Listing	226
Chapter 11. StorageTek HTTP Server Messages	231
Overview	231
Message Format	231
StorageTek HTTP Server Message Listing	232
Appendix A. Intercepted Messages	241
IBM Operating System Messages	241
JES3 Messages	243
Tape Management System Messages	243
CA-1 Messages	243
CONTROL-M/TAPE (formerly CONTROL-T) Messages	244
DFSMSrmm Messages	245

Appendix B. Differences Between SMC and NCS 4.1 and Below	247
Allocation Differences	247
SMS Differences	249
Message Handling Differences	249
Client/Server Differences	249
Appendix C. Message Change Summary	251
New Messages	251
SMC Messages	251
StorageTek HTTP Server Messages	252
Changed Messages	252
Deleted Messages	252
Index	253

Figures

Figure 1. NCS 6.0 Local Library Configuration	5
Figure 2. NCS 6.0 Remote Library Configuration	7
Figure 3. NCS 6.0 Local Library with Alternate Path Configuration	9
Figure 4. NCS 6.0 Remote Library Configuration with Two Server Instances	11
Figure 5. NCS 6.0 with Two Libraries - One Local and One Remote with Multiple Paths	13
Figure 6. NCS 6.0 with a Local Library and a Remote Library with Two Server Instances	15
Figure 7. Sample SMC START Procedure	20
Figure 8. Sample Storage Class/Storage Group Routine	50
Figure 9. Sample Data Class Routine	52
Figure 10. Sample Management Class Routine	54
Figure 11. Volume Locations for the Pass-thru Example	95
Figure 12. Sample Storage Class/Storage Group Routines	102
Figure 13. Sample Data Class Routine	104
Figure 14. Creating Management Class Routines	105
Figure 15. Scratch Synchronization Between SMC Client host and HSC CDS	168
Figure 16. Sample JCL for Scratch Extraction Utility	174
Figure 17. Scratch Extraction Utility Sample Output-Messages	175
Figure 18. JCL to Produce the JES3 Configuration Report	176

Tables

Table 1.	JES2 Drive Exclusion Levels (Specific Request)	29
Table 2.	JES2 Drive Exclusion Levels (Scratch Request)	34
Table 3.	JES3 Drive Exclusion Levels (Specific Request)	64
Table 4.	JES3 Drive Exclusion Levels (Scratch Request)	69
Table 5.	3490 Drive List	79
Table 6.	Sample Configuration	85
Table 7.	Variable Data Definitions	182
Table 8.	Intercepted Operating System Messages	242
Table 9.	Tape Management System Messages - CA-1	243
Table 10.	Tape Management System Messages - DFSMSrmm	245

What's New With This Release?

SMC Release 6.0 includes the following enhancements and modifications:

Enhancement/Modification	Publication(s)/ Primary Locations
Revision C:	
SMC exclusion level 1 for specific requests is updated to exclude all virtual drives for non-labeled (NL) specific volume requests.	Chapter 4
The SMC SERver command is revised as follows: <ul style="list-style-type: none"> The new INITFail parameter can be used to specify the number of failures (before communication is established) before the specified server path is disabled or placed out of service. The FAil limit default for the SMC SERver command is changed from 10 to 0. Rules for the <i>hostname</i> value (HOSt parameter) are changed. 	Chapter 7
The SMC SMSDEF command includes a new MOD parameter, used to specify whether SMS esoteric substitution (if specified) applies to DISP=MOD data sets as well as DISP=NEW data sets.	Chapter 7
Message changes and additions	Chapter 10 Appendix C
Second Edition (Revision B):	
Added GDG ALL considerations	Chapter 5
Message changes and additions	Chapter 10 Appendix C
First Edition (Revision A):	
The SMC adds support for the StorageTek™ HTTP Server for OS/390 and z/OS, allowing the SMC to communicate with remote HSC subsystems. This enhancement logically eliminates the 16 host limit for sharing a library. In an MVS-only environment, you can use the SMC and HTTP server to control your libraries, eliminating the need for MVS/CSC and LibraryStation.	SMC Config & Admin Guide Chapters 2, 3, 7, 10, Appendix C NCS Installation Guide Chapters 1, 5
SMC 6.0 provides the ability to manage and report certain mount/dismount exception conditions from the client console rather than the server console.	SMC Config & Admin Guide Chapter 7
The TCPip, LIBrary, and SERVer commands provide the ability to define the SMC remote server environment.	SMC Config & Admin Guide Chapters 6, 7

Enhancement/Modification	Publication(s)/ Primary Locations
The Tape Management Extract utility (SMCUDBX) supports scratch synchronization between the SMC client host Tape Management System and the HSC CDS. SMCUDBX shares TMS extract modules (SLUDR*) with the HSC and MVS/CSC.	<i>SMC Config & Admin Guide</i> Chapter 9
The SMC provides server switching capability, automatically switching to another defined server when the current server path becomes unavailable.	<i>SMC Config & Admin Guide</i> Chapter 7
User exits 1, 2, 4, 8, 9, 10, 11, 12, and 13 are now managed and invoked by the SMC. NCS 6.0 user exits are now documented in the <i>NCS User Exit Guide</i> .	<i>NCS User Exit Guide</i>
The UEXit operator command is used to define the module that is invoked for a specific user exit. It also allows you to enable or disable a user exit.	<i>SMC Config & Admin Guide</i> Chapter 7
The CA1rts parameter is added for the ALLOCDef operator command. This parameter specifies whether DEFER processing is performed at SSI24 time for compatibility with Computer Associates Real Tape Stacking feature of its CA-1 tape management system.	<i>SMC Config & Admin Guide</i> Chapter 7
During DFSMS processing, the SMC can invoke the pre-ACS Installation exit, IGDACSXT, prior to invoking the DFSMS ACS routines.	<i>SMC Config & Admin Guide</i> Chapters 4, 5, 7
The SMSDef operator command is used to tailor default SMC DFSMS support by bypassing certain DFSMS ACS routine calls.	<i>SMC Config & Admin Guide</i> Chapter 7
Message changes, additions and deletions. Message listings are added for the StorageTek™ HTTP Server for OS/390 and z/OS.	<i>SMC Config & Admin Guide</i> Chapters 10, 11, Appendix C

About this Guide

Intended Audience

This guide provides administration and configuration information for the Storage Management Component (SMC) software. It is intended for system programmers and operators responsible for configuring and maintaining the SMC software at their site.

How this Guide is Organized

This guide contains the following chapters:

- Chapter 1, “Introduction” describes the SMC product; its general features and functions.
- Chapter 2, “SMC and StorageTek Library Management” describes typical SMC configuration scenarios and provides an introduction to the SMC LIBRARY and SERVER commands as well as the StorageTek HTTP Server for OS/390 and z/OS.
- Chapter 3, “Starting the SMC” describes how to initialize the SMC software.
- Chapter 4, “SMC Allocation in a JES2 Environment” describes the SMC allocation function for a JES2 environment.
- Chapter 5, “SMC Allocation in a JES3 Environment” describes the SMC allocation function for a JES3 environment.
- Chapter 6, “SMC Message Handling” describes the SMC message intercept function.
- Chapter 7, “Operator Commands” describes the SMC operator commands.
- Chapter 8, “Recovery Procedures” describes procedures used when SMC or a library subsystem (HSC or MVS/CSC) becomes inactive.
- Chapter 9, “SMC Utilities” describes SMC system utilities.
- Chapter 10, “SMC Messages” describes the SMC system messages.
- Chapter 11, “StorageTek HTTP Server Messages” describes StorageTek HTTP Server messages.
- Appendix A, “Intercepted Messages” lists MVS, JES3, and TMS messages that are intercepted by the SMC.

- Appendix B, “Differences Between SMC and NCS 4.1 and Below” describes various differences between SMC and NCS 4.1, including those related to allocation and message handling.
- Appendix C, “Message Change Summary” lists SMC messages that have been added, changed or deleted for this release.

An index is also included.

Conventions Used in this Guide

Product Names

HSC refers to the MVS implementation of StorageTek’s Host Software Component. The VM implementation of HSC is not supported.

MVS/CSC refers to StorageTek’s Client System Component for MVS.

Typographic

In the JCL examples in this guide, some fields appear in lower case. You must update these fields to match your installation requirements.

Symbols

The following symbols are used to highlight text in this guide:



Note: Information that may be of special interest to you. Notes are also used to point out exceptions to rules or procedures.



Warning: Information necessary to keep you from damaging your hardware or software.



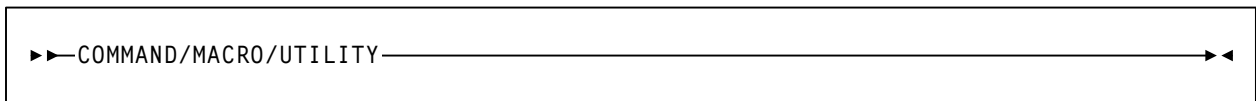
Caution: Information necessary to keep you from corrupting your data.

Syntax Flow Diagrams

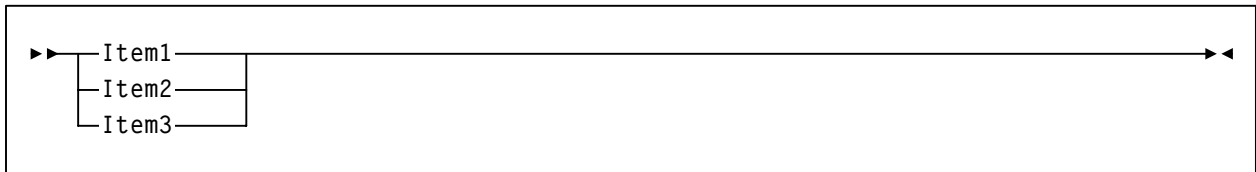
Syntax flow diagramming conventions include the following:

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.



or

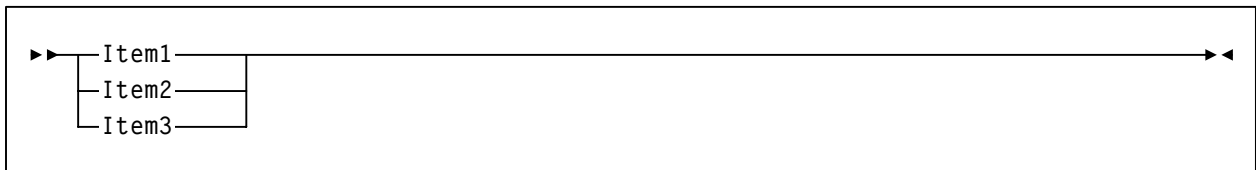


Diagrams are read left to right and top to bottom. Arrows indicate flow and direction.

- a statement begins with ►►
- a statement ends with ►◄
- diagrams continuing to the next line begin with ►
- fragments begin and end with |

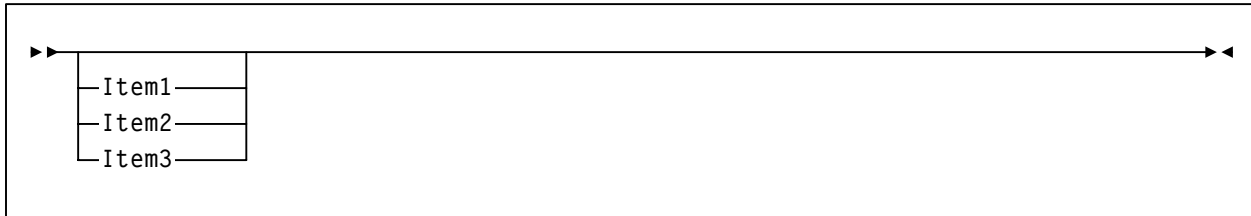
Single Required Choice

Branch lines (without repeat arrows) indicate that a single choice must be made. If one of the items from which a choice is being made is positioned on the base line of the diagram, a single choice is required.



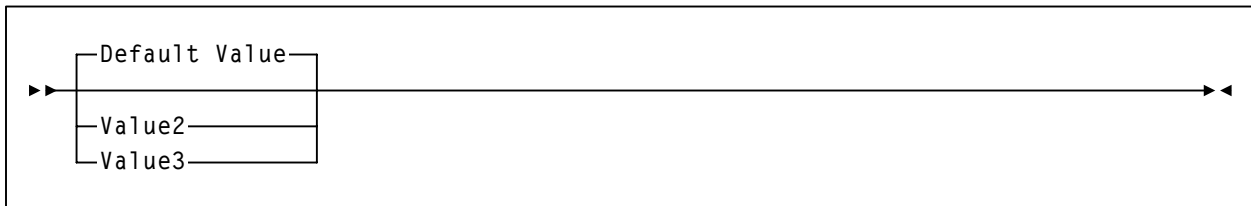
Single Optional Choice

If the first item is positioned on the line below the base line, a single choice of items in the stack is optional.

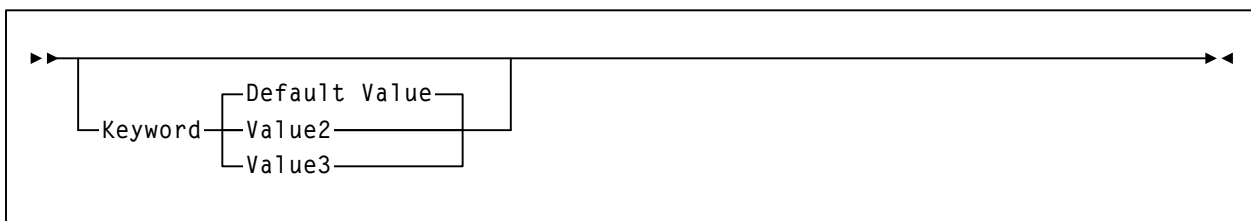


Defaults

Default values and parameters appear above the base line. In the following example, if a value is not specified with the command, the Default Value is used.



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 base line to indicate that they are optional, and the default value appears above the keyword line. In the following example, if the keyword is not specified with the command, the Default Value is used.



Repeat Symbol

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



Keywords

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

Variables

Italic type is used to indicate a variable.

Alternatives

A bar (|) is used to separate alternative parameter values.

Delimiters

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

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 (i.e., 1-9, or 010-094). Leading zeros are not required. The decimal portion is referred to as 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 referred to as 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, only the right-most portion is valid as the incremental range. For example:

A00B<u>00</u>	the largest range that can be specified is A00B00 through A00B99.
A0B<u>0</u>CC	the largest range that can be specified is A0B0CC through A0B9CC.
<u>000</u>XXX	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 referred to as 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:

<u>A00A0-A99A0</u>	increments VOLSERs A00A0 through A09A0, then A10A0 through A99A0.
<u>9AA9A-9ZZ9A</u>	increments VOLSERs 9AA9A through 9AZ9A, then 9BA9A through 9ZZ9A.
<u>111AAA-111ZZZ</u>	increments VOLSERs 111AAA through 111AAZ, then 111ABA through 111ZZZ
<u>999AM8-999CM8</u>	increments VOLSERs 999AM8 through 999AZ8, then 999BA8 through 999CM8
<u>A3BZZ9-A3CDE9</u>	increments VOLSERs A3BZZ9 through A3CAA9, then A3CAB9 through A3CDE9
<u>AAAAAA-AAACCC</u>	increments VOLSERs AAAAAA through AAAAAZ, then AAAABA through AAACCC
<u>CCCN NN-DDDN NN</u>	increments VOLSERs CCCN NN through CCCN NZ, then CCCNOA through DDDN NN *

* **Caution:** 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	26^1	26
AA-ZZ	26^2	676
AAA-ZZZ	26^3	17,576
AAAA-ZZZZ	26^4	456,976
AAAAA-ZZZZZ	26^5	11,881,376
AAAAAA-ZZZZZZ	26^6	308,915,776

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 Statements

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.
- /* and */ can be used to enclose comments in the job stream. Comments can be continued over multiple lines, but cannot be nested.
- The maximum length for a control statement is 1024 characters.

Related Publications

The following publications contain information about specific topics relating to the use of the Storage Management Component (SMC):

StorageTek Nearline Control Solution (NCS) Publications

- *NCS Installation Guide*
- *NCS User Exit Guide*
- *Requesting Help from Software Support*

StorageTek Host Software Component (MVS/HSC) Publications

- *MVS/HSC Configuration Guide*
- *MVS/HSC Operator's Guide*
- *MVS/HSC System Programmer's Guide*
- *MVS/HSC Messages and Codes Guide*

StorageTek Client System Component (MVS/CSC) Publications

- *MVS/CSC Configuration Guide*
- *MVS/CSC Operator's Guide*
- *MVS/CSC System Programmer's Guide*
- *MVS/CSC Messages and Codes Guide*

StorageTek Virtual Storage Manager Publications

- *VTCS Installation and Configuration Guide*
- *VTCS Administration Guide*
- *VTCS Messages and Codes Guide*
- *VTCS Command and Utility Reference*

IBM JES3 Publications

- *MVS/ESA JES3 Initialization and Tuning Reference*
- *OS/390 JES3 Initialization and Tuning Reference*

Technical Support

StorageTek Software Support and the StorageTek Customer Resource Center (CRC) maintain information about known SMC product updates. You can contact Software Support or access the CRC for the latest information available concerning product updates (i.e. documentation, PTFs, PUTs).

Refer to the *Requesting Help from Software Support* guide (included in the NCS package) for information about contacting StorageTek for technical support and for requesting changes to software products, or access StorageTek's CRC homepage at:

<http://www.support.storagetek.com>



Note: You must obtain a login ID and password in order to access the CRC. You can request a login ID and password from the CRC homepage.

Reader's Comments

We'd like to know what you think about this guide. E-mail your comments to us directly. Our Internet address is:

sid@stortek.com

Be sure to include the part number and title of the guide along with your comments.

Chapter 1. Introduction

What is SMC?

SMC, the Storage Management Component, is the interface between IBM's OS/390 and z/OS operating systems and Sun StorageTek™ NCS automated library control systems, HSC and MVS/CSC. SMC operates on both JES2 and JES3 systems and is a **required** NCS component. Its primary functions are:

- Influencing tape allocation according to hardware requirements and customer policies to ensure that appropriate tape drives are selected.
- Intercepting tape management, and operating system mount, dismount, and swap messages and translating them in order to request the required tape hardware functions from the appropriate NCS automated library control system.
- Coordinating requests among multiple StorageTek automated libraries.

SMC resides on every MVS host that accesses StorageTek real and virtual tape hardware. SMC may communicate with any number of StorageTek automated libraries, using cross address space facilities to communicate with HSC or MVS/CSC running on the same host, and TCP/IP to communicate with HSC systems executing on other hosts.



Note: For the purposes of this publication, **HSC** refers to the **MVS** implementation of the StorageTek Host Software Component. The VM implementation of HSC is not supported.

Terminology

SMC introduces three important terms:

library

A single StorageTek hardware configuration, normally represented by a single CDS.

drive exclusion

The SMC function of excluding drives for an allocation request based on SMC exclusion criteria.

See Chapter 4, “SMC Allocation in a JES2 Environment” and Chapter 5, “SMC Allocation in a JES3 Environment” for more information about this term.

drive prioritization

The SMC function of influencing selection of a particular drive based on allocation criteria, including volume location.

See Chapter 4, “SMC Allocation in a JES2 Environment” and Chapter 5, “SMC Allocation in a JES3 Environment” for more information about this term.

Chapter 2. SMC and StorageTek Library Management

Overview

This chapter describes sample SMC configuration scenarios and provides an introduction to the SMC LIBrary and SERVer commands as well as the StorageTek HTTP Server for OS/390 and z/OS.

The SMC provides the interface between IBM's OS/390 and z/OS operating systems and StorageTek NCS library control systems, HSC and MVS/CSC. These library control systems can operate in three ways:

- HSC or MVS/CSC can operate on the same local MVS host as the SMC
- The HSC can operate on a remote MVS host
- A combination of local and remote library control systems can be used.

In a simple configuration, with SMC residing on the same MVS host as HSC and/or one or more MVS/CSC systems each accessing a different library, the SMC automatically detects the library control system(s) using MVS interface facilities.

In a more complex configuration, where you wish to define remote HSC systems or enable simple server switching by defining multiple server paths to one or more libraries, you must use the SMC LIBrary and SERVer operator commands and the StorageTek HTTP Server for OS/390 and z/OS.

SMC LIBrary and SERVer Commands

A remote library is defined to SMC using the SMC LIBrary command. And, because a remote library can be shared by multiple instances of HSC running on different remote hosts, the path to each remote host is defined to SMC using the SMC SERVer command. See Chapter 7, "Operator Commands" for more information about these commands.

StorageTek HTTP Server for OS/390 and z/OS

The StorageTek HTTP Server for OS/390 and z/OS provides the middleware to allow communication between the SMC (client) and a remote HSC subsystem (server). The HTTP server executes as a separate subsystem on the MVS host where the remote HSC subsystem resides.

 **Notes:**

- Refer to the *NCS Installation Guide* for information about installing and configuring the StorageTek HTTP server.
- The StorageTek HTTP server is also packaged with the VTCS GUI product. However, for performance reasons, it is recommended that you do not use the same instance of the HTTP server for both SMC server programs and VTCS GUI programs. To execute multiple HTTP server subsystems on a single host, they must each listen on a unique PORT number.

Allocation and Multiple StorageTek Libraries

If your configuration includes multiple StorageTek library systems (as illustrated in Scenarios 5 and 6), SMC directs the allocation of each DD to the appropriate library based on TAPEREQ statements, user exits, specific volume locations, and scratch subpools.

SMC Configuration Scenarios

This section provides several scenarios for SMC in local, remote, and multiple library configurations.

Scenario 1: SMC and a Single Local HSC Library Configuration

In this scenario, the SMC and HSC execute on the same MVS host, as illustrated in the following figure:

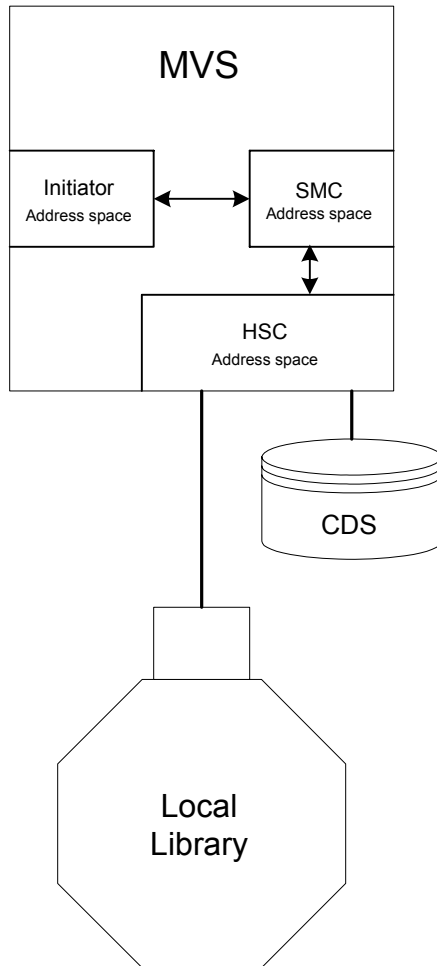


Figure 1. NCS 6.0 Local Library Configuration

This configuration utilizes three address spaces:

- the Initiator Address Space from which allocation and mount events originate
- the SMC Address Space, which intercepts those events
- the HSC Address Space, to which SMC sends requests for policy, drive, and volume data, and mount requests.

This local configuration does not require the StorageTek HTTP server, nor does it require any LIBRARY or SERVER commands. In this scenario, the SMC automatically detects the local HSC subsystem (and any MVS/CSC subsystems) by scanning the MVS SSCVT subsystem chain. The SMC then routes requests to subsystems on the same host using OS/390 and z/OS facilities for cross address space communication.

Optionally, you can enter LIBRARY commands to define the local subsystems. For example, the following LIBRARY command is used to define the local HSC library:

```
LIBRARY NAME(LCLLIB) LOCSUBSYS(HSC)
```

where LCLLIB is the desired name of the local library, and HSC is the local MVS subsystem name for the HSC.



Note: When you enter a LIBRARY command, the SMC facility to automatically detect local libraries is disabled until the SMC is recycled without any LIBRARY commands.

Once the local HSC has been defined by the LIBRARY command, if your configuration includes an MVS/CSC executing on the same host, a second library command is needed to define the CSC library. For example:

```
LIBRARY NAME(CSCLIB) LOCSUBSYS(CSC)
```

Scenario 2: SMC and a Single Remote HSC Library Configuration

In this scenario, the SMC communicates with a single HSC on a remote MVS host, as illustrated in the following figure:

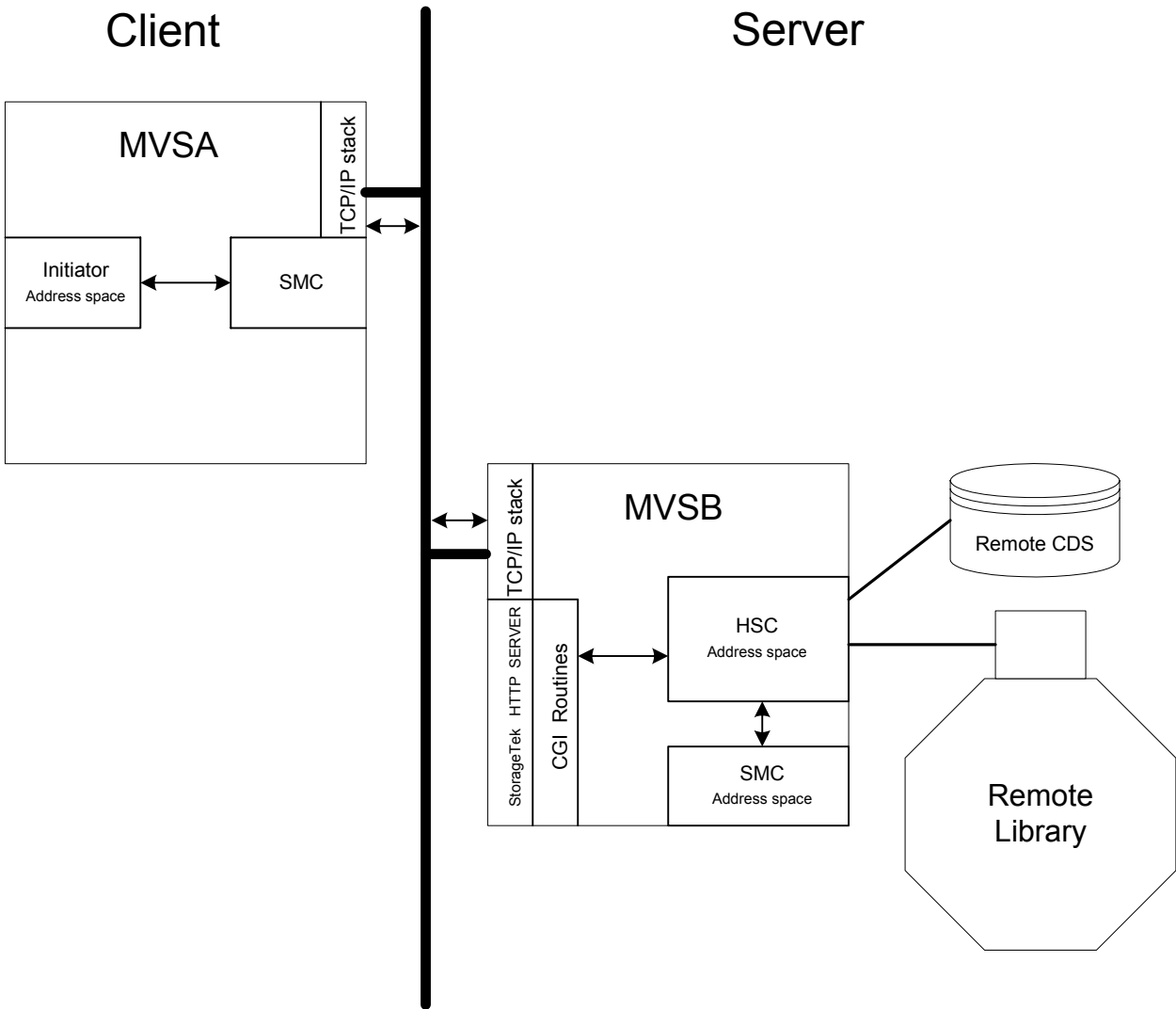


Figure 2. NCS 6.0 Remote Library Configuration



Note: HSC initialization requires the presence of the SMC. Therefore, the SMC must be present on the server side, even if you are not running local jobs.

In this scenario, the SMC acts as a client and the HSC acts as a server.

The SMC Address Space intercepts allocation and mount requests from an Initiator Address Space on MVSA and sends the requests through the TCP/IP stack to the designated StorageTek HTTP server on a remote host (MVSB). The HTTP server on MVSB executes the SMC CGI routine, which routes the request to the HSC executing on MVSB using OS/390 and z/OS facilities for cross address space communication.

Any configuration that includes a remote library requires the StorageTek HTTP server and LIBRARY and SERVER commands. The following example illustrates the LIBRARY and SERVER commands for this scenario:

```
LIBRARY NAME(REMLIB)  
SERVER NAME(REMPATH) LIBRARY(REMLIB) HOST(MVSB)
```



Note: When using SMC with a remote HSC server, the drive configuration defined on the server host **must** include all drives defined on the client host.

For example, assume that drives AB0-AB3 are defined on MVSA but are not defined on MVSB. In order for MVSA to perform allocations and mounts for drives AB0-AB3, these drives must be included in the HSC LIBGEN for MVSB. Since the drives are not actually defined on MVSB, they are not used for any local allocations or mounts on MVSB.

Scenario 3: SMC and a Single HSC Library Configuration with One Local and One Remote (alternate) Path

When a single library hardware configuration is shared between multiple MVS hosts, you can define multiple paths to the library as shown in the following figure:

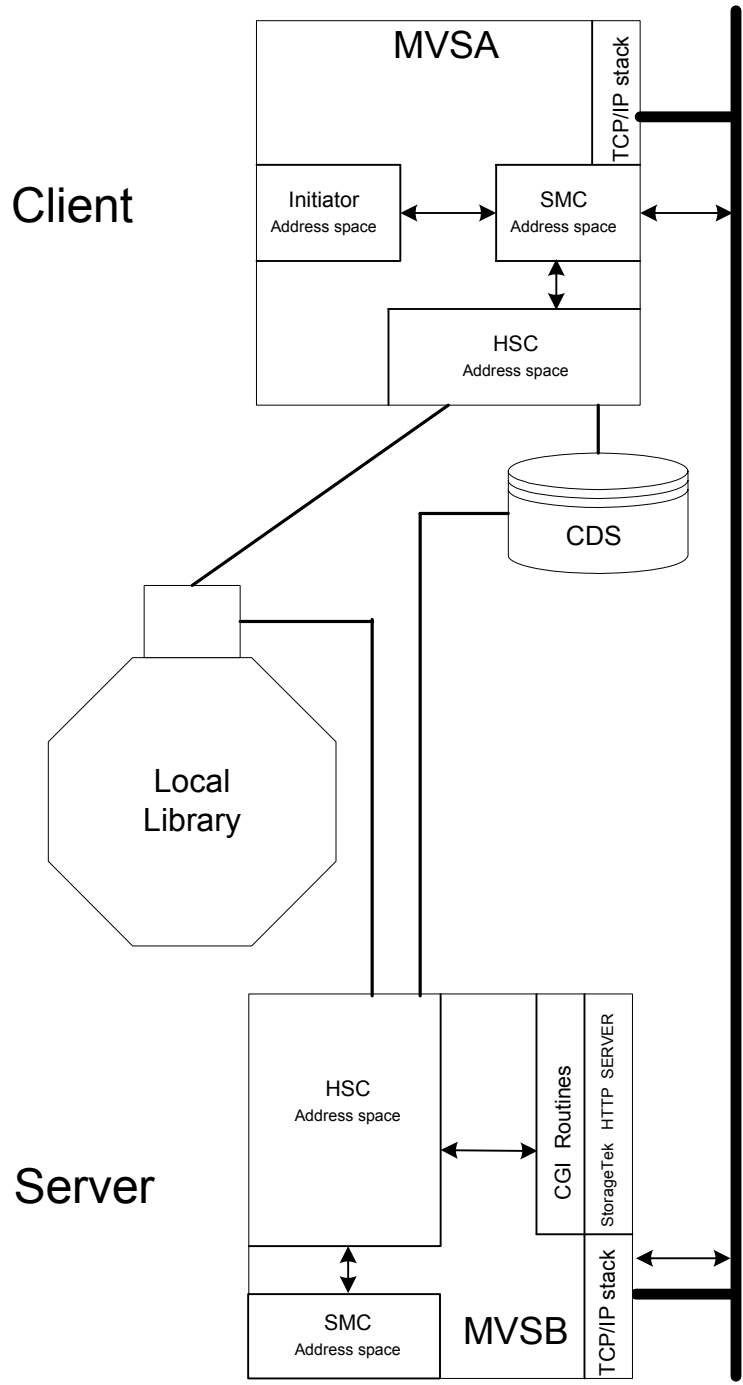


Figure 3. NCS 6.0 Local Library with Alternate Path Configuration



Note: HSC initialization requires the presence of the SMC. Therefore, the SMC must be present on the server side, even if you are not running local jobs.

The following example illustrates the LIBRARY and SERVER commands for this scenario:

```
LIBRARY NAME(SHRLIB) LOCSUBSYS(HSC)
SERVER NAME(REMPATH) LIBRARY(SHRLIB) HOST(MVSB)
```

In this scenario, if the HSC is active on host MVSA, the SMC communicates with the local HSC as described in Scenario 1. If the HSC on MVSA is inactive, the SMC communicates with the remote HSC as described in Scenario 2.

Primary And Alternate Path Selection Hierarchy To A Single Library With A Local Path

The path utilized by the SMC in this scenario is dependent upon the following factors:

- Whenever a local path is enabled to a library, the SMC attempts to use it.
- If a local path to a library is not enabled, or the local HSC is inactive, the SMC attempts to communicate with the first defined remote SERVER path. Once successful communication is established for a remote SERVER, it becomes the active path. All subsequent requests use the active path unless the local path becomes active or the remote SERVER path becomes unavailable.
- If a remote SERVER path becomes unavailable, the SMC attempts to use the next available SERVER in the order the SERVER commands were entered.

In this example, when HSC is active on host MVSA, the SERVER path defined to MVSB is not used. However, if the HSC on MVSA becomes unavailable, the SMC attempts to send its requests to the library using the REMPATh SERVER definition. When the HSC on MVSA becomes active again, the SMC automatically adjusts to use the HSC on MVSA.



Note: All drives defined on MVSB **must** be defined using the same drive addresses on the HSC CDS for MVSA.

Scenario 4: SMC and a Single HSC Library Configuration with Multiple Remote Server Paths

In this scenario, the SMC runs on a host with no local HSC, but with multiple paths to a remote HSC, as illustrated in the following figure:

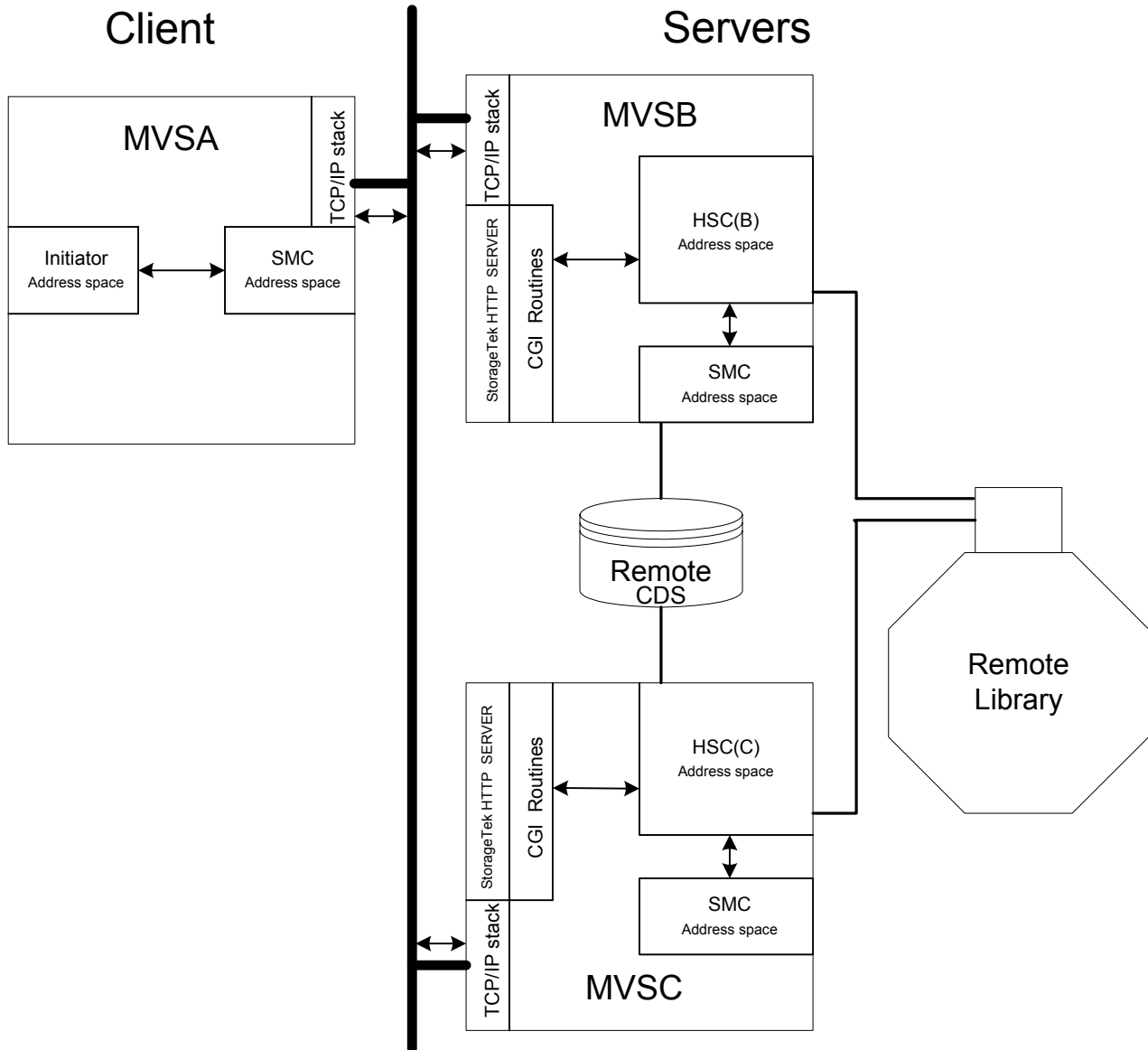


Figure 4. NCS 6.0 Remote Library Configuration with Two Server Instances



Note: HSC initialization requires the presence of the SMC. Therefore, the SMC must be present on the server side, even if you are not running local jobs.

The following example illustrates the LIBRARY and SERVER commands for this scenario:

```
LIBRARY NAME(SHRLIB)
SERVER NAME(REMPATH1) LIBRARY(SHRLIB) HOST(MVSB)
SERVER NAME(REMPATH2) LIBRARY(SHRLIB) HOST(MVSC)
```

Requests originating in an Initiator Address Space on MVSA are intercepted by the SMC Address Space on MVSA. The SMC on MVSA sends requests for volume and drive data, and mount requests to the SERVER on MVSB or MVSC.

The SMC attempts to communicate with the first defined server (MVSB). If this attempt fails, the SMC attempts to communicate with MVSC. Once successful communication with MVSC is established, the active path is not changed even if the server on MVSB becomes available. The following two methods can be used to change the active path from MVSC to MVSB:

1. Disable the REMPATH2 server using the following command:

```
SERVER NAME(REMPATH2) DISABLE
```

The SMC automatically attempts to communicate with the first server in the list (REMPATH1).

2. Issue the RESYNChronize REStart command. See “RESYNChronize Command” on page 141 for more information.

See “SMC Policy And Drive Type Information Synchronization” on page 16 for information about how the SMC acquires policy and drive type information from the HSC and MVS/CSC.

Scenario 5: SMC with Two Libraries; One Local and One Remote

In this scenario, the SMC can coordinate allocation and mount requests among multiple library hardware configurations as illustrated in the following figure.

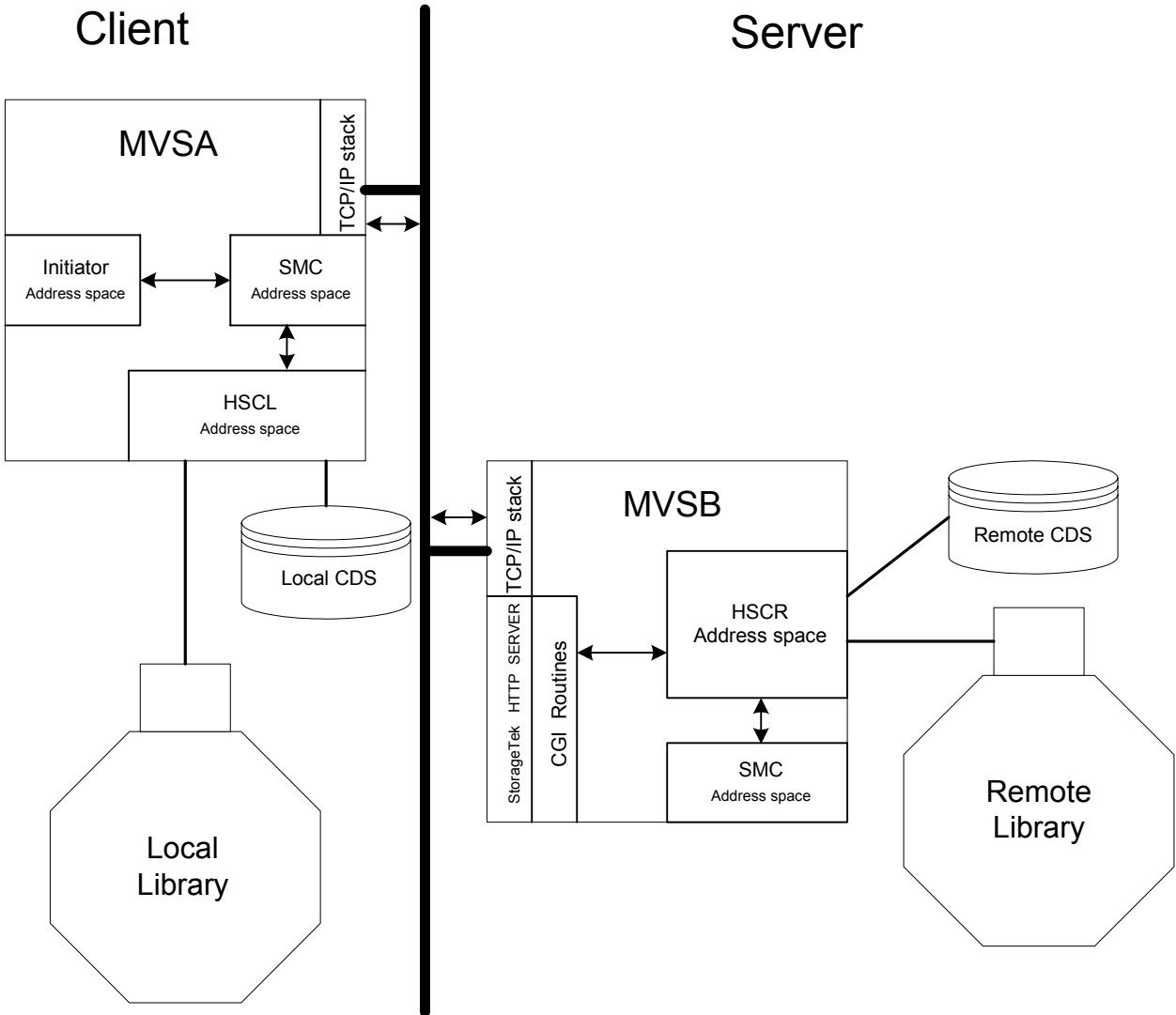


Figure 5. NCS 6.0 with Two Libraries - One Local and One Remote with Multiple Paths



Note: HSC initialization requires the presence of the SMC. Therefore, the SMC must be present on the server side, even if you are not running local jobs.

In this scenario, assume there are two separate libraries, each defined by a different CDS. In previous releases, this configuration was supported by executing MVS/CSC on MVSA and executing HSC with LibraryStation on MVSB. SMC 6.0 eliminates the need for MVS/CSC and LibraryStation in an MVS-only environment. The SMC can now communicate directly with the remote HSC using the StorageTek HTTP server.

Allocation and mount requests originating in an Initiator Address Space on MVSA are intercepted by the SMC on MVSA. These requests are then sent to either the local HSCL executing on the same host, or to the remote HSCR executing on the remote host MVSB.

The following example illustrates the LIBRARY and SERVER commands for this scenario:

```
LIBRARY NAME(LCLLIB) LOCSUBSYS(HSCL)
LIBRARY NAME(REMLIB)
SERVER NAME(REMPATH) LIBRARY(REMLIB) HOST(MVSB)
```

LIBRARY commands are required to define both the local and remote libraries.

See “SMC Library Subsystem Selection” on page 16 for information on how the SMC selects among multiple library subsystems to determine an “owner” for each allocation request (i.e. each DD in a job step may have a different library subsystem owner).

Scenario 6: SMC with Two Libraries; One Local and One Remote with Multiple Paths

In this scenario, multiple libraries are configured with multiple paths as illustrated in the following figure.

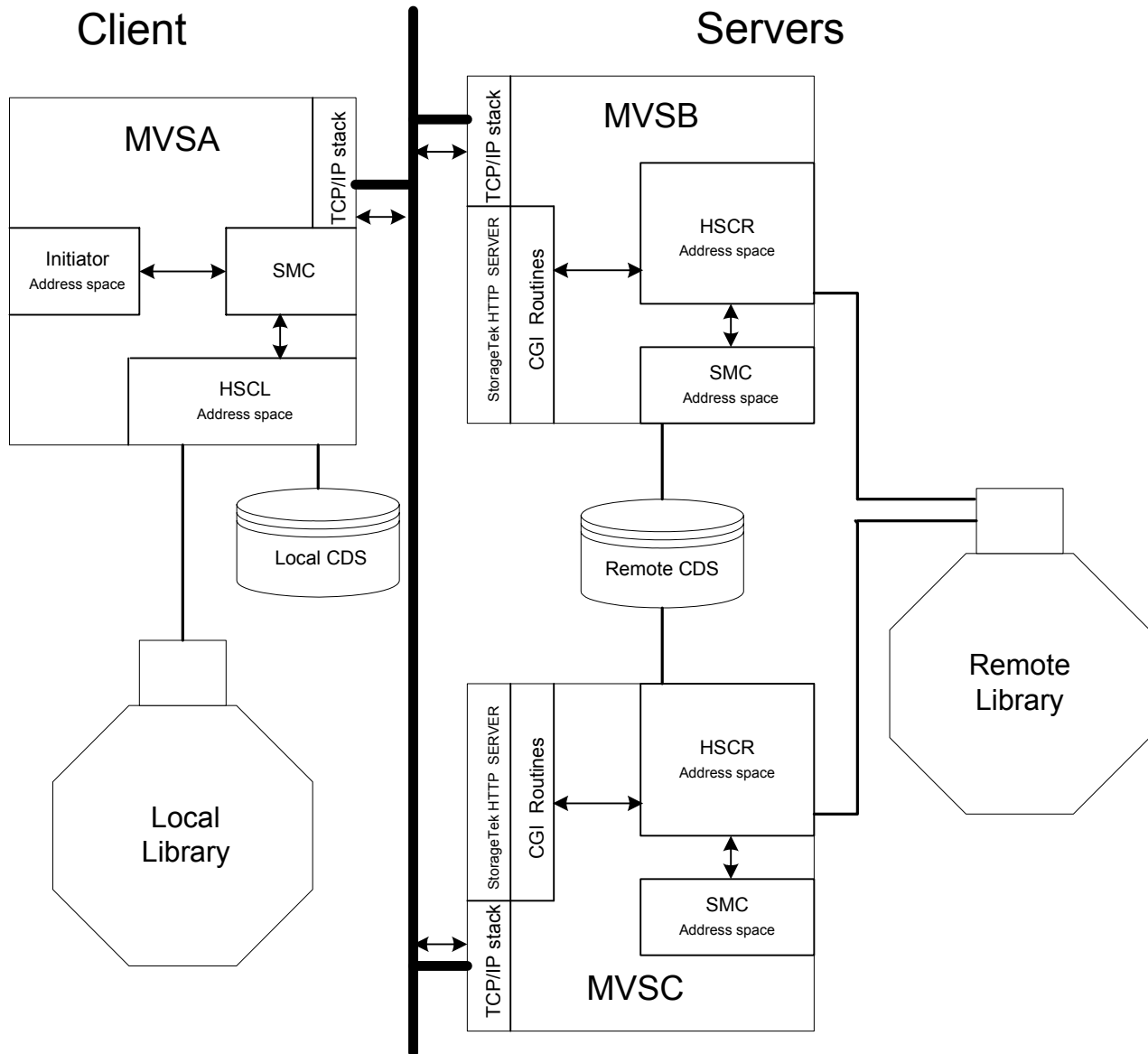


Figure 6. NCS 6.0 with a Local Library and a Remote Library with Two Server Instances



Note: HSC initialization requires the presence of the SMC. Therefore, the SMC must be present on the server side, even if you are not running local jobs.

This scenario is a combination of scenarios 1 and 4. The following example illustrates the LIBRARY and SERVER commands for this scenario:

```
LIBRARY NAME(LCLLIB) LOCSUBSYS(HSCL)
LIBRARY NAME(REMLIB)
SERVER NAME(REMPATH1) LIBRARY(REMLIB) HOST(MVSB)
SERVER NAME(REMPATH2) LIBRARY(REMLIB) HOST(MVSC)
```

SMC Policy And Drive Type Information Synchronization

The SMC acquires both policy information and drive type information from the NCS library control systems, HSC and MVS/CSC. For local paths, this information is automatically obtained whenever a new allocation or mount request is initiated. However for remote SERVER paths, this information is automatically acquired only the first time the SERVER is accessed. The RESYNChronization command is used to re-acquire policy and drive type information for all remote libraries. See “RESYNChronize Command” on page 141 for more information.



Note: When SMC acquires drive information from a remote host, only drives defined to that host on the CDS are returned. Therefore, all drives used by any client host **must** be defined on the server host, and all hosts in a client/server relationship **must** use the same drive addresses.

If you are executing in an environment where multiple hosts share drive definitions and are using the HSC CDS and UNITATTR statements specifying MODEL=IGNORE to limit the allocation of drives to specific MVS hosts, you may not be able to run with a single HSC as a remote server to multiple SMC hosts, since only the HSC image of drive definitions is available to the SMC host.

SMC Library Subsystem Selection

When the SMC intercepts a specific or scratch allocation request, it selects an owning library to service the request. The following criteria are evaluated by the SMC in the order shown to determine which library subsystem controls the allocation request:

1. Libraries are interrogated in the order they are defined. If LIBRARY commands are defined to the SMC, the order of the LIBRARY commands is used. If LIBRARY commands are not defined to the SMC, the subsystem order in the MVS SSCVT subsystem table is used.
2. If the Eligible Device List (EDL) for the request does not contain drives owned by a specific library, that library cannot own the request unless overridden by explicit subsystem or library selection.
3. If a specific requested volume is found in a library, that library is considered the owner unless overridden by explicit subsystem or library selection. If the volume is not found in a library, but that library contains a VOLATTR for that volume, then the library is considered the owner if the specific volume is not found in any other library.

4. If a library indicates that it has scratch volumes for the request, it is considered the owner unless overridden by explicit subsystem or library selection. If the library does not have scratch volumes for the request, but the specified subpool name is known to the library, then the library will be considered the owner if scratch volumes are not found in any other library.
5. If a user exit explicitly selects a subsystem name or library name that is active, that library is the owner, even if it does not contain any drives in the EDL.



Note: Unlike previous SMC releases, only a single version of an allocation user exit is executed, regardless of the number of libraries defined. Therefore, user exit return codes such as “USE LIBRARY DRIVES”, “USE VIRTUAL DRIVES”, etc. are **not** used to determine a request owner, nor does the user exit return code or “USE NONLIBRARY DRIVES” result in the selection of a “different” library.

The preferred way to select a library owner from among multiple libraries is through esoteric substitution on TAPEREQ. Alternatively, user exits can be used to specify a library name or esoteric.



Warning: If your pre-NCS 6.0 user exit uses the “USE NONLIBRARY DRIVES” return code to select an MVS/CSC subsystem in a mixed HSC and MVS/CSC environment, you must now use one of the methods described above to properly select a library.

MVS/CSC Library Issues

Converting MVS/CSC Systems to Use HSC and the StorageTek HTTP Server

If you have MVS/CSC library control software used only for MVS attached transports, it is recommended that you convert such installations to use remote HSC and StorageTek HTTP server facilities.

If your MVS/CSC library control software is used to control transports shared with open systems, it **cannot** be converted to use HSC and the StorageTek HTTP server.

If you wish to test the HTTP server communicating with HSC (replacing an MVS/CSC and LibraryStation communicating with the HSC), do the following:

1. Define a local MVS/CSC library and a second library for the remote HSC server.

```
LIBRARY NAME(LIB1) LOCSUB(CSC1)
LIBRARY NAME(LIB2) DISABLE
SERVER NAME(REM1) LIBRARY(LIB2)
```

2. To switch to the MVS/CSC, issue the following command:

```
LIBRARY NAME(LIB1) DISABLE
LIBRARY NAME(LIB2) ENABLE
```



Note: The SMC will **not** automatically switch from a local MVS/CSC to a remote HSC.

SMC LIBrary Commands and MVS/CSC Systems

If you have a mixed HSC and MVS/CSC environment and are using SMC LIBrary commands to define libraries and remote server paths, you cannot define both HSC and MVS/CSC subsystems using the same library name. That is, you **cannot** define a SERVER command for a LIBRARY where the local subsystem is an MVS/CSC. For example, the following is **not** valid (assuming subsystem CSC1 is an MVS/CSC system).

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

Chapter 3. Starting the SMC

Overview

The SMC manages all interfaces with MVS for allocation and message handling, and therefore must be started as a task on every MVS host where tape processing occurs.

The SMC calls on HSC and MVS/CSC for policy, volume, and drive information. Therefore, HSC 6.0 or MVS/CSC 6.0 can be active on the same host, or HSC 6.0 must be active on another host with the StorageTek HTTP server.



Note: The SMC **must** be initialized before HSC or MVS/CSC can complete initialization. These products detect the presence of an active SMC subsystem when they attempt to initialize. If the HSC or MVS/CSC is started before the SMC, the HSC or MVS/CSC issues message SLS4510A/SCS4510A, indicating that the SMC must be started before initialization can complete.

If the SMC is started without any active HSC or MVS/CSC subsystems and without any LIBRARY and SERVER commands, message SMC0049 is issued each time the SMC attempts to manage a tape allocation, indicating that SMC allocation processing has been bypassed. Start the HSC or MVS/CSC to correct this situation, or specify a LIBRARY and SERVER command to direct requests to a remote HSC.

SMC startup requires the following tasks:

- creating the SMC START procedure
- executing the SMC START procedure.



Note: Refer to the *NCS 6.0 Installation Guide* for SMC installation procedures and JES3 post-installation tasks.

Creating the SMC START Procedure

An SMC START procedure must be created in the procedure library of the host system. The MVS START command invokes this catalogued procedure, thus activating the SMC with the specified startup parameter settings.

Figure 7 provides a sample SMC START procedure.

```
//yourprocname PROC PRM='WARM'  
//stepname EXEC PGM=SMCBINT,REGION=4M,TIME=1440,  
// PARM='&PRM'  
//*  
//STEPLIB DD DSN=linklib_name,DISP=SHR  
// DD DSN=exitlib_name,DISP=SHR  
//*  
//SMCPARMS DD DSN=parmlib_name(param_member_name),DISP=SHR  
//*  
//SMCCMDS DD DSN=cmdlib_name(cmd_member_name),DISP=SHR
```

Figure 7. Sample SMC START Procedure



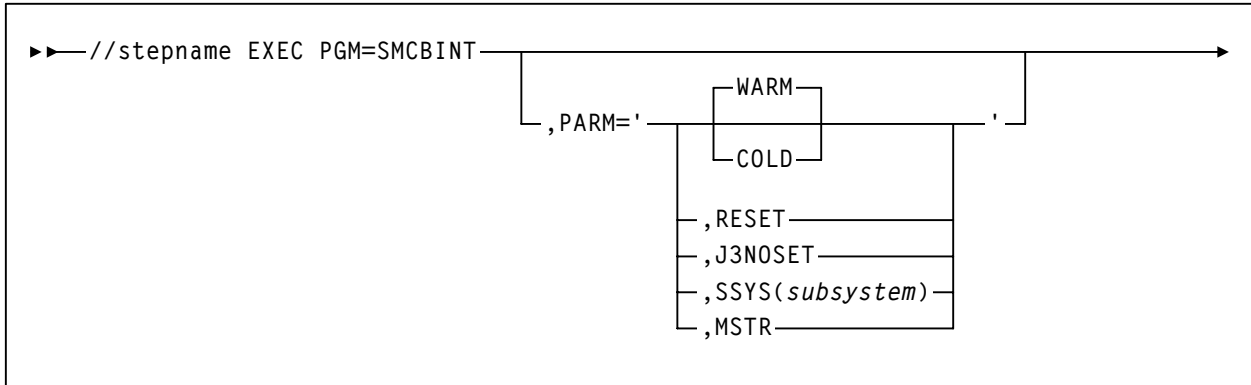
Notes:

- The first four characters of *yourprocname* specify the SMC subsystem name (unless the SSYS startup parameter is specified). Sun Microsystems recommends a value of SMCx, where *x* is any valid jobname character.
- The SMCPARMS and SMCCMDS DD statements are optional. Both statements are processed during initialization. The SMCCMDS data set may be re-processed during execution. See Chapter 7, “Operator Commands” on page 117 for information about SMC commands that may be specified in either of these data sets.
- TIME=1440 must be coded to ensure that the SMC does not time out and terminate.

SMC EXEC Statement

The EXEC statement is used to define SMC startup parameter settings.

EXEC Statement Syntax



EXEC Statement Parameters

PARM=

defines the list of parameters passed to the SMC initialization routine.



Note: Execution parameters must be separated with commas. Separating parameters with blanks results in a syntax error.

WARM

specifies that the SMC main control block is not rebuilt. This is the default setting for normal operation.

COLD

specifies that all SMC control blocks are rebuilt. This parameter is mutually exclusive with WARM.

SSYS

specifies a *subsystem* ID that is different from the first four characters of the SMC START procedure. The SMC searches for this subsystem ID during initialization.

subsystem must be one to four characters in length.

RESET

specifies that the active subsystem status flag in the MVS Subsystem Communications Vector Table (SSCVT) for the SMC is reset. This parameter may correct a situation in which the SMC was terminated abnormally. It can be specified with WARM or COLD.



Warning: Using this parameter when an SMC subsystem is active and functional causes unpredictable results.

J3NOSET

indicates that a JES3 system is not using JES3 tape setup. When this parameter is specified, allocation influencing behaves as described for JES2.

MSTR

specifies that the SMC start under the MSTR subsystem instead of under JES.

When specifying this parameter, you **must** also perform **one** of the following actions:

- Start the SMC subsystem using SUB=MSTR on the MVS Start command.
- Add the SMC subsystem to the IEFSSNxx subsystem table using the keyword format.



Notes:

- This parameter is not supported for JES3 with SETUP environments.
- If you wish to run the SMC under the master MVS subsystem, the PROCLIB containing the SMC START procedure must be present in the PROCLIB concatenation for the master address space. This concatenation is defined in SYS1.PARMLIB(MSTJCLxx), under DD IEFPSI.

Executing the SMC START Procedure

The MVS START command is used to initialize the SMC software. When this command is issued, the SMC subsystem initialization routine is invoked. This routine determines what parameters are in effect, performs any cleanup necessary, and begins normal processing.

Parameters associated with PARM= on the EXEC statement of the SMC Start Procedure can also be supplied via PARM= on the MVS START command. The PARM= specification on the MVS START command overrides the PARM= specification in the SMC Start Procedure. See “EXEC Statement Parameters” on page 21 for parameter descriptions.

MVS START Command Syntax

```
▶▶—START—smc-proc-name————▶◀
```

MVS START Command Parameters

START or S

initiates the MVS START command

smc-proc-name

indicates the name of the SMC START procedure member.

Chapter 4. SMC Allocation in a JES2 Environment

Overview

This chapter describes SMC allocation in a JES2 or JES3 without SETUP environment. The following topics are discussed:

- jobs selected for SMC allocation management
- DFSMS esoteric substitution
- drive exclusion
- affinity separation
- drive prioritization
- deferring mounts
- multiple library subsystem support
- SMC DFSMS processing
- interaction with other software



Note: See Appendix B for differences between SMC and NCS allocation.

Jobs Selected for SMC Allocation Management

All I/O device allocations on a JES2 system are examined by the SMC to determine if it should process the allocation request.

The SMC uses the MVS subsystem interface (SSI) IEFJFRQ Subsystem Function Request exit to gain control during tape allocation events. The SMC takes control in the JES2 environment for these Subsystem Functions:

- SSI55 - DFSMS Interpreter/Dynamic Allocation Exit (IDAX)
- SSI24 - common allocation
- SSI78 - tape allocation.

SSI55 Interpreter/Dynamic Allocation Exit (IDAX)

During MVS JCL interpretation processing, IDAX provides an opportunity to examine new data set allocations and to replace the unit parameter value with a different value (esoteric).



Note: This function is optional for SMC customers. For information about implementing DFSMS esoteric substitution, refer to “SMC DFSMS Processing” on page 46.

SSI24 Common Allocation

During SSI24 common allocation processing, the SMC performs the following processes to arrive at the best set of eligible drives:

- drive exclusion
- unit affinity separation
- defer processing (when CA1RTS is set to ON)
- EDL updated with the drive exclusion results (when MIACOMPAT is set to ON)

The results of the drive exclusion process are not reflected in MVS control blocks until tape allocation time, unless MIACOMPAT is set to ON.

The results of unit affinity separation are used to update MVS VOLUNIT entries in the SIOT.

SSI78 Tape Allocation

During SSI78 tape allocation processing, the SMC performs the following:

- updates to MVS control blocks based on drive exclusion results (unless MIACOMPAT is set to ON)
- drive prioritization
- mount deferral (unless CA1RTS is set to ON)

The SMC sets all unacceptable drives to ineligible status and assigns a priority to each drive that remains eligible for the allocation. The higher the priority, the more likely the device will be chosen for the mount.

The mount may be deferred until the data set is opened. The customer's library subsystem defer policy determines whether or not the mount is deferred.

The SMC implements these decisions by updating the IEFSSSTA control blocks for deferral, drive exclusion, and prioritization during SSI78 processing.

Exceptions

The SMC does not influence the following types of cartridge tape allocation:

- demand allocation (i.e., request for a specific drive(s))



Note: The SMC does perform DEFER processing for demand allocation.

- allocations excluded explicitly by entering the ALLOCJob command BYPASS parameter. Refer to the description of this command in Chapter 7, "Operator Commands".
- allocations where the list of eligible devices contains only devices that are "unknown" (i.e. not virtual, not library, and not present in the NONLIB esoteric).
- DFSMS-managed allocation. An SMS-managed data set is defined as a data set that has a storage class defined. A storage class is assigned when either
 - the STORCLAS parameter is specified on the DD statement, or
 - an installation-written ACS routine selects a storage class for a new data set.

DFSMS Esoteric Substitution

During MVS JCL interpretation processing, the Interpreter/Dynamic Allocation Exit (IDAX) provides an opportunity to examine new data set allocations and to replace the unit parameter value with a different value (esoteric).



Note: This function is optional for SMC customers. For information about implementing DFSMS esoteric substitution, refer to “SMC DFSMS Processing” on page 46.

Drive Exclusion

When the SMC determines an interest in an allocation event at common allocation time (SSI24 subsystem function), its first step is to narrow down the list of eligible drives to those that best support the allocation request. This step is called *drive exclusion*.

The SMC performs drive exclusion by building a list of drives contained in the initial EDL and then by following a step-by-step process using an ordered list of exclusion criteria to remove drives from this list.

Like the HSC and MVS/CSC device separation processes, the SMC drive exclusion process does **not** take into account the status of drives (e.g., offline, busy) when selecting drives that are eligible for the allocation request. If all drives compatible with the allocation request are unavailable, the job goes into allocation recovery.

Before any drives are removed from consideration for the allocation, the SMC first determines whether a job step can be allocated.

For example:

```
//JOB1 JOB ...
//STEP1 EXEC PGM=...
//DD1 DD UNIT=3490,...
//DD2 DD UNIT=3490,...
.
.
.
//DD25 DD UNIT=3490,...
```

Assume STEP1 is requesting twenty-five 3490 drives. If the MVS EDL contains twenty-four 3490 drives in its lists of eligible devices, MVS cannot allocate the required drives for the job. In this instance, the SMC issues message SMC0042, which indicates the job cannot allocate before SMC modification, and allows the job to fail allocation.

Once the SMC determines a job step can be allocated, the SMC begins excluding drives based on an ordered list of criteria. Table 1 on page 29 and Table 2 on page 34 list these criteria and show the order the exclusion process follows. The lower the criteria level, the higher its importance. Thus, level 2 is considered more important than level 6.

Minimum Exclusion Level

By default, level 2 is the *minimum exclusion level*, that is, the desired minimum level of allocation exclusion, for both specific and scratch volume processing. However, if necessary, the ALLOCDef and ALLOCJob commands can reset the minimum exclusion level (refer to Chapter 7, “Operator Commands” for descriptions of these commands).



Note: If your VTCS configuration allows recalls of all migrated virtual volumes to any VTSS, it is recommended that you set the minimum exclusion level to 1 using the ALLOCDef command. See “Exclusion Level 2” on page 30 for more information.

In the first step of drive exclusion, the SMC determines if the job can be allocated at the minimum drive exclusion level. If not enough drives remain eligible at the minimum exclusion level, the SMC issues message SMC0043, which indicates the job is not allocatable at the minimum exclusion level. The SMC then excludes all devices, causing the job to fail allocation.

For example:

```
//JOB2 JOB ...
//STEP1 EXEC
//DD1 DD UNIT=3490,VOL=SER=984001,DISP=OLD
//DD2 DD UNIT=3490,VOL=SER=984002,DISP=OLD
//DD3 DD UNIT=3490,VOL=SER=984003,DISP=OLD
```

In this system environment, assume that twenty 3490-type drives have been configured. Only two of those twenty are actually T9840 drives. If all three volumes require a separate T9840 tape drive, then not enough T9840 drives exist for the job to allocate. Without SMC influence, MVS allows the job to proceed through allocation. However, with SMC influence, message SMC0043 is issued and the job fails.



Note: No exclusion criteria (including minimum level) are applied when the user specifies demand allocation (i.e., requests a specific drive or drives).

Subsequent Levels

Once the minimum level of exclusion has succeeded, the remaining exclusion levels are performed. Any level above the minimum level that fails is skipped and exclusion continues to the next level.



Note: A level fails if it results in too few drives for the DD statement to allocate. The only required exclusion levels are those at the minimum level or below.

After the last drive exclusion level is completed, the SMC examines the final drive lists. If the job step cannot be allocated because not enough drives remain eligible, drive exclusion regresses to the previous exclusion level for the failing DD statement. This process repeats until the job step can allocate.

Intentionally Failing a Job

Depending upon the characteristics of your installation, you may want specific jobs to fail at allocation time rather than execute them with non-optimal devices. As an example, if you have two ACSs, one of which is located at a remote site, you may prefer to fail the job, rather than having to transfer the correct volume from one ACS to another.

For example:

```
//DD1 DD UNIT=CART ,VOL=SER=INACS0 ,DISP=OLD  
//DD2 DD UNIT=AFF=DD1 ,VOL=SER=INACS1 ,DISP=OLD
```

Assume the following:

- Both volume INACS0 and INACS1 share common media characteristics.
- Volume INACS0 resides in ACS0, and volume INACS1 resides in ACS1.
- Only ACS0 contains a compatible drive.

If this affinity chain is not broken, then volume INACS1 must be ejected from ACS1 and entered into ACS0 for this job to execute, which may not be possible given the geographical locations of the ACSs.

In this case, failure at allocation time may be preferable to failure at run time. By specifying the correct MINLVL value (i.e., by issuing the ALLOCJob command with a MINLVL of 7 for this job), the job is failed at allocation time.



Caution: Setting the minimum exclusion level may cause jobs to fail unexpectedly. Whenever the SMC is unable to apply an exclusion criterion at or below the minimum level, the job fails allocation.

For example, if a TAPEREQ specifies a media of STANDARD for a scratch volume and the eligible device list contains only T9840 drives, when the SMC attempts to apply the “policy media” exclusion level, all drives are excluded. If the minimum level is set to 3 or higher, the job fails allocation. Therefore, it is recommended that the default exclusion level (as specified by the ALLOCDef command) should normally not be set to a level greater than 2.

Specific Volume Request Drive Exclusion Criteria

For a specific volume allocation, the SMC excludes drives in order from lowest exclusion level to highest based on the criteria in Table 1. The lower the level number, the more important the exclusion criteria.

Table 1. JES2 Drive Exclusion Levels (Specific Request)

Level	Specific Volume Criteria	Keyword*
1	For non-labeled (NL) specific volume requests, exclude all virtual drives. Exclude drives incompatible with the volume media. Primary source: external volume label Secondary source: VOLATTR MEDIA parameter The volume media can be obtained from the volume label or from an HSC VOLATTR statement MEDIA parameter.	VIRTUALLABEL MEDRECTECH
2**	For virtual volumes only, exclude drives not located in an accessible VTSS. Note: No further exclusion is performed for virtual volumes.	VTSSLOCATION
3	Exclude drives based on the required recording technique. Source: VOLATTR RECTECH parameter.	VOLATTRRECTECH
4	Exclude drives based on user location policies. Primary source: TAPEREQ ESOTERIC parameter. Secondary source: User Exit 08 or 10 return codes.	USERPOLICY
5	Exclude drives based on the HSC ALLOC command SPECVOL parameter. Note: This exclusion level does not apply to MVS/CSC.	SPECVOL
6	Exclude drives based on volume location type, i.e., library or nonlibrary.	LOCTYPE
7	Exclude drives based on the ACS location of the volume, if the volume resides in the library.	ACSLOCATION
8	Exclude drives based on the requested recording technique. Primary source: DFSMS data class recording technique. Secondary source: TAPEREQ RECTECH parameter.	POLRECTECH

* Keywords associated with each exclusion level are specified in the exclusion criteria displayed in message SMC0046.

** Level 2 is the default minimum level.

Exclusion Level 1

The SMC excludes drives that are not compatible with the volume media. The volume media can be obtained from the volume label or from an HSC VOLATTR statement MEDIA parameter.

The SMC excludes all virtual drives for non-labeled (NL) specific volume requests.

Exclusion Level 2

If the specific volume requested is a virtual volume, the VTSS location for a resident volume must match the volume location. The VTSS location for a migrated virtual volume must match the location to which the volume can be recalled. The remaining exclusion levels do not apply to virtual volumes and are bypassed.



Note: If all VTSSs in your virtual configuration allow migration to common ACSs, you may want to set the minimum level exclusion level to 1 to allow jobs to proceed by migrating and recalling resident VTVs.

Refer to the VTCS documentation for more information about virtual volumes.

Exclusion Level 3

If an HSC VOLATTR statement specified the recording technique (RECTECH parameter) for this specific volume, the SMC excludes drives that do not provide that recording technique.

For example:

```
//DD1 DD UNIT=TLACS0 ,VOL=SER=VOL000 ,DISP=OLD
```

A VOLATTR statement has been defined to the HSC.

```
VOLATTR SERIAL(VOL000) RECTECH(36B) MEDIA(ECART)
```

VOL000 requires a 9490 drive, and if TLACS0 contains both 9490 and 9490EE drives, the 9490EE drives are excluded from consideration for this allocation.



Note: In MVS/CSC, the VOLATTR RECTECH is implied by the composite recording technique of the drives in the list returned by the library server.

Exclusion Level 4

If a TAPEREQ statement specifies an esoteric unit name for this allocation request, the SMC verifies the esoteric unit name is defined to MVS. If the esoteric passes this test, any drives in the original EDL that are not also defined to the esoteric unit name are excluded from consideration for this allocation.

For example:

```
//DD2 DD UNIT=CART ,VOL=SER=TR0001 ,DSN=SYS4 .TR1 .DATA ,DISP=OLD
```

The following TAPEREQ statement is defined to HSC or MVS/CSC.

```
TAPEREQ DSN(SYS4.TR1.***) ESOT(TLACS1) RECT(LONGI)
```

The drives that are not defined to the esoteric TLACS1 are excluded from consideration for this allocation.

If no TAPEREQ statement specifies an esoteric for this allocation request, User Exit 08 or 10 return codes and values can define the criteria for drive exclusion at this level. The user exit return codes can direct the SMC to select nonlibrary drives, a specific ACS, or the user exit can return an esoteric name that contains customer-selected drives. Any drives in the current list of eligible devices that did not match the user exit return values are excluded here.

For example:

```
//DD1 DD UNIT=CART,VOL=SER=VOL001,DISP=OLD
```

User Exit 08 returns UX08ASUB and ACS0, which requests library drives and ACS0. The drives that are not in ACS0 are excluded from consideration for this allocation.

Exclusion Level 5

The SPECVOL parameter of the HSC ALLOC command instructs the SMC which library ACS locations can be used for this request if one of two circumstances exists:

- the specific volume is not in the library and no drives are defined in the nonlibrary esoteric to the HSC, or
- the specific volume is not in the library, and user exit 08 requested a library location.

Drives not in SPECVOL ACS locations are excluded from consideration.



Note: This exclusion level does not apply to MVS/CSC.

For example:

```
//DD1 DD UNIT=CART,VOL=SER=NL001,DISP=OLD
```

Volume NL001 is outside the library, and the HSC command ALLOC SPECVOL(00) has been issued.

If only library drives are defined to the HSC, only the drives defined to ACS00 remain under consideration for this allocation.

Another example:

```
//DD2 DD UNIT=CART,VOL=SER=NL002,DISP=OLD
```

Assume volume NL002 is not in the library, and the HSC command ALLOC SPECVOL ACS(01) has been issued to allow ACS01 to be used for specific nonlibrary volumes.

If nonlibrary drives have been defined on the HSC LIBGEN SLILBRY macro but HSC user exit 08 returned UX08SVOL, all nonlibrary and drives in ACS00 are excluded from consideration for this allocation.

Exclusion Level 6

The generic location of the volume (library or nonlibrary) reduces the remaining list of eligible drives.



Note: For exclusion level 6, nonlibrary drives are outside the library and contain **either** known (from the HSC>NNLBDRV or MVS/CSC NONLIB esoteric) or unknown device characteristics.

If the volume resides in a library, all drives outside the library are excluded, and if the volume resides outside the library, all library drives are excluded.

For example:

```
//DD1 DD UNIT=CART,VOL=SER=LIB001,DISP=OLD
```

Assume volume LIB001 resides in a library.

All drives outside the library are excluded from consideration for this allocation.

Exclusion Level 7

For volumes residing in the library, the ACS location of the volume reduces the remaining list of eligible drives. Any drives that remain that do not reside in the same ACS as the volume are excluded.

For example:

```
//DD1 DD UNIT=CART,VOL=SER=A00VL1,DISP=OLD
```

Assume volume A00VL1 resides in ACS00, and the drives reside in ACS00 and ACS01.

The drives residing in ACS01 are excluded from consideration for this allocation.

Exclusion Level 8

Exclusion levels 1 and 3 restrict the list of eligible drives to those compatible with the volume's actual media and its recording technique, if specified on the HSC or MVS/CSC server VOLATTR statement. Exclusion level 8 may further restrict the drives for the request based on the TAPEREQ or DFSMS data class recording technique.

The HSC or MVS/CSC TAPEREQ definitions can explicitly denote a recording technique for the allocation request. DFSMS data class definitions can also specify an 18- or 36-track recording technique for the request and override any TAPEREQ recording technique specification.

For example:

```
//DD1 DD UNIT=CART,VOL=SER=TV9840,DISP=NEW,  
// DSN=SYS4.TR4.DATA
```

Assume the requested volume is a 9840 volume in the library with no recording technique specified on a VOLATTR. Assume an HSC TAPEREQ has been defined:

```
TAPEREQ DSN(SYS4.TR4.** ) RECT(STK1RB)
```

Previous exclusion levels restricted the list of devices to all T9840 models. In this example the TAPEREQ recording technique of STK1RB leaves only T9840B devices in consideration for this allocation.

Scratch Volume Request Drive Exclusion Criteria

For a scratch volume allocation, the SMC excludes drives in order from lowest exclusion level to highest based on the criteria in Table 2. The lower the level number, the more important the exclusion criteria.

Table 2. JES2 Drive Exclusion Levels (Scratch Request)

Level	Scratch Volume Criteria	Keyword*
1	For nonlabeled (NL) scratch volume requests, exclude all virtual drives.	VIRTUALLABEL
2**	No level 2 for scratch volume exclusion.	
3	Exclude drives based on the requested media. Primary source: DFSMS data class media specification. Secondary source: TAPEREQ MEDIA parameter. Tertiary source: User Exit 02 request for virtual media.	POLMEDIA
4	Exclude drives based on user location policies. Primary source: TAPEREQ ESOTERIC parameter. Secondary source: HSC and MVS/CSC User Exit 02 and 10 return codes.	USERPOLICY
5	Exclude drives based on the media of available scratch volumes in subpool. Primary source: TAPEREQ SUBPOOL parameter. Secondary source: User Exit 02 subpool values. Tertiary source: scratch subpool 0.	SUBPOOL
6	Exclude drives based on available scratch location type, i.e., library, nonlibrary, or virtual.	LOCTYPE
7	Exclude drives based on the HSC ALLOC or MVS/CSC ALTER command ZEROSCR parameter.	ZEROSCRATCH
8	Exclude drives based on the requested recording technique. Primary source: DFSMS data class recording technique. Secondary source: TAPEREQ RECTECH parameter.	POLRECTECH

* Keywords associated with each exclusion level are specified in the exclusion criteria displayed in message SMC0046.

** The MINLVL parameter value default, level 2 on the ALLOCDef or ALLOCJob command, applies to both scratch and specific volume requests. Even though level 2 has no meaning for scratch, level 2 is considered the default minimum level.

Exclusion Level 1

The SMC excludes virtual drives if the volume label type requested is NL (nonlabeled). This is the only required criterion. Refer to the VTCS documentation for more information about virtual volumes.

Exclusion Level 2

No level 2 scratch volume exclusion is performed.

Exclusion Level 3

Under HSC and MVS/CSC, a DFSMS data class definition can explicitly request the media desired, or the media can be derived from the DFSMS data class recording technique. Drives that do not provide the requested media support are excluded at this level.

If no DFSMS data class definition applies for this allocation, the HSC or MVS/CSC TAPEREQ definitions can explicitly specify the media desired for the allocation request, or media can be derived from the TAPEREQ RECTECH. Drives that do not provide the requested media support are excluded at this level.

For example:

```
//DD1 DD UNIT=CART,DISP=(NEW,KEEP),DSN=SYS4.VIRT.DATA
```

No HSC TAPEREQ has been defined and no DFSMS data class definition applies. Assume the HSC User Exit 02 return code requested virtual media (UX02VIRT). The HSC User Exit 02 media selected for this request is virtual. Only virtual drives remain in consideration for this allocation. The next example shows how TAPEREQ can override User Exit 02 return codes.

For example:

```
//DD1 DD UNIT=CART,DISP=(NEW,KEEP),DSN=SYS4.TR4.DATA
```

An HSC TAPEREQ has been defined:

```
TAPEREQ DSN(SYS4.TR4.** ) MEDIA(DD3A) RECT(DD3)
```

Assume User Exit 02 return code requested virtual media (UX02VIRT).

If no DFSMS data class definition or TAPEREQ definition applies to this allocation request, User Exit 02 return code, UX02VIRT, can explicitly request virtual media. Non-virtual drives are excluded at this level.

However, the TAPEREQ media definition overrides the user exit return code, and the media selected for this request is helical. Only helical drives remain in consideration for this allocation. Messages SMC0045 and SMC0046 are issued indicating that the user exit request for virtual volumes was not honored.

Exclusion Level 4

If a TAPEREQ statement specifies an esoteric for this allocation request, the SMC determines the common drives between the TAPEREQ esoteric and the list of remaining eligible drives. If common drives exist, any drives remaining in the current list of eligible drives that are not also defined to the esoteric are excluded from consideration for this allocation.

If no TAPEREQ statement specifies an esoteric for this allocation request, User Exit 02 and User Exit 10 return codes and values can define this criterion for drive exclusion. The user exit return codes can direct the SMC to select library drives, nonlibrary drives, or a specific ACS, or the user exit can return an esoteric unit name that contains customer-selected drives. Any drives in the current list of eligible drives that did not match the user exit return values are excluded here.

For example:

```
//DD1 DD UNIT=CART,DSN=&&TEMP1,DISP=(NEW,PASS)
```

Assume User Exit 02 returns UX02LIB, which requests library drives.

The drives that are not in the library are excluded from consideration for this allocation.

A TAPEREQ example is:

```
//DD2 DD UNIT=CART,DSN=SYS4.TR1.DATA,DISP=(NEW,KEEP)
```

The following TAPEREQ statement is defined to the HSC or MVS/CSC.

```
TAPEREQ DSN(SYS4.TR1.** ) ESOT(TLIB9840) RECT(LONGI)
```

The drives that are not defined to the esoteric TLIB9840 are excluded from consideration for this allocation. Any User Exit 02 return codes would be ignored.

Exclusion Level 5

The combined media and specified recording technique of the volumes in a scratch subpool define this criterion for drive exclusion. An HSC or MVS/CSC TAPEREQ SUBPOOL parameter can specify a scratch subpool name for the request.

If no TAPEREQ SUBPOOL parameter is specified for this allocation, User Exit 02 can return a scratch subpool number or subpool name when it also sets the return code to use default processing or library drives. User Exit 10 can also apply to affinity chains here. Refer to “Affinity Separation” on page 40 for a discussion about the interaction between User Exit 02 and User Exit 10.



Note: When no specific subpool applies to the allocation, the default subpool, subpool 0, is used.

Drives that do not provide a recording technique compatible with a volume(s) in the subpool are excluded from consideration for this allocation.

For example:

```
//DD1 DD UNIT=CART,DSN=NEW.TRK36.DATA,DISP=(NEW,KEEP)
```

Assume HSC has scratches of media types 36TRACK and DD3 in a subpool named SUBPOOL3, and a TAPEREQ entry exists:

```
TAPEREQ DSN(NEW.*) SUBPOOL(SUBPOOL3)
```

Drives providing 36-track recording and helical recording remain in consideration for this allocation.

A User Exit 02 example for this criteria:

```
//DD1 DD UNIT=CART,DSN=SYS4.MYGROUP.DATA,DISP=(NEW,KEEP)
```

Assume the following:

- HSC User Exit 02 returns UX02LIB
- HSC User Exit 02 returns MYSUBPL in UX02SNAM
- MYSUBPL has been defined to the HSC using a SCRPOOL statement in the SLSSYSxx PARMLIB member or SCRPEDEF file
- HSC has scratch volumes defined in MYSUBPL scratch subpool.

The drives that are not in the library are excluded from consideration for this allocation during level 4 drive exclusion.

Exclusion Level 6

The generic location of available scratch volumes provides the next level of drive exclusion. If the library contains no scratch volumes, all library drives and virtual drives are excluded from selection, leaving only nonlibrary drives eligible. If the request is for a virtual scratch volume, eligible virtual drives are selected based on the management class from DFSMS or TAPEREQ; VTSSs that cannot support the requested management class migration are excluded. Lastly, if the library contains scratch volumes and the request is not for a virtual volume, then all virtual drives and all nonlibrary drives are excluded, leaving only library drives eligible.

For example:

```
//DD1 DD UNIT=CART,DSN=MY.NEW.DATASET,DISP=(NEW,KEEP)
```

Assume the HSC is active and has no nonlibrary drives defined, and the library contains no scratch volumes.

All HSC library drives are excluded from consideration for this allocation. When all drives are excluded during an exclusion level, and the level is not at or below minimum level, then the following occurs:

- messages SMC0045 and SMC0046 are issued
- all drives excluded at that level are restored to the list of eligible drives
- processing continues at the next level of exclusion.

In this case, all the HSC drives that were excluded by this criterion are now restored to the list. Processing continues at level 7.

Exclusion Level 7

The HSC ALLOC command and MVS/CSC ALTER command or startup parameter, ZEROSCR, determines whether or not to include drives in ACSs that do not contain any scratch cartridges of the required type. ZEROSCR(OFF), the default, indicates to include drives in all ACSs whether they contain scratch cartridges or not. ZEROSCR(ON) excludes drives in ACSs that do not contain any scratch volumes.

For example:

```
//DD1 DD UNIT=CART,DSN=MY.NEW.DATASET,DISP=(NEW,KEEP)
```

Assume that ZEROSCR(OFF) has been specified, and two ACSs have scratch counts as follows: ACS0 has 400 and ACS1 has 0.

Drives in ACS0 and ACS1 remain eligible for the allocation.

Another example is:

```
//DD2 DD UNIT=CART,DSN=MY.OTHER.DATASET,DISP=(NEW,KEEP)
```

Assume ZEROSCR(ON) has been specified, and three ACSs have scratch counts as follows: ACS0 has 400, ACS1 has 500, ACS2 has 0.

Drives in ACS0 and ACS1 remain eligible. ACS2 drives are excluded from consideration because ACS2 does not contain any scratch volumes.

Exclusion Level 8

Exclusion level 3 restricts the list of eligible drives to those compatible with the TAPEREQ or DFSMS requested media. Exclusion level 5 restricts the list of eligible drives to those compatible with the scratch media in the requested subpool. Exclusion level 8 may further restrict the devices for the request based on the TAPEREQ or DFSMS data class recording technique.

The HSC or MVS/CSC TAPEREQ definitions can explicitly specify a recording technique for the allocation request. DFSMS data class definitions can also denote an 18- or 36-track recording technique for the request and will override any TAPEREQ recording technique specification.

This criterion is the last to be applied to the drives that are eligible for the allocation. Any drive that does not provide the requested recording technique is excluded from consideration for the allocation.

For example:

```
//DD1 DD UNIT=CART,DISP=(NEW,KEEP),DSN=SYS4.TR4.DATA
```

Assume an HSC TAPEREQ has been defined:

```
TAPEREQ DSN(SYS4.TR4.** ) RECT(STK1RB)
```

Previous exclusion levels restricted the list of devices to all T9840 models. In this example, the TAPEREQ recording technique of STK1RB leaves only T9840B devices in consideration for this allocation.

Affinity Separation

Explicit unit affinity is an MVS facility that allows volumes associated with two separate JCL DD statements, or allocation requests, to be mounted serially on the same drive. A request for all generations of a GDG group (GDG ALL chain) can be considered as a GDGALL affinity.

The SMC makes no distinction between these two types of affinity. When processing an affinity chain begins, the drive exclusion process examines each allocation in the chain separately up to and including the minimum exclusion level. The chain is always separated when the minimum exclusion level processing results in lists of eligible drives, for two or more members of the chain, that do not contain common drives.

For example:

```
//DD1 DD UNIT=CART,DSN=MY.STK1R.DATASET,DISP=OLD
//DD2 DD UNIT=AFF=DD1,DSN=MY.LONGI.DATASET,DISP=OLD
```

DD1 specifies a data set on 9840 or T9840B media and DD2 specifies a data set on longitudinal media. Drive exclusion level 1 for specific volumes creates a list of eligible drives for each DD according to volume media required. The two lists do not contain a common drive. As a result, DD1 and DD2 no longer represent one drive allocation but two separate allocation requests. At this point, the SMC breaks the affinity chain between them.

Affinity Head-Of-Chain

For SMC affinity chain processing, the “head” of the affinity chain containing only scratch or only specific volumes is the first DD statement in the chain. If an affinity chain contains both scratch and specific volumes, the first specific volume is the “head.”

User Policy Influence on Affinity Separation

After the minimum level of drive exclusion and affinity separation completes, user policy influences the remaining affinity separation decisions.

Further affinity chain processing decisions are based on the first value found between the following:

- User Exit 10 return code separation decision
- ALLOCDef or ALLOCJob command SEPLvl parameter value.

User Exit 10

User Exit 10 sends an entire affinity chain to the user for examination. The user can decide whether the affinity chain remains together or separates during exclusion processing after the minimum level.

If User Exit 10 returns UX10SEP, indicating the chain members can be separated, the SMC proceeds as if ALLOCDEF SEPLVL=MAX were specified.

If User Exit 10 returns UX10LDRV, UX10NDRV, UX10ESUB, or UX10ASUB, the affinity chain remains together for the remainder of drive exclusion processing. The location requested (e.g., library drives if UX10LDRV is returned) replaces any location value provided by User Exits 02 or 08. This location information is used by drive exclusion level 4. Refer to Table 1 on page 29 and Table 2 on page 34 for drive exclusion level information.

For example:

```
//DD1 DD UNIT=CART,DSN=SYS4.DATASET1,VOL=SER=NOLIB1,DISP=OLD
//DD2 DD UNIT=AFF=DD1,DSN=SYS4.DATASET2,VOL=SER=INLIB2,DISP=OLD
```

Assume the following:

- DD1 requests a nonlibrary volume and DD2 requests a library volume.
- User Exit 08 returns a nonlibrary esoteric for DD1 (UX08ESUB).
- User Exit 10 returns a no separate decision and a library location (UX10LDRV).

The affinity chain stays together. Drive exclusion ignores the esoteric returned by User Exit 08 for DD1 and excludes all but library drives as requested by User Exit 10.

Any one of the four return values UX10LDRV, UX10NDRV, UX10ESUB, and UX10ASUB from User Exit 10 causes the SMC to operate as if ALLOCDEF SEPLvl=MIN were specified.



Note: Any User Exit 10 separation decision overrides the SEPLvl parameter of the ALLOCDef or ALLOCJob commands.

SEPLvl Parameter Value

If the decision to separate or keep the affinity chain together still has not been made, further affinity chain processing depends on the SEPLvl parameter value of the ALLOCDef and ALLOCJob commands (refer to Chapter 7, “Operator Commands,” for descriptions of these commands and setting an affinity separation level).

The SEPLvl parameter operates in three distinct ways depending on the value specified. The SEPLvl MAX parameter allows drive exclusion to be performed to the maximum level for each allocation element in the affinity chain. At the end of the drive exclusion process, if two or more elements of the chain no longer have common drives, the affinity chain is separated.

For example:

```
//DD1 DD UNIT=3490,VOL=SER=INACS0,DSN=MY.DSN1,DISP=OLD
//DD2 DD UNIT=AFF=DD1,VOL=SER=INACS1,DSN=MY.DSN2,DISP=OLD
```

Assume the following:

- Both volume INACS0 and INACS1 share common media characteristics.
- Volume INACS0 resides in ACS0, and volume INACS1 resides in ACS1.
- ALLOCDEF SEPLVL(MAX) has been specified.

During specific volume exclusion level 7, the list of eligible drives in DD1 are all located in ACS0 and the list of eligible drives in DD2 are all located in ACS1. No drives are common to both lists. This affinity chain separates.



Note: MAX is the default value for SEPLvl.

The SEPLvl parameter can be specified as MIN. The value MIN requests that any affinity chains that remain together after the minimum level of drive exclusion should not be separated during the remaining levels of drive exclusion processing.

Using the same example:

```
//DD1 DD UNIT=3490 ,VOL=SER=INACS0 ,DSN=MY.DSN1 ,DISP=OLD
//DD2 DD UNIT=AFF=DD1 ,VOL=SER=INACS1 ,DSN=MY.DSN2 ,DISP=OLD
```

Once again, assume:

- Both volume INACS0 and INACS1 share common media characteristics.
- Volume INACS0 resides in ACS0, and volume INACS1 resides in ACS1.
- ALLOCDEF SEPLVL(MIN) has been specified.

During specific volume exclusion level 7, the list of eligible drives for DD1 is all located in ACS0. DD1 is considered the “head” of the chain and exclusion is performed to ACS0. Only drives in ACS0 remain eligible for the allocation, and the affinity chain remains together.

The SEPLvl parameter can also be set to a specific drive exclusion level. The level number indicates the highest drive exclusion level at which separation can occur. Using the previous example, if the user specifies

```
ALLOCDEF SEPLVL=6
```

The chain remains together through drive exclusion level 6 because each level produces a drive list containing common drives for both DD1 and DD2.

Another example:

```
//DD1 DD UNIT=3490 ,VOL=SER=INACS0 ,DSN=VIRTUAL.DATASET1 ,DISP=OLD
//DD2 DD UNIT=AFF=DD1 ,VOL=SER=INACS1 ,DSN=VIRTUAL.DATASET2 ,DISP=OLD
//DD3 DD UNIT=AFF=DD1 ,VOL=SER=NL0001 ,DSN=REAL.DATASET ,DISP=OLD
```

Assume the following:

- ALLOCDEF SEPLVL=5 has been specified.
- Volume INACS0 is in ACS0 and VOLATTR MEDIA(ECART) RECTECH(36B) has been defined.
- Volume INACS1 is in ACS1 and VOLATTR MEDIA(ECART) RECTECH(36B) has been defined.
- Volume NL0001 is a nonlibrary volume and VOLATTR MEDIA(DD3A) has been defined.

At drive exclusion level 1, two chains result because of incompatible media (ECART and DD3A). Statements DD1 and DD2 are still chained together and continue through exclusion levels 2 through 5 because the exclusion criteria do not create disparate sets of eligible drives for the two DD statements.

No further separation of the DD1/DD2 chain can occur because the SEPLVL parameter has been set to level 5.

After the last drive exclusion level is completed for the job, the SMC examines the final drive lists. If the job cannot be allocated because not enough drives remain eligible, drive exclusion regresses to the previous exclusion level of the failing DD statement and a separated chain may be rejoined. This process repeats until the job can allocate.



Note: The SEPLvl parameter replaces the function of the HSC ALLOC and MVS/CSC ALTER command UNITAFF and GDGALL parameters. These parameters are no longer supported.

If your installation previously specified UNITAFF(NOSEP) and GDGALL(NOSEP), to preserve that behavior, specify a SEPLvl value of 3 on the ALLOCDef command. If your installation previously specified UNITAFF(SEP) and GDGALL(SEP), no SEPLVL parameter change is required, since the default SMC behavior separates affinity chains at the maximum separation level.

Refer to the ALLOCDef and ALLOCJob commands in Chapter 7, “Operator Commands,” for more information about setting an installation default minimum exclusion level and unit affinity separation level.

Drive Prioritization

After drive exclusion has occurred during the SSI24 common allocation event, the SMC then proceeds to assign a priority value to each remaining eligible drive during the corresponding SSI78 tape allocation event.



Note: Between a common allocation event (SSI24) for a tape and its tape allocation event (SSI78), MVS may make drives ineligible. When this occurs, the SMC retests the job step to ensure that it is allocatable with the remaining drives. If it is not, drives that were excluded by the SMC SSI24 process are reincluded until the job step is allocatable. Only the remaining drives are considered when the SMC assigns a priority value at SSI78 time.

One exception to the reinclusion process occurs when the ALLOCDef MIAcompat parameter is set to ON. In this case, exclusion is already done, so drive reinclusion cannot take place, and the job is allowed to fail.

The SMC drive priority is assigned based on the following criteria:

- For specific volumes, drives in LSMs closest to the volume are preferred.
- For scratch volumes, drives in LSMs with the largest number of scratch volumes are preferred.
- The TAPEREQ DEVTPREF parameter causes drives with specific recording techniques to be preferred.
- The TAPEREQ SCOPE parameter determines the relative priority of LSM location and drive type preferencing.

The following parameters, control statements, and user exits also influence the drive priority:

- HSC ALLOC command parameters LOWSCR, LSMPREF, and SCRTECH
- SLSUX02 (HSC format user exit 02).

After a final list of drives has been selected for allocation, the preference order of the eligible drives, after considering LSM and drive type preferencing, is selected based on a “last use” algorithm.

To reduce excessive wear on allocated drives, the SMC allocates drives by rotation based on the “last mount time” for each drive. This value is examined for every drive in the final drive list. The drive that had the most recent mount is located, and the drive immediately following it in the list is selected as the most preferred for the current allocation.



Note: This algorithm does not apply to virtual drives.

Deferring Mounts

The SMC can also set the deferred mount status if the installation chooses to defer the mount of library volumes.

The HSC ALLOC and MVS/CSC ALTER command DEFER parameter can set defaults for an installation. The following list describes the DEFER parameter settings and how the value determines if mounts are deferred. The HSC value is listed first followed by the corresponding MVS/CSC value.

Defer(OFF/NO)

The volume mount is deferred only if the user's JCL requests it. This is the default.

Defer(ON/YES)

The volume mount is deferred until the data set is opened. User Exit 09 can be used to override this value on an individual mount basis.

For optimal performance, Sun Microsystems recommends setting Defer(ON/YES).



Note: Virtual mounts are always deferred.

Library Subsystem Selection

In order to support the execution of SMC on a different host from the library subsystem, all HSC and MVS/CSC allocation and message processing user exits are now managed and invoked by the SMC. Therefore, only one version (HSC or MVS/CSC) of each user exit can be executed.

For this reason, it is recommended that you carefully review the contents of your existing user exits to determine the most effective method to direct allocations to the correct library. Specifically, the use of the user exit return code “use non library drives” to direct allocation to another library subsystem is no longer supported. Sun Microsystems recommends that you use TAPEREQ or user exit esoterics to direct allocations to the appropriate library subsystem.

An alternative method to select a library subsystem is to alter your scratch or specific allocation user exit to use the new HSC format, and explicitly select a LIBRARY name to direct a request.

See “SMC Library Subsystem Selection” on page 16 for information on how the SMC selects among multiple library subsystems to determine an “owner” for each allocation request (i.e. each DD in a job step may have a different library subsystem owner).

SMC DFSMS Processing

The SMC interface to DFSMS provides the following capabilities:

- esoteric substitution during MVS JCL interpretation
- at device allocation time, the DFSMS DATACLAS value can request a specific recording technique and/or media.
- at device allocation time, the DFSMS MGMTCLAS value can be used for VTV allocations.
- at volume mount, the DFSMS MGMTCLAS value can be assigned to a VTV.
- at volume mount, the DFSMS DATACLAS value can request a media for a scratch request.

Enabling/Disabling the SMC DFSMS Interface

To enable the SMC DFSMS interface, specify the SMS parameter of the ALLOCDef command as follows:

```
ALLOCDEF SMS=ON
```

To disable the SMC DFSMS interface, specify the SMS parameter of the ALLOCDef command as follows:

```
ALLOCDEF SMS=OFF
```

The SMC DFSMS interface can be selectively enabled or disabled by job or job step by specifying the SMS parameter of the ALLOCJob command.

Tailoring the SMC DFSMS Interface

The SMSDef command can be used to tailor the default SMC DFSMS support for your installation's requirements. This command provides the ability to include or bypass certain SMC DFSMS functions. See "SMSDef Command" on page 147 for more information about the options that can be specified.

DATACLAS, STORCLAS, and MGMTCLAS Specification

DFSMS DATACLAS can be specified by using the DATACLAS parameter on the DD JCL statement or by executing an Automatic Class Selection (ACS) routine. STORCLAS and MGMTCLAS can only be specified by ACS routines.



Note: STORCLAS and MGMTCLAS JCL parameters are not supported by the SMC DFSMS interface due to conflicts with IBM MVS DFSMS. Using the STORCLAS JCL parameter causes a data set to become DFSMS-managed, and the MGMTCLAS JCL parameter requires a DFSMS-managed data set. The data sets assigned STORCLAS and MGMTCLAS values in the STKTAP1 environment are not actually DFSMS-managed.

Invoking ACS Routines

IBM DFSMS invokes ACS routines with the variable `&ACSENVIR` set to `ALLOC` before the SMC invokes the ACS routines with variable `&ACSENVIR` set to `STKTAPI`.

The SMC invokes the ACS routines at the following points in processing:

- SSI55 Interpreter/Dynamic Allocation Exit (IDAX)
- SSI24 common allocation
- mount message interception.

The ACS routines are invoked in the following order:

1. data class
2. storage class
3. management class
4. storage group.

Management class and storage group ACS routines are called only if a storage class is assigned.

DFSMS Automatic Class Selection (ACS) Routine Environment for SMC

The following list of read-only variables is passed by SMC to DFSMS when the information is available to the SMC. Not all variables are available for every call to the ACS routines. See the descriptions of each DFSMS interface for exceptions.

- `&ACSENVIR` (equals `STKTAPI` for the SMC interface)
- `&ALLVOL`
- `&ANYVOL`
- `&DATACLAS`
- `&DD`
- `&DSORG`
- `&DSN`
- `&DSTYPE`
- `&EXPDT`
- `&FILENUM`
- `&JOB`
- `&LABEL`
- `&NVOL`
- `&PGM`
- `&RETPD`
- `&SYSNAME`
- `&SYSPLEX`
- `&UNIT`.

In the `STKTAPI` environment, the `&ANYVOL` variable is used only to match a specific `VOLSER` and does not contain the “`REF=xx`” values for `VOL=REF` allocations.

The `&DATACLAS` field is set when the `JCL DD` statement specifies this parameter.

If your installation uses the IGDACSXT routine to modify the read-only variables before the DFSMS ACS routine calls, be aware that the following variables, even if initialized, are not passed to the DFSMS ACS routines when &ACSENVIR is set to STKTAP1.

- &ACCT_JOB
- &ACCT_STEP
- &GROUP
- &MSGVP
- &USER
- &XMODE

Refer to the IBM manual *DFSMSdfp Storage Administration Reference* for more information about constraints when using read-only variables.

Esoteric Substitution During IDAX

When a job is in the JCL interpretation phase, the unit name (esoteric) can be changed indiscriminately (for example, you could substitute disk for tape). Considerations:

- Esoteric substitution applies to all newly created data sets. New allocations are signified by specifying DISP=NEW, or in some cases, DISP=MOD in the JCL (refer to the IBM *MVS JCL Reference* manual for a list of requirements when using DISP=MOD to specify a new data set).
- Unit affinity chains are always separated if ACS routines return different esoterics for different members of the chain.
- VOL=REF chains within a job are validated and updated with the head-of-chain esoteric if necessary to ensure that volume references are honored.

When the SMC locates a new allocation that is not DFSMS-managed, the SMC calls the DFSMS ACS routines with the environment variable &ACSENVIR set to STKTAP1. When the ACS routines return a pair of storage group names with the first name being STK1, the second storage group name replaces the original value of the UNIT parameter of the DD statement. For example, the following DD statement allocates a 3490 nonlibrary tape drive (esoteric TNLIB):

```
//DD1 DD DSN=&TEMP.DSN,UNIT=TNLIB,DISP=NEW
```

Assume that your installation wants to move all new tape allocations into a tape library. You can code the DFSMS ACS routines to return storage groups STK1 and a library esoteric such as TACS0, where TACS0 contains all drives in library location ACS0. After esoteric replacement, the only drives considered for the allocation are contained within the TACS0 esoteric. The nonlibrary drives are no longer considered eligible for the allocation.

Defining Storage Groups and Storage Classes

To enable esoteric substitution during IDAX, complete the following steps:

1. Define a storage group of type POOL named STK1.
2. Define a storage group of type POOL for each possible esoteric returned, giving it the same name as the esoteric name.
3. Define at least one volume to each storage group. Sun Microsystems recommends that you use a nonexistent volume for this definition.
4. Create a storage class ACS routine that, when it is invoked by the SMC (i.e., when the environmental variable is `&ACSENVIR=STKTAP1`), returns a storage class to be passed to the storage group ACS routine. Refer to Figure 8 on page 50.
5. Create a storage group ACS routine that, when the environmental variable is `&ACSENVIR=STKTAP1`, returns two storage groups (e.g., `&STORGRP='STK1', 'CART'`). In this case, “STK1” is the first storage group returned, and “CART,” the replacement esoteric, is the second. Refer to Figure 8.
6. Test for the `&ACSENVIR=STKTAP1` read-only variable in the storage class and storage group ACS routines. This enables you to prevent the storage class being assigned when MVS invokes the ACS routines. If MVS invokes the SMC version of the ACS routines, the data set becomes DFSMS-managed and all SMC allocation functions are bypassed.
7. Return both a storage class and a storage group during the SMC invocation of the DFSMS ACS routines. If a storage class is returned but a storage group is not, DFSMS issues a message stating that allocation has failed, but in reality that may not be the case.

```

PROC STORCLAS

FILTLIST LOCALDSN INCLUDE (BACKUP*.**,
                           PROD.BKP*.**)

FILTLIST RMTDSN INCLUDE (PROD.OFFSITE.***)

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
      WHEN &DSN = &LOCALDSN
        SET &STORCLAS = 'CART'
      WHEN &DSN = &RMTDSN
        SET &STORCLAS = 'RMT CART'
    END
  END

=====

PROC STORGRP

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
      WHEN &STORCLAS = 'CART'
        SET &STORGRP = 'STK1', 'CART'
      WHEN &STORCLAS = 'RMT CART'
        SET &STORGRP = 'STK1', 'RMT CART'
    END
  END

```

Figure 8. Sample Storage Class/Storage Group Routine

Availability of Read-only Variables

During DFSMS STORCLAS/STORGRP ACS routine processing, all of the read-only variables listed in “DFSMS Automatic Class Selection (ACS) Routine Environment for SMC” on page 47, except &DSORG, are available to the IDAX interface for esoteric substitution. The &DATACLAS field is set when the DD statement in the JCL specifies this parameter.

Validating DFSMS STORCLAS/STORGRP ACS Routine Execution

DFSMS provides the following ways to validate correct ACS routine execution:

- DFSMS WRITE statement in the ACS routines
- DFSMS ISMF test function.

When the SMC invokes the DFSMS ACS routines, all messages issued by DFSMS WRITE statements are routed to the GTF trace file if SMC tracing is active.

Refer to the publication *DFSMS/MVS DFSMSdfp Storage Administration Reference* for further information about writing and testing DFSMS ACS routines.

Retrieving DFSMS Values During Allocation and Volume Mount

Requesting Recording Technique/Media using DFSMS DATACLAS

During allocation processing, the media and/or recording technique can be specified for a DD allocation by coding:

- TAPEREQ statement
- DATACLAS parameter on the DD statement
- DATACLAS ACS routines.

During scratch volume mount processing, only the volume media can be specified.



Note: DATACLAS media and recording technique values override TAPEREQ values.

Media/Rectech Support

The following media and recording techniques are supported:

Media:	
MEDIA1	Standard capacity for both 18-track and 36-track media types (this is the default)
MEDIA2	Enhanced capacity 36-track media
Recording Techniques:	
18TRACK	18-track recording technique
36TRACK	36-track recording technique

Defining Data Classes

To use DATACLAS, customers must define their own data class and specify the recording technique and media. If the recording technique is left blank, the SMC defaults to allowing both 18TRACK and 36TRACK to remain eligible for the allocation request, unlike MVS, which defaults to 36TRACK.

As an example, a customer can define a DATACLAS named STD18 that specifies a recording technique of 18TRACK and standard media.

When using an ACS routine to supply a DATACLAS to set recording technique and media, do not specify `&ACSENVIR='STKTAP1'`. The DATACLAS ACS routine is invoked both by the SMC and MVS, and the resulting recording technique and media should be consistent.

The following figure provides a sample data class routine.

```
PROC DATACLAS
FILTLIST RMTDSN INCLUDE(PROD.OFFSITE.***)

  SELECT
  WHEN &DSN = &RMTDSN
    SET &DATACLAS = 'ECART36'
  END
```

Figure 9. Sample Data Class Routine

In the example above, assume DATACLAS ECART36 is defined with a recording technique of 36-track and a media type of MEDIA2. This data class routine example assigns ECART36 to data sets with names that start with “PROD.OFFSITE.”

By default, any DATACLAS value specified by the DFSMS ACS routine overrides the DATACLAS parameter specified on the JCL statement. Your installation can change this behavior by adding the following statements to your DFSMS DATACLAS ACS routine.

```
  WHEN (&DATACLAS NE ‘ ’)
  DO
    SET &DATACLAS = &DATACLAS
  EXIT
  END
```


Requesting Management Class Using DFSMS ACS Routines

During allocation and mount processing, a management class name can be specified for a virtual allocation by coding any of the following:

- TAPEREQ statement
- MGMTCLAS ACS routine.

The MGMTCLAS JCL parameter is not supported by the SMC DFSMS interface.



Note: The MGMTCLAS management class name overrides a TAPEREQ management class name.

When writing a management class routine, keep in mind:

- The management class routine is invoked only when a storage class is assigned.
- The management class routine must test for the &ACSENVIR='STKTAP1' read-only variable value.
- During volume mount message IEC501A interception, the &UNIT read-only variable contains the generic unit type, such as 3490. Therefore, careful consideration should be taken when coding ACS routines that use the &UNIT read-only variable.

Figure 10 on page 54 provides a sample management class routine.

```

PROC STORCLAS

FILTLIST CART INCLUDE ('CART')

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
      WHEN (&UNIT EQ &CART)
        SET &STORCLAS = 'VIRTAPE'
      WHEN (&UNIT NE &CART)
        SET &STORCLAS = 'STKDFLT'
    END
  END

=====

PROC STORGRP

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
      WHEN (&STORCLAS = 'VIRTAPE')
        SET &STORGRP = 'STK1', 'VDRIVES'
    END
  END

=====

PROC MGMTCLAS

FILTLIST LOCAL INCLUDE(BACKUP*.*,
                      PROD.BKP*.)

FILTLIST REMOTE INCLUDE(PROD.OFFSITE.*)

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
      WHEN (&DSN = &LOCAL)
        SET &MGMTCLAS = 'INVTAPE'
      WHEN (&DSN = &REMOTE)
        SET &MGMTCLAS = 'OFFVTAPE'
    END
  END

```

Figure 10. Sample Management Class Routine



Note: In this example, the STORCLAS ACS routine assigns a storage class to every call. This ensures that the MGMTCLAS routine is also driven at mount time. Only the VTCS component of NCS currently uses the management class value.

Availability of Read-only Variables

- At allocation time and at mount time processing of message IEF233A, the &UNIT read-only variable is set by the SMC to the value specified by the UNIT= parameter of the DD statement. At mount time processing of message IEC501A, the SMC sets the &UNIT read-only variable to the generic unit type, such as 3490.
- During dynamic allocation, the &DATACLAS value can be specified by coding the DADACL text unit. However, this value is not available to the SMC when processing mount message IEF233A.

Validating DFSMS ACS Routine Execution

DFSMS provides the following ways to validate correct ACS routine execution:

- DFSMS WRITE statement in the ACS routines
- DFSMS ISMF test function.

When the SMC invokes the DFSMS ACS routines, all messages issued by DFSMS WRITE statements or by DFSMS processing are routed to the SYSMSG data set of the job and are also routed to the GTF trace file if SMC tracing is active.

Refer to the publication *DFSMS/MVS DFSMSdfp Storage Administration Reference* for further information about writing and testing DFSMS ACS routines.

Interaction with Other Software

Automated Operations

Customers who use an automated operations product should review the following WTORs for possible auto-ops rule changes:

- SMC0110 - issued during SMC swap processing

Open Type J

The SMC allocation enhancements may not operate if you use the MVS Open Type J macro. Since that macro allows you to change volume serial numbers or data set names at open time, information available at Job Step Allocation time may be incorrect as interpreted by the SMC.



Note: Some vendor software products use MVS Open Type J. If you are experiencing unexpected allocation results using a vendor software product, check with the vendor to determine if Open Type J is used and follow the recommendations below.

SMC allocation may be influencing MVS allocation erroneously based on information that may have changed at open time. To prevent this problem when using the Open Type J macro, specify the appropriate esoteric in the JCL or specify an appropriate esoteric in an applicable TAPEREQ control statement.

SAMS: DISK (DMS)

Sterling Software's SAMS: DISK (DMS) has two methods of allocating transports:

- allocates transports at session startup, holds onto the transports throughout the session and uses Open Type J (refer to the section about "Open Type J").
- uses dynamic allocation (DYNALLOC) to allocate transports when required.

The SMC allocates correctly when dynamic allocation is used. Therefore, the latter method of allocating transports is recommended.

CA-MIA Tape Sharing

The Computer Associates Unicenter CA-MIA Tape Sharing for the z/OS and OS/390 product relies upon direct modification of the EDL at SSI24 time to determine what tape drives remain eligible for an allocation event. However, the SMC does not directly modify the EDL as part of its normal allocation processing. To enable proper coexistence with CA-MIA Tape Sharing, set the MIAcompat parameter of the ALLOCDef command to ON.

CA1-RTS Real Time Stacking

The Computer Associates Real Time Stacking product relies on DEFER processing being performed at SS124 time. However, the SMC normally performs DEFER processing during SS178 time. To enable proper coexistence with CA1-RTS, set the CA1rts parameter of the ALLOCDef command to ON.

MVS Security Packages

Ensure that your MVS security package (e.g. RACF, TopSecret) is configured to grant SMC the necessary authority to respond to MVS swap messages.

Chapter 5. SMC Allocation in a JES3 Environment

Overview

This chapter describes SMC allocation in a JES3 environment. The following topics are discussed:

- SMC allocation – JES3 not managing drives
- SMC allocation – JES3 managing drives
- DFSMS esoteric substitution
- JES3 initialization parameter considerations
- SMC normal operations
- JES3 constraints
- multiple library subsystem support
- SMC DFSMS processing
- GDG ALL processing
- interaction with other software



Note: See Appendix B for differences between SMC and NCS allocation.

SMC Allocation - JES3 Not Managing Drives

If JES3 is not managing any devices and SETUP=NONE has been specified on the JES3 STANDARDS initialization statement, the SMC operates the same as it does in a JES2 environment.

If JES3 is not managing any cartridge drives but is managing other types of devices, specify the J3NOSET parameter on the EXEC statement of the SMC START procedure (refer to “Creating the SMC START Procedure” on page 20). When J3NOSET is specified, the SMC operates the same as it does in a JES2 environment.

If either SETUP=NONE or J3NOSET is specified, no Type 1 modifications need to be installed on your JES3 system. HSC and MVS/CSC user exits invoked are the JES2 versions. For example, in HSC, the specific volume user exit SLSUX08 is invoked rather than user exit SLSUX13.

See Chapter 4, “SMC Allocation in a JES2 Environment” section for all further information about how the SMC operates in a JES2 environment.

SMC Allocation - JES3 Managing Drives

The SMC supports JES3-managed drives. JES3 manages drives through SETUP processing, which allocates drives identified on SETNAME statements when JOB, HWS (high watermark setup), or THWS (tape high watermark setup) is specified on the SETUP parameter of the JES3 STANDARDS initialization statement. In this environment, JES3 must manage all cartridge drives for the SMC to operate correctly.

SMC support operates during the following MVS subsystem interfaces (SSIs) and JES3 component phases:

- SSI55 Interpreter/Dynamic Allocation Exit (IDAX)
- JES3 Converter/Interpreter (C/I)
- SSI23 JES3 Dynamic Allocation
- JES3 Main Device Scheduler (MDS)
- SSI24 common allocation.

SSI55 Interpreter/Dynamic Allocation Exit (IDAX)

During MVS JCL interpretation processing, IDAX provides an opportunity to examine new data set allocations and to replace the unit parameter value with a different value (esoteric).



Note: This function is optional for SMC customers. For information about implementing DFSMS esoteric substitution, refer to “SMC DFSMS Processing” on page 98.

JES3 Converter/Interpreter (C/I)

During JES3 C/I subtask SWA processing, the SMC retrieves the media and recording technique obtained from the DFSMS DATACLAS specified by a DATACLAS JCL parameter on the DD allocation. This support is optional. For information about implementing DFSMS media and recording technique support, refer to “SMC DFSMS Processing” on page 98.

During JES3 C/I POSTSCAN processing, the SMC substitutes an esoteric to eliminate unacceptable drives from the allocation. The SMC performs the following processes to arrive at the best set of eligible drives:

- drive exclusion
- affinity separation
- esoteric unit name replacement.

At the end of JES3 C/I POSTSCAN processing, the SMC can defer the allocation until the job enters the initiator according to the customer’s library subsystem defer policy. Also, at this point of processing, fetch messages can be suppressed according to the customer’s library subsystem fetch policy.

These processes are discussed in this chapter.

SSI23 JES3 Dynamic Allocation

During SSI23 JES3 Dynamic Allocation processing, the SMC performs the same functions for dynamic allocations that the POSTSCAN C/I processes for common allocations:

- drive exclusion
- GDGALL affinity separation
- esoteric unit name replacement
- mount deferral.

JES3 Main Device Scheduler (MDS)

At the beginning of JES3 MDS processing, the SMC provides the ability to suppress fetch messages for dynamic allocation requests according to the customer’s library subsystem fetch policy.

During MDS device selection, the SMC sets preference values for drives according to their relative desirability, that is, JES3 selects the available drive with the highest preference value for the allocation.

SSI24 Common Allocation

If a mount has been deferred until the job enters an initiator, during SSI24 common allocation processing, the mount may be deferred further until the data set is opened. The customer’s library subsystem defer policy determines whether or not the mount is deferred.

Exceptions

The SMC does not influence the following types of cartridge tape allocation:

- demand allocation (i.e., request for a specific drive(s))



Notes: The SMC does perform DEFER processing for demand allocation.

- allocations excluded explicitly by entering the ALLOCJob command BYPASS parameter. Refer to the description of this command in Chapter 7, “Operator Commands”.
- allocations where the list of eligible devices contains only devices that are "unknown" (i.e. not virtual, not library, and not present in the NONLIB esoteric).
- DFSMS-managed allocation. An SMS-managed data set is defined as a data set that has a storage class defined. A storage class is assigned when either
 - the STORCLAS parameter is specified on the DD statement, or
 - an installation-written ACS routine selects a storage class for a new data set.

DFSMS Esoteric Substitution

During MVS JCL interpretation processing, the Interpreter/Dynamic Allocation Exit (IDAX) provides an opportunity to examine new data set allocations and to replace the unit parameter value with a different value (esoteric).



This function is optional for SMC customers. For information about implementing DFSMS esoteric substitution, refer to “SMC DFSMS Processing” on page 98.

Drive Exclusion

The JES3 C/I process creates an Intermediate Job Summary Table (IJS) that represents the device requirements for the JES3 managed devices for an entire job. The Type 1 modification to JES3 C/I module IATIIP1 enables the SMC to examine the IJS and to determine whether the SMC should influence the allocation.

When the SMC determines an interest in the allocation, its first step is to narrow down the list of eligible drives to those that best support the allocation request. This step is called *drive exclusion*.

Like the HSC and MVS/CSC device separation processes, the SMC drive exclusion process does **not** take into account the status of drives (e.g., offline, busy) when selecting drives that are eligible for the allocation request. If all drives compatible with the allocation request are unavailable, the job goes onto the allocation wait queue.

The SMC performs drive exclusion by building a list of drives defined to the original unit name and then by following a step-by-step process using an ordered list of exclusion criteria to remove drives from this list.

Before any drives are removed from consideration for the allocation, the SMC first determines whether a job can be allocated.

For example:

```
//JOB1 JOB ...
//STEP1 EXEC PGM=...
//DD1 DD UNIT=3490,...
//DD2 DD UNIT=3490,...
.
.
.
//DD25 DD UNIT=3490,...
```

Assume STEP1 is requesting twenty-five 3490 drives. If JES3 has configured only twenty-four 3490 drives, JES3 cannot allocate the required drives for the job. In this instance, two messages are issued:

- SMC0042, which indicates the job cannot allocate before SMC modification and allows the job to fail allocation.
- IAT4801, a JES3 message which indicates the job has been express-canceled by the interpreter Dynamic Support Program (DSP).

Once the SMC determines a job can be allocated, the SMC begins excluding drives based on an ordered list of criteria. Table 3 on page 64 and Table 4 on page 69 list these criteria and show the order the exclusion process follows. The lower the criteria level, the higher its importance. Thus, level 2 is considered more important than level 6.

Minimum Exclusion Level

By default, level 2 is the *minimum exclusion level*, that is, the desired minimum level of allocation exclusion, for both specific and scratch volume processing. However, if necessary, the ALLOCDef and ALLOCJob commands can reset the minimum exclusion level (refer to Chapter 7, “Operator Commands” for descriptions of these commands).



Note: If your VTCS configuration allows recalls of all migrated virtual volumes to any VTSS, Sun Microsystems recommends you set the minimum exclusion level to 1 using the ALLOCDef command. See “Exclusion Level 2” on page 65 for more information.

In the first step of drive exclusion, the SMC determines if the job can be allocated at the minimum drive exclusion level. If not enough drives remain eligible at the minimum exclusion level, the SMC sets a JES3 indicator that causes JES3 to express-cancel the job.

For example:

```
//JOB2 JOB ...
//STEP1 EXEC
//DD1 DD UNIT=3490,VOL=SER=984001,DISP=OLD
//DD2 DD UNIT=3490,VOL=SER=984002,DISP=OLD
//DD3 DD UNIT=3490,VOL=SER=984003,DISP=OLD
```

In this system environment, imagine that twenty 3490-type drives have been configured. Only two of those twenty are actually T9840 drives. If all three volumes require a separate T9840 tape drive, then not enough T9840 drives exist for the job to allocate. Without SMC influence, JES3 would allow the job to proceed through allocation. However, with SMC influence, the job fails and two messages are issued:

- SMC0043, which indicates the job could not allocate at the specified minimum exclusion level.
- IAT4801, a JES3 message which indicates the job has been express-canceled by the interpreter DSP.



Note: No exclusion criteria (including minimum level) are applied when the user specifies demand allocation (i.e., requests a specific drive or drives).

Subsequent Levels

Once the minimum level of exclusion has succeeded, the remaining exclusion levels are performed. Any level above the minimum level that fails is skipped and exclusion continues to the next level.



Note: A level fails if it results in too few drives for the DD statement to allocate. The only required exclusion levels are those at the minimum level or below.

After the last drive exclusion level is completed, the SMC examines the final drive lists. If the job cannot be allocated because not enough drives remain eligible, drive exclusion regresses to the previous exclusion level for the failing DD statement. This process repeats until the job can allocate.

Intentionally Failing a Job

Depending upon the characteristics of your installation, you may want specific jobs to fail at allocation time rather than execute them with non-optimal devices. As an example, if you have two ACSs, one of which is located at a remote site, you may prefer to fail the job, rather than having to transfer the correct volume from one ACS to another.

For example:

```
//DD1 DD UNIT=CART ,VOL=SER=INACS0 ,DISP=OLD  
//DD2 DD UNIT=AFF=DD1 ,VOL=SER=INACS1 ,DISP=OLD
```

Assume the following:

- Both volume INACS0 and INACS1 share common media characteristics.
- Volume INACS0 resides in ACS0, and volume INACS1 resides in ACS1.
- Only ACS0 contains a compatible drive.

If this affinity chain is not broken, then volume INACS1 must be ejected from ACS1 and entered into ACS0 for this job to execute, which may not be possible given the geographical locations of the ACSs.

In this case, failure at allocation time may be preferable to failure at run time. By specifying the correct MINLVL value (i.e., by issuing the ALLOCJob command with a MINLVL of 7 for this job), the job is failed at allocation time.



Caution: Setting the minimum exclusion level may cause jobs to fail unexpectedly. Whenever the SMC is unable to apply an exclusion criterion at or below the minimum level, the job fails allocation.

For example, if a TAPEREQ specifies a media of STANDARD for a scratch volume and the eligible device list contains only T9840 drives, when the SMC attempts to apply the “policy media” exclusion level, all drives are excluded. If the minimum level is set to 3 or higher, the job fails allocation. Therefore, Sun Microsystems recommends that the default exclusion level (as specified by the ALLOCDef command) should normally not be set to a level greater than 2.

Specific Volume Request Drive Exclusion Criteria

For a specific volume allocation, the SMC excludes drives in order from lowest exclusion level to highest based on the criteria in Table 3. The lower the level number, the more important the exclusion criteria.

Table 3. JES3 Drive Exclusion Levels (Specific Request)

Level	Specific Volume Criteria	Keyword*
1	For non-labeled (NL) specific volume requests, exclude all virtual drives. Exclude drives incompatible with the volume media. Primary source: external volume label Secondary source: VOLATTR MEDIA parameter The volume media can be obtained from the volume label or from an HSC VOLATTR statement MEDIA parameter.	VIRTUALLABEL MEDRECTECH
2**	For virtual volumes only, exclude drives not located in an accessible VTSS. Note: No further exclusion is performed for virtual volumes.	VTSSLOCATION
3	Exclude drives based on the required recording technique. Source: VOLATTR RECTECH parameter	VOLATTRRECTECH
4	Exclude drives based on user location policies. Primary source: TAPEREQ ESOTERIC parameter. Secondary source: User Exit 12 or 13 return codes.	USERPOLICY
5	Exclude drives based on the HSC ALLOC command SPECVOL parameter. Note: This exclusion level does not apply to MVS/CSC.	SPECVOL
6	Exclude drives based on volume location type, i.e., library or nonlibrary.	LOCTYPE
7	Exclude drives based on the ACS location of the volume, if the volume resides in the library.	ACSLOCATION
8	Exclude drives based on the requested recording technique. Primary source: DFSMS data class recording technique. Secondary source: TAPEREQ RECTECH parameter.	POLRECTECH

* Keywords associated with each exclusion level are specified in the exclusion criteria displayed in message SMC0046.

** Level 2 is the default minimum level.

Exclusion Level 1

The SMC excludes drives that are not compatible with the volume media. The volume media can be obtained from the volume label or from an HSC VOLATTR statement MEDIA parameter.

The SMC excludes all virtual drives for non-labeled (NL) specific volume requests.

Exclusion Level 2

If the specific volume requested is a virtual volume, the VTSS location for a resident volume must match the volume location. The VTSS location for a migrated virtual volume must match the location to which the volume can be recalled. The remaining exclusion levels do not apply to virtual volumes and are bypassed.



Note: If all VTSSs in your virtual configuration allow migration to common ACSs, you may want to set the minimum level exclusion level to 1 to allow jobs to proceed by migrating and recalling resident VTVs.

Refer to the VTCS documentation for more information about virtual volumes.

Exclusion Level 3

If an HSC VOLATTR statement specified the recording technique (RECTECH parameter) for this specific volume, the SMC excludes drives that do not provide that recording technique.

For example:

```
//DD1 DD UNIT=TLACS0,VOL=SER=VOL000,DISP=OLD
```

A VOLATTR statement has been defined to the HSC.

```
VOLATTR SERIAL(VOL000) RECTECH(36B) MEDIA(ECART)
```

VOL000 requires a 9490 drive, and if TLACS0 contains both 9490 and 9490EE drives, the 9490EE drives are excluded from consideration for this allocation.



Note: In MVS/CSC, the VOLATTR RECTECH is implied by the composite recording technique of the drives in the list returned by the library server.

Exclusion Level 4

If a TAPEREQ statement specifies an esoteric unit name for this allocation request, the SMC verifies the esoteric unit name is defined to JES3. If the esoteric passes this test, any drives in the original unit name that are not also defined to the esoteric unit name are excluded from consideration for this allocation.



Note: The SMC does not substitute the requested esoteric directly, but uses it to alter the list of eligible drives.

For example:

```
//DD2 DD UNIT=CART,VOL=SER=TR0001,DSN=SYS4.TR1.DATA,DISP=OLD
```

The following TAPEREQ statement is defined to HSC or MVS/CSC.

```
TAPEREQ DSN(SYS4.TR1.***) ESOT(TLACS1) RECT(LONGI)
```

The drives that are not defined to the esoteric TLACS1 are excluded from consideration for this allocation.

If no TAPEREQ statement specifies an esoteric for this allocation request, HSC and MVS/CSC User Exit 12 or 13 return codes and values can define the criteria for drive exclusion at this level. The user exit return codes can direct the SMC to select library drives, nonlibrary drives, or a specific ACS, or the user exit can return an esoteric name that contains customer-selected drives. Any drives in the current list of eligible devices that did not match the user exit return values are excluded here.

For example:

```
//DD1 DD UNIT=CART,VOL=SER=VOL001,DISP=OLD
```

HSC User Exit 13 returns UX13LIB, which requests a library drive. The drives that are not in the library are excluded from consideration for this allocation.

Exclusion Level 5

The SPECVOL parameter of the HSC ALLOC command instructs the SMC which library ACS locations can be used for this request if one of two circumstances exists:

- the specific volume is not in the library and no drives are defined in the nonlibrary esoteric to the HSC, or
- the specific volume is not in the library, and user exit 13 requested a library location.

Drives not in SPECVOL ACS locations are excluded from consideration.



Note: This exclusion level does not apply to MVS/CSC.

For example:

```
//DD1 DD UNIT=CART ,VOL=SER=NL0001 ,DISP=OLD
```

Volume NL0001 is not in the library and the HSC command ALLOC SPECVOL(00) has been issued.

If the NNLBDRV was not specified on the HSC LIBGEN SLILBRY macro meaning no drives exist outside the library, the drives not defined to ACS00 are excluded from consideration for this allocation.

Another example:

```
//DD2 DD UNIT=CART ,VOL=SER=NL0002 ,DISP=OLD
```

Assume volume NL0002 is not in the library, and the HSC command ALLOC SPECVOL(01) has been issued to allow ACS01 to be used for specific nonlibrary volumes.

If nonlibrary drives were defined on the HSC LIBGEN SLILBRY macro but HSC user exit 13 returned UX13LIB, all nonlibrary drives and drives in ACS00 are excluded from consideration for this allocation.

Exclusion Level 6

The generic location of the volume (library or nonlibrary) reduces the remaining list of eligible drives.



Note: For exclusion level 6, nonlibrary drives are outside the library and contain **either** known (from the HSC NNLBDRV or MVS/CSC NONLIB esoteric) or unknown device characteristics.

If the volume resides in a library, all drives outside the library are excluded, and if the volume resides outside the library, all library drives are excluded.

For example:

```
//DD1 DD UNIT=CART ,VOL=SER=LIB001 ,DISP=OLD
```

Assume volume LIB001 resides in a library.

All drives outside the library are excluded from consideration for this allocation.

Exclusion Level 7

For volumes residing in the library, the ACS location of the volume reduces the remaining list of eligible drives. Any drives that remain that do not reside in the same ACS as the volume are excluded.

For example:

```
//DD1 DD UNIT=CART,VOL=SER=A00VL1,DISP=OLD
```

Assume volume A00VL1 resides in ACS00, and the drives reside in ACS00 and ACS01.

The drives residing in ACS01 are excluded from consideration for this allocation.

Exclusion Level 8

Exclusion levels 1 and 3 restrict the list of eligible drives to those compatible with the volume's actual media and its recording technique, if specified on the HSC or MVS/CSC server VOLATTR statement. Exclusion level 8 may further restrict the drives for the request based on the TAPEREQ or DFSMS data class recording technique.

The HSC or MVS/CSC TAPEREQ definitions can explicitly denote a recording technique for the allocation request. DFSMS data class definitions can also specify an 18- or 36-track recording technique for the request and override any TAPEREQ recording technique specification.

For example:

```
//DD1 DD UNIT=CART,VOL=SER=TV9840,DISP=NEW,  
// DSN=SYS4.TR4.DATA
```

Assume the requested volume is a 9840 volume in the library with no recording technique specified on a VOLATTR. Assume an HSC TAPEREQ has been defined:

```
TAPEREQ DSN(SYS4.TR4.** ) RECT(STK1RB)
```

Previous exclusion levels restricted the list of devices to all T9840 models. In this example, the TAPEREQ recording technique of STK1RB leaves only T9840B devices in consideration for this allocation.

Scratch Volume Request Drive Exclusion Criteria

For a scratch volume allocation, the SMC excludes drives in order from lowest exclusion level to highest based on the criteria in Table 4. The lower the level number, the more important the exclusion criteria.

Table 4. JES3 Drive Exclusion Levels (Scratch Request)

Level	Scratch Volume Criteria	Keyword*
1	For nonlabeled (NL) scratch volume requests, exclude all virtual drives.	VIRTUALLABEL
2**	No level 2 for scratch volume exclusion.	
3	Exclude drives based on the requested media. Primary source: DFSMS data class media specification. Secondary source: TAPEREQ MEDIA parameter. Tertiary source: User Exit 04 request for virtual media.	POLMEDIA
4	Exclude drives based on user location policies. Primary source: TAPEREQ ESOTERIC parameter. Secondary source: User Exit 04 and 12 return codes.	USERPOLICY
5	Exclude drives based on the media of available scratch volumes in subpool. Primary source: TAPEREQ SUBPOOL parameter. Secondary source: User Exit 04 subpool values. Tertiary source: scratch subpool 0.	SUBPOOL
6	Exclude drives based on available scratch location type, i.e., library, nonlibrary, or virtual.	LOCTYPE
7	Exclude drives based on the HSC ALLOC or MVS/CSC ALTER command ZEROSCR parameter.	ZEROSCRATCH
8	Exclude drives based on the requested recording technique. Primary source: DFSMS data class recording technique. Secondary source: TAPEREQ RECTECH parameter.	POLRECTECH

* Keywords associated with each exclusion level are specified in the exclusion criteria displayed in message SMC0046.

** The MINLVL parameter value default, level 2 on the ALLOCDef or ALLOCJob command, applies to both scratch and specific volume requests. Even though level 2 has no meaning for scratch, level 2 is considered the default minimum level.

Exclusion Level 1

The SMC excludes virtual drives if the volume label type requested is NL (nonlabeled). This is the only required criterion. Refer to the VTCS documentation for more information about virtual volumes.

Exclusion Level 2

No level 2 scratch volume exclusion is performed.

Exclusion Level 3

Under HSC and MVS/CSC, a DFSMS data class definition can explicitly request the media desired, or the media can be derived from the DFSMS data class recording technique. Drives that do not provide the requested media support are excluded at this level.

If no DFSMS data class definition applies for this allocation, the HSC or MVS/CSC TAPERREQ definitions can explicitly specify the media desired for the allocation request, or media can be derived from TAPERREQ RECTECH. Drives that do not provide the requested media support are excluded at this level.

For example:

```
//DD1 DD UNIT=CART,DISP=(NEW,KEEP),DSN=SYS4.VIRT.DATA
```

No HSC TAPERREQ has been defined and no DFSMS data class definition applies. Assume HSC User Exit 04 return code requested virtual media (UX04VIRT). The HSC User Exit 04 media selected for this request is virtual. Only virtual drives remain in consideration for this allocation. The next example shows how TAPERREQ can override User Exit 04 return codes.

For example:

```
//DD1 DD UNIT=CART,DISP=(NEW,KEEP),DSN=SYS4.TR4.DATA
```

An HSC TAPERREQ has been defined:

```
TAPERREQ DSN(SYS4.TR4.** ) MEDIA(DD3A) RECT(DD3)
```

Assume HSC User Exit 04 return code requested virtual media (UX04VIRT).

If no DFSMS data class definition or TAPERREQ definition applies to this allocation request, User Exit 04 return code, UX04VIRT, can explicitly request virtual media. Non-virtual drives are excluded at this level.

However, the TAPERREQ media definition overrides the user exit return code, and the media selected for this request is helical. Only helical drives remain in consideration for this allocation. Messages SMC0045 and SMC0046 are issued indicating that the user exit request for virtual volumes was not honored.

Exclusion Level 4

If a TAPEREQ statement specifies an esoteric unit name for this allocation request, the SMC verifies the esoteric unit name is defined to JES3. If the esoteric passes this test, any drives in the original unit name that are not also defined to the esoteric unit name are excluded from consideration for this allocation.



Note: The SMC does not substitute the requested esoteric directly, but uses it to alter the list of eligible drives.

If no TAPEREQ statement specifies an esoteric for this allocation request, User Exit 04 and User Exit 12 return codes and values can define this criterion for drive exclusion. The user exit return codes can direct the SMC to select library drives, nonlibrary drives, or a specific ACS, or the user exit can return an esoteric unit name that contains customer-selected drives. Any drives in the current list of eligible drives that did not match the user exit return values are excluded here.

For example:

```
//DD1 DD UNIT=CART,DSN=&&TEMP1,DISP=(NEW,PASS)
```

Assume HSC User Exit 04 returns UX04LIB, which requests library drives.

The drives that are not in the library are excluded from consideration for this allocation.

A TAPEREQ example is:

```
//DD2 DD UNIT=CART,DSN=SYS4.TR1.DATA,DISP=(NEW,KEEP)
```

The following TAPEREQ statement is defined to the HSC or MVS/CSC.

```
TAPEREQ DSN(SYS4.TR1.**)  
ESOT(TLIB9840) RECT(LONGI)
```

The drives that are not defined to the esoteric TLIB9840 are excluded from consideration for this allocation. Any User Exit 04 return codes would be ignored.

Exclusion Level 5

The combined media and specified recording technique of the volumes in a scratch subpool define this criterion for drive exclusion. An HSC or MVS/CSC TAPEREQ SUBPOOL parameter can specify a scratch subpool name for the request.

If no TAPEREQ SUBPOOL parameter is specified for this allocation, User Exit 04 can return a scratch subpool number or subpool name when it also sets the return code to use default processing or library drives. User Exit 12 can also apply to affinity chains here. Refer to “Affinity Separation” on page 75 for a discussion about the interaction between User Exit 04 and User Exit 12.



Note: When no specific subpool applies to the allocation, the default subpool, subpool 0, is used.

Drives that do not provide a recording technique compatible with a volume(s) in the subpool are excluded from consideration for this allocation.

For example:

```
//DD1 DD UNIT=CART,DSN=NEW.TRK36.DATA,DISP=(NEW,KEEP)
```

Assume HSC has scratches of media types 36TRACK and DD3 in a subpool named SUBPOOL3, and a TAPEREQ entry exists:

```
TAPEREQ DSN(NEW.*) SUBPOOL(SUBPOOL3)
```

Drives providing 36-track recording and helical recording remain in consideration for this allocation.

A User Exit 04 example for this criteria:

```
//DD1 DD UNIT=CART,DSN=SYS4.MYGROUP.DATA,DISP=(NEW,KEEP)
```

Assume the following:

- HSC User Exit 04 returns UX04LIB
- HSC User Exit 04 returns MYSUBPL in UX04SNAM
- MYSUBPL has been defined to the HSC using a SCRPOOL statement in the SLSSYSxx PARMLIB member or SCRDEF file
- HSC has scratch volumes defined in MYSUBPL scratch subpool.

The drives that are not in the library are excluded from consideration for this allocation during level 4 drive exclusion.

Exclusion Level 6

The generic location of available scratch volumes provides the next level of drive exclusion. If the library contains no scratch volumes, all library drives and virtual drives are excluded from selection, leaving only nonlibrary drives eligible.

If the request is for a virtual scratch volume, eligible virtual drives are selected based on the management class from DFSMS or TAPEREQ; VTSSs that cannot support the requested management class migration are excluded. Last, if the library contains scratch volumes and virtual volumes were not requested, all virtual drives and all nonlibrary drives are excluded, leaving only library drives eligible.

For example:

```
//DD1 DD UNIT=CART,DSN=MY.NEW.DATASET,DISP=(NEW,KEEP)
```

Assume the HSC is active and has no nonlibrary drives defined, and the HSC library contains no scratch volumes.

All HSC library drives are excluded from consideration for this allocation. When all drives are excluded during an exclusion level, and the level is not at or below minimum level, then all drives excluded at that level are restored to the list of eligible drives and processing continues at the next level of exclusion.

In this case, all the HSC drives that were excluded by this criterion are now restored to the list. Processing continues at level 7.

Exclusion Level 7

The HSC ALLOC command and MVS/CSC ALTER command or startup parameter, ZEROSCR, is supported for JES3. ZEROSCR(OFF) provides drive exclusion found in earlier NCS releases. Drives residing in the ACS with the largest number of available scratch volumes remain eligible for selection. All other drives are excluded from selection. ZEROSCR(ON) excludes drives residing in ACSs that do not contain any scratch volumes.

For example:

```
//DD1 DD UNIT=CART,DSN=MY.NEW.DATASET,DISP=(NEW,KEEP)
```

Assume that ZEROSCR(OFF) has been specified, and two ACSs have scratch counts as follows: ACS0 has 400 and ACS1 has 500.

Only drives in ACS1 remain eligible for the allocation. Drives in ACS0 have been excluded.

Another example is:

```
//DD2 DD UNIT=CART,DSN=MY.OTHER.DATASET,DISP=(NEW,KEEP)
```

Assume ZEROSCR(ON) has been specified, and three ACSs have scratch counts as follows: ACS0 has 400, ACS1 has 500, ACS2 has 0.

Drives in ACS0 and ACS1 remain eligible. ACS2 drives are excluded from consideration because ACS2 does not contain any scratch volumes.

Exclusion Level 8

Exclusion level 3 restricts the list of eligible drives to those compatible with the TAPEREQ or DFSMS requested media. Exclusion level 5 restricts the list of eligible drives to those compatible with the scratch media in the requested subpool. Exclusion level 8 may further restrict the devices for the request based on the TAPEREQ or DFSMS data class recording technique.

The HSC or MVS/CSC TAPEREQ definitions can explicitly specify a recording technique for the allocation request. DFSMS data class definitions can also denote an 18- or 36-track recording technique for the request and will override any TAPEREQ recording technique specification.

This criterion is the last to be applied to the drives that are eligible for the allocation. Any drive that does not provide the requested recording technique is excluded from consideration for the allocation.

For example:

```
//DD1 DD UNIT=CART,DISP=(NEW,KEEP),DSN=SYS4.TR4.DATA
```

Assume an HSC TAPEREQ has been defined:

```
TAPEREQ DSN(SYS4.TR4.** ) RECT(STK1RB)
```

Previous exclusion levels restricted the list of devices to all T9840 models. In this example, the TAPEREQ recording technique STK1RB leaves only T9840B devices in consideration for this allocation.

Affinity Separation

Explicit unit affinity is an MVS facility that allows volumes associated with two separate JCL DD statements, or allocation requests, to be mounted serially on the same drive. A request for all generations of a GDG group (GDG ALL chain) can be considered as a GDGALL affinity.

Unlike HSC and MVS/CSC, SMC makes no distinction between these two types of affinity. When processing an affinity chain begins, the drive exclusion process examines each allocation in the chain separately up to and including the minimum exclusion level. The chain is always separated when the minimum exclusion level processing results in lists of eligible drives, for two or more members of the chain, that do not contain common drives.

For example:

```
//DD1 DD UNIT=CART,DSN=MY.STK1R.DATASET,DISP=OLD  
//DD2 DD UNIT=AFF=DD1,DSN=MY.LONGI.DATASET,DISP=OLD
```

DD1 specifies a data set on 9840 or T9840B media and DD2 specifies a data set on longitudinal media. Drive exclusion level 1 for specific volumes creates a list of eligible drives for each DD according to volume media required. The two lists do not contain a common drive. As a result, DD1 and DD2 no longer represent one drive allocation but two separate allocation requests. At this point, the SMC breaks the affinity chain between them.

Affinity Head-Of-Chain

For SMC affinity chain processing, the “head” of the affinity chain containing only scratch or only specific volumes is the first DD statement in the chain. If an affinity chain contains both scratch and specific volumes, the first specific volume is the “head.”

User Policy Influence on Affinity Separation

After the minimum level of drive exclusion and affinity separation completes, user policy influences the remaining affinity separation decisions.

Further affinity chain processing decisions are based on the first value found between the following:

- User Exit 12 return code separation decision
- ALLOCDef or ALLOCJob command SEPLvl parameter value.

User Exit 12

User Exit 12 sends an entire affinity chain to the user for examination. The user can decide whether the affinity chain remains together or separates during exclusion processing after the minimum level.

If User Exit 12 returns UX12SEP, indicating the chain members can be separated, the SMC proceeds as if ALLOCDEF SEPLVL=MAX were specified.

If User Exit 12 returns UX12LIB, UX12NLIB, UX12REPL, or UX12SACS, the affinity chain remains together for the remainder of drive exclusion processing. The location requested (e.g., library drives if UX12LIB is returned) replaces any location value provided by User Exits 04 or 13. This location information is used by drive exclusion level 4. Refer to Table 3 on page 64 and Table 4 on page 69 for drive exclusion level information.

For example:

```
//DD1 DD UNIT=CART,DSN=SYS4.DATASET1,VOL=SER=NOLIB1,DISP=OLD
//DD2 DD UNIT=AFF=DD1,DSN=SYS4.DATASET2,VOL=SER=INLIB2,DISP=OLD
```

Assume the following:

- DD1 requests a nonlibrary volume and DD2 requests a library volume.
- User Exit 13 returns a nonlibrary esoteric for DD1.
- User Exit 12 returns a no separate decision and a library location.

The affinity chain stays together. Drive exclusion ignores the esoteric returned by User Exit 13 for DD1 and excludes all but library drives requested by User Exit 12.

Any one of the four return values UX12LIB, UX12NLIB, UX12REPL, and UX12SACS from User Exit 12 causes the SMC to operate as if ALLOCDEF SEPLVL=MIN were specified.



Note: Any User Exit 12 separation decision overrides the SEPLVL parameter of the ALLOCDEF or ALLOCJOB commands.

SEPLVL Parameter Value

If the decision to separate or keep the affinity chain together still has not been made, further affinity chain processing depends on the SEPLVL parameter value of the ALLOCDEF and ALLOCJOB commands (refer to Chapter 7, “Operator Commands,” for descriptions of these commands and setting an affinity separation level).

The SEPLVL parameter operates in three distinct ways depending on the value specified. The SEPLVL MAX parameter allows drive exclusion to be performed to the maximum level for each allocation element in the affinity chain. At the end of the drive exclusion process, if two or more elements of the chain no longer have common drives, the affinity chain is separated.

For example:

```
//DD1 DD UNIT=3490,VOL=SER=INACS0,DSN=MY.DSN1,DISP=OLD
//DD2 DD UNIT=AFF=DD1,VOL=SER=INACS1,DSN=MY.DSN2,DISP=OLD
```

Assume the following:

- Both volume INACS0 and INACS1 share common media characteristics.
- Volume INACS0 resides in ACS0, and volume INACS1 resides in ACS1.
- ALLOCDEF SEPLVL(MAX) has been specified.

During specific volume exclusion level 7, the list of eligible drives in DD1 are all located in ACS0 and the list of eligible drives in DD2 are all located in ACS1. No drives are common to both lists. This affinity chain separates.



Note: MAX is the default value for SEPLVL.

The SEPLVL parameter can be specified as MIN. The value MIN requests that any affinity chains that remain together after the minimum level of drive exclusion should not be separated during the remaining levels of drive exclusion processing.

Using the same example:

```
//DD1 DD UNIT=3490 ,VOL=SER=INACS0 ,DSN=MY.DSN1 ,DISP=OLD
//DD2 DD UNIT=AFF=DD1 ,VOL=SER=INACS1 ,DSN=MY.DSN2 ,DISP=OLD
```

Once again, assume:

- Both volume INACS0 and INACS1 share common media characteristics.
- Volume INACS0 resides in ACS0, and volume INACS1 resides in ACS1.
- ALLOCDEF SEPLVL(MIN) has been specified.

During specific volume exclusion level 7, the list of eligible drives for DD1 is all located in ACS0. DD1 is considered the “head” of the chain and exclusion is performed to ACS0. Only drives in ACS0 remain eligible for the allocation, and the affinity chain remains together.

The SEPLVL parameter can also be set to a specific drive exclusion level. The level number indicates the highest drive exclusion level at which separation can occur. Using the previous example, if the user specifies

```
ALLOCDEF SEPLVL=6
```

The chain remains together through drive exclusion level 6 because each level produces a drive list containing common drives for both DD1 and DD2.

Another example:

```
//DD1 DD UNIT=3490 ,VOL=SER=INACS0 ,DSN=SYS4.DATASET1 ,DISP=OLD
//DD2 DD UNIT=AFF=DD1 ,VOL=SER=INACS1 ,DSN=SYS4.DATASET2 ,DISP=OLD
//DD3 DD UNIT=AFF=DD1 ,VOL=SER=NL0001 ,DSN=REAL.DATASET ,DISP=OLD
```

Assume the following:

- ALLOCDEF SEPLVL=5 has been specified.
- Volume INACS0 is in ACS0 and VOLATTR MEDIA(ECART) RECTECH(36B) has been defined.
- Volume INACS1 is in ACS1 and VOLATTR MEDIA(ECART) RECTECH(36B) has been defined.
- Volume NL0001 is a nonlibrary volume and VOLATTR MEDIA(DD3A) has been defined.

At drive exclusion level 1, two chains result because of incompatible media (ECART and DD3A). Statements DD1 and DD2 are still chained together and continue through exclusion levels 2 through 5 because the exclusion criteria do not create disparate sets of eligible drives for the two DD statements.

No further separation of the DD1/DD2 chain can occur because the SEPLVL parameter has been set to level 5.

After the last drive exclusion level is completed for the job, the SMC examines the final drive lists. If the job cannot be allocated because not enough drives remain eligible, drive exclusion regresses to the previous exclusion level of the failing DD statement and a separated chain may be rejoined. This process repeats until the job can allocate.



Note: The SEPLVL parameter replaces the function of the HSC ALLOC and MVS/CSC ALTER command UNITAFF and GDGALL parameters. These parameters are no longer supported.

If your installation previously specified UNITAFF(NOSEP) and GDGALL(NOSEP), to preserve that behavior, specify a SEPLVL value of 3 on the ALLOCDef command. If your installation previously specified UNITAFF(SEP) and GDGALL(SEP), no SEPLVL parameter change is required, since the default SMC behavior separates affinity chains at the maximum separation level.

Refer to the ALLOCDef and ALLOCJob commands in Chapter 7, “Operator Commands,” for more information about setting an installation default minimum exclusion level and unit affinity separation level.

Esoteric Unit Name Replacement

After drive exclusion and affinity separation successfully complete, each allocation may have a new list of eligible drives. The search begins to find an esoteric containing that exact list of drives. The SMC replaces the original JCL unit name in the Intermediate Job Summary Table (IJS) with this new esoteric.

The search for the “perfect” esoteric begins with the original JCL unit name or the unit name from the catalog entry for that data set. For example, assume the data set being allocated has been cataloged with the unit name 3490. Table 5 lists all the “3490” drives in the system.

Table 5. 3490 Drive List

ACS0	ACS1	Nonlibrary Location
0A10: 9490	0C10: 9490	0E10: 9490
0B10: 9840	0C11: 9490	0E11: 9490

JES3 groups devices by XTYPE names and groups XTYPE names by esoterics. The DEVICE statements coded in the JES3 initialization parameters follow:

```
DEVICE,TYPE=TA33490,XTYPE=(ACS09490,CA),JNAME=CA10,
JUNIT=(A10,MVS1,TAP,ON),XUNIT=(A10,MVS1,TAP,ON)

DEVICE,TYPE=TA33490,XTYPE=(ACS09840,CA),JNAME=CA11,
JUNIT=(B10,MVS1,TAP,ON),XUNIT=(B10,MVS1,TAP,ON)

DEVICE,TYPE=TA33490,XTYPE=(ACS19490,CA),JNAME=CC10,
JUNIT=(C10,MVS1,TAP,ON),XUNIT=(C10,MVS1,TAP,ON)

DEVICE,TYPE=TA33490,XTYPE=(ACS19490,CA),JNAME=CC11,
JUNIT=(C11,MVS1,TAP,ON),XUNIT=(C11,MVS1,TAP,ON)

DEVICE,TYPE=TA33490,XTYPE=(NLIB9490,CA),JNAME=CE10,
JUNIT=(E10,MVS1,TAP,ON),XUNIT=(E10,MVS1,TAP,ON)

DEVICE,TYPE=TA33490,XTYPE=(NLIB9490,CA),JNAME=CE11,
JUNIT=(E11,MVS1,TAP,ON),XUNIT=(E11,MVS1,TAP,ON)
```

Each unique location and device type pair has a unique XTYPE name. For example, the 9490 drive in ACS0 has a unique XTYPE name because it is the only 9490 in that location. The two nonlibrary 9490 drives share an XTYPE name because they are the same device type in the same location. An XTYPE should always define one type of device.

XTYPE names are associated with esoteric unit names in the initialization parameters as shown here:

```
SETNAME, XTYPE=ACS09490, NAMES=(CART, 3490, LIBDRVS, ACS0DRVS, A09490)
SETNAME, XTYPE=ACS09840, NAMES=(CART, 3490, LIBDRVS, ACS0DRVS, A09840)
SETNAME, XTYPE=ACS19490, NAMES=(CART, 3490, LIBDRVS, ACS1DRVS, A19490)
SETNAME, XTYPE=NLIB9490, NAMES=(CART, 3490, NLIBDRVS, NL9490)
```

Assume that during drive exclusion processing, the SMC determined the volume specified for this allocation resides in ACS0 and requires a 9490 drive. The drive exclusion process eliminates groups of drives by XTYPE.

In the environment defined above, the following XTYPE groups are no longer eligible for the allocation:

- ACS09840 – excluded at level 1 because T9840 drives are incompatible with the volume media.
- NLIB9490 – excluded at level 6 because the volume is in the library and these drives are not.
- ACS19490 – excluded at level 7 because the volume is in ACS0 and these drives are in ACS1.

One XTYPE, ACS09490, remains eligible for allocation at the end of drive exclusion. SMC esoteric unit name replacement now searches the SETNAME definitions for an esoteric that only contains the XTYPE ACS09490. For this allocation, the SMC selects the esoteric A09490 because it contains only XTYPE ACS09490. The A09490 esoteric replaces the original unit name, 3490, in the Intermediate Job Summary (IJS) table for that job.

If the example required two drives for the allocation (e.g., UNIT=(3490,2)) and the first volume to be mounted resides in ACS0, the results of drive exclusion would be as follows:

- ACS09840 – excluded at level 1 because T9840 drives are incompatible with the volume media.
- NLIB9490 – excluded at level 6 because the volumes are in the library and these drives are not.
- Exclusion level 7 fails.

On entry to level 7, three drives remain, two drives defined to XTYPE ACS19490 and the other drive to XTYPE ACS09490. If XTYPE ACS19490 were excluded because of ACS location, only one drive would remain. This allocation requires two drives. Thus, exclusion level 7 does not exclude the drives in ACS1.

Two XTYPEs, ACS09490 and ACS19490, remain eligible for allocation at the end of drive exclusion. The SMC esoteric unit name replacement now determines that XTYPE ACS09490 cannot be used for the allocation.

IBM APAR OW38427 to JES3 introduced the restriction that multi-unit allocations use devices defined in the same XTYPEs. Since XTYPE ACS09490 only contains one drive, it cannot satisfy the allocation requirements. The SMC esoteric unit name replacement now searches the SETNAME definitions for an esoteric that only contains the XTYPE ACS19490. The A19490 esoteric replaces the original unit name, 3490, in the IJS for that job.

After the SMC has updated the IJS, JES3 C/I processing continues. JES3 creates a Job Summary Table (JST) from the IJS table and performs any high watermark setup (HWS) chaining. During HWS chaining, JES3 can also change the esoteric unit name in the JST after the SMC changes the esoteric. The HWSNAME initialization statements define which esoteric unit names are subsets of other esoteric unit names. This allows JES3 to reuse devices in following steps.

Refer to “JES3 Initialization Parameter Considerations” on page 85 for more information about setting up your installation’s JES3 DEVICE, SETNAME and HWSNAME statements.

Suppressing Fetch Messages

By the time JES3 C/I processing completes, the IJS becomes the JST that represents the job for the remainder of its existence. The JST reflects the esoteric substitutions made by the SMC and by JES3. The next stage for the job is the Main Device Scheduler (MDS).

At the beginning of MDS processing, JES3 begins preparing the job for allocation. Asking the operator to fetch volumes is an optional phase in MDS. JES3 issues a fetch message when a job requires a volume that is not currently mounted and the SETPARAM statement FETCH parameter is set to YES (the default). If the SETPARAM statement also specifies ALLOCATE=MANUAL, jobs are placed on the volume wait queue until the operator retrieves the volume(s) and issues the *START SETUP command.

A customer's installation may not want to receive fetch messages for volumes in the library. To do so for common allocation requests (JCL statement allocation), install the SMC version of the JES3 user exit IATUX09. For dynamic allocation requests, install the SMC Type-1 modification to IATMDFE.

If you are running HSC, issue the ALLOC FETCH(OFF) or ALLOC FETCH(NONLIB) command. FETCH(OFF) is the default and suppresses fetch messages for any volume that is to be mounted on a library drive. If fetch messages are desired for nonlibrary volumes that are to be mounted on a library drive, FETCH(NONLIB) should be entered.



Note: FETCH(NONLIB) causes another volume lookup request to the HSC, which can affect performance.

If you are running MVS/CSC, issue the ALTER FETCH(NO) or ALTER FETCH(NONLIB) command. FETCH(NO) is the default and suppresses fetch messages for any volume that is to be mounted on a library drive. If fetch messages are desired for nonlibrary volumes that are to be mounted on a library drive, FETCH(NONLIB) should be entered.



Note: FETCH(NONLIB) causes another volume lookup request to the MVS/CSC, which can affect performance.

If your system is running with ALLOCATE=MANUAL as described above, when fetch messages are suppressed for a volume allocation, that allocation does not go onto the volume wait queue.

If your system is running with the SETPARAM statement set to FETCH=NO, or if you prefer to receive fetch messages for all volumes, then the IATMDFE Type-1 modification does not need to be applied to your system. The IATUX09 user exit also performs other functions and should be applied.

Drive Prioritization

The next step in Main Device Scheduler (MDS) allocates the devices required for the job. The SMC Type-1 modification to IATMDAL provides the SMC with the ability to review the list of drives available for each tape allocation. The list of drives contains drives that are online and available and are members of the group of drives defined in the esoteric placed in the Job Summary Table (JST) after drive exclusion processing.

The SMC drive priority is assigned based on the following criteria:

- For specific volumes, drives in LSMs closest to the volume are preferred.
- For scratch volumes, drives in LSMs with the largest number of scratch volumes are preferred.
- The TAPEREQ DEVTPREF parameter causes drives with specific recording techniques to be preferred.
- The TAPEREQ SCOPE parameter determines the relative priority of LSM location and drive type preferencing.

The following parameters, control statements, and user exits also influence the drive priority:

- HSC ALLOC command parameters LOWSCR, LSMPREF, and SCRTECH
- SLSUX04 (HSC format user exit 04).

After a final list of drives has been selected for allocation, the preference order of the eligible drives, after considering LSM and drive type preferencing, is selected based on a “last use” algorithm.

To reduce excessive wear on allocated drives, the SMC allocates drives by rotation based on the “last mount time” for each drive. This value is examined for every drive in the final drive list. The drive that had the most recent mount is located, and the drive immediately following it in the list is selected as the most preferred for the current allocation.



Note: This algorithm does not apply to virtual drives.

Deferring Mounts

After drive prioritization completes and all drives are allocated, the volumes required by the job, but not currently mounted, can be mounted before the job enters an initiator, during common allocation in the initiator, or at data set open time. Your installation may choose to defer the mount of library volumes.

The IATUX09 JES3 user exit modification must be installed to defer mounts to the initiator or to data set open time. This SMC modification interacts with the setting of the DEFER parameter of the HSC ALLOC command or the MVS/CSC ALTER command.

The following list describes the DEFER parameter settings and how the value determines what is deferred and when. The HSC value is listed first followed by the corresponding MVS/CSC value.

Defer(OFF/NO)

The volume mount is deferred only if the user's JCL requests it. This is the default.

Defer(JEs3)

The volume mount is deferred until the job enters the initiator. HSC SLSUX11 or MVS/CSC SCSUX11 can be used to override this value on an individual mount basis. For instance, the user exit can return a value to allow the mount to proceed before the job enters the initiator. JEs3 is the default value.

Defer(ON/YES)

The volume mount is deferred until the job enters the initiator and then again until the data set is opened. HSC SLSUX09 or MVS/CSC SCSUX09 can be used to override this value on an individual mount basis.

For optimal performance, Sun Microsystems recommends setting Defer(ON/YES).



Note: Virtual mounts are always deferred.

JES3 Initialization Parameter Considerations

The library and nonlibrary drive environment must be defined to JES3 in the initialization deck using the following parameter statements:

- DEVICE statements to define drive addresses, device types, and XTYPEs
- SETNAME statements to define esoteric names and to associate them with XTYPEs
- HWSNAME statements to define the esoteric name relationships used during HWS processing.

This section describes these statements and shows how to code them for a sample configuration. Table 6 consists of the following drive addresses and esoterics attached to two systems, MVS1 and MVS2.

Table 6. Sample Configuration

Nonlibrary	ACS0	ACS1	Virtual
120-127 3480	220-223 4490	320-327 9490	A20-A5F Virtual
140-143 3490	240-243 9490	440-447 9490	
180-189 9840	280-289 9840	460-461 9940	



Note: The drive addresses and esoterics in this example are not meant to be taken literally but are intended to show how a wide variety of device types can be defined using JES3. Exact JES3 initialization statements are configuration dependent.

JES3 DEVICE Initialization Statements

DEVICE statements define the drives that JES3 can use to satisfy allocation requests. These statements define:

- drive addresses
- JES3/MVS systems that can access the drives
- initial drive online status
- device type of the drive.

The XTYPE parameter is especially important to SMC allocation. XTYPE connects devices with the same XTYPE value to a group of esoteric unit names. For example:

```
DEVICE,XTYPE=(DEV0220,CA),XUNIT=(220,MVS1,TAP,ON,220,MVS2,TAP,ON),
      NUMDEV=4,...
SETNAME,XTYPE=DEV0220,NAME=(CART,3490,SYS3480R,SYS348XR,ACS0TAP,...)
```

Devices 220-223 in ACS0 in the “Sample Configuration” on page 85 have been associated with the XTYPE name DEV0220. This name allows JES3 to allocate a device from the group 200-223 when any of the esoteric unit names listed on the SETNAME statement associated with XTYPE DEV0220 are specified in JCL or in a catalog entry.

The SMC relies on each XTYPE group to be unique with regard to real drive type and location. In the list of drives for ACS0, the 4490 drives should not be defined with the same XTYPE as the 9490 drives. Also, the T9840 drives located in ACS0 should not be defined with the same XTYPE as the nonlibrary T9840 drives.

During SMC initialization, XTYPE groupings are examined to verify these XTYPE restrictions. If an XTYPE contains mixed devices types or mixed locations, the characteristics of the first drive in the XTYPE group defines the remaining drives.

The SMC configuration report utility shows XTYPE, esoteric, and drive information. Refer to Chapter 9, “SMC Utilities” for more information about the configuration report.

The following example shows how DEVICE statements can be coded for this sample configuration.



Note: Drives must be defined to MVS prior to defining them to JES3. Use the Hardware Configuration Definition (HCD) facility to assign MVS unit addresses to the devices in the I/O Configuration.

```

* NONLIBRARY DRIVES
* 3480 DRIVES 120-127
DEVICE, XTYPE=(DEV0120, CA), JNAME=TNL, DTYPE=TA03480, NUMDEV=8,
JUNIT=(0120, MVS1, TAP, ON, 0120, MVS2, TAP, ON),
XUNIT=(0120, MVS1, TAP, ON, 0120, MVS2, TAP, ON)
*
* 3490 DRIVES 140-143
DEVICE, XTYPE=(DEV0140, CA), JNAME=TNL, DTYPE=TA03490, NUMDEV=4,
JUNIT=(0140, MVS1, TAP, ON, 0140, MVS2, TAP, ON),
XUNIT=(0140, MVS1, TAP, ON, 0140, MVS2, TAP, ON)
*
* 9840 DRIVES 180-189 DEFINED AS 3490'S
DEVICE, XTYPE=(DEV0180, CA), JNAME=TNL, DTYPE=TA03490, NUMDEV=10,
JUNIT=(0180, MVS1, TAP, ON, 0180, MVS2, TAP, ON),
XUNIT=(0180, MVS1, TAP, ON, 0180, MVS2, TAP, ON)
*
* DRIVES IN ACS0
* 4490 DRIVES 220-223
DEVICE, XTYPE=(DEV0220, CA), JNAME=ACS0, DTYPE=TA03490, NUMDEV=4,
JUNIT=(0220, MVS1, TAP, ON, 0220, MVS2, TAP, ON),
XUNIT=(0220, MVS1, TAP, ON, 0220, MVS2, TAP, ON)
*
* 9490 DRIVES 240-243
DEVICE, XTYPE=(DEV0240, CA), JNAME=ACS0, DTYPE=TA03490, NUMDEV=4,
JUNIT=(0240, MVS1, TAP, ON, 0240, MVS2, TAP, ON),
XUNIT=(0240, MVS1, TAP, ON, 0240, MVS2, TAP, ON)
*
* 9840 DRIVES 280-289 DEFINED AS 3590'S
DEVICE, XTYPE=(DEV0280, CA), JNAME=ACS0, DTYPE=TA435901, NUMDEV=10,
JUNIT=(0280, MVS1, TAP, ON, 0280, MVS2, TAP, ON),
XUNIT=(0280, MVS1, TAP, ON, 0280, MVS2, TAP, ON)
*
* DRIVES IN ACS1
* 9490 DRIVES 320-327
DEVICE, XTYPE=(ACS19490, CA), JNAME=ACS1, DTYPE=TA03490, NUMDEV=8,
JUNIT=(0320, MVS1, TAP, ON, 0320, MVS2, TAP, ON),
XUNIT=(0320, MVS1, TAP, ON, 0320, MVS2, TAP, ON)
*
*9490 DRIVES 440-447
DEVICE, XTYPE=(ACS19490, CA), JNAME=ACS1, DTYPE=TA03490, NUMDEV=8,
JUNIT=(0440, MVS1, TAP, ON, 0440, MVS2, TAP, ON),
XUNIT=(0440, MVS1, TAP, ON, 0440, MVS2, TAP, ON)
*
* 9940 DRIVES DEFINED AS 3590'S
DEVICE, XTYPE=(DEV0460, CA), JNAME=ACS1, DTYPE=TA435901, NUMDEV=2,
JUNIT=(0460, MVS1, TAP, ON, 0460, MVS2, TAP, ON),
XUNIT=(0460, MVS1, TAP, ON, 0460, MVS2, TAP, ON)
*
* VIRTUAL DRIVES
DEVICE, XTYPE=(DEV0A20, CA), JNAME=VIRT, DTYPE=TA03490, NUMDEV=64,
JUNIT=(0A20, MVS1, TAP, ON, 0A20, MVS2, TAP, ON),
XUNIT=(0A20, MVS2, TAP, ON, 0A20, MVS2, TAP, ON)

```

JES3 SETNAME Initialization Statements

The SETNAME statements define all esoteric unit names and device type names associated with JES3-managed devices. These esoteric unit names and device type names can be specified by the UNIT parameter on a DD statement or as the unit type in a cataloged data set entry.

DEVICE statements associate a set of drives with an XTYPE. The SETNAME statement associates the XTYPE with a group of esoteric unit names.

During SMC esoteric unit name replacement, the relationships among the devices, the XTYPEs, and the esoteric unit names enable the SMC to choose the optimal esoteric unit name.



Note: During allocation of specific volumes, the SMC attempts to substitute an esoteric containing only drives compatible with the volume. If all esoterics that are a subset of the original esoteric contain some drives not compatible with the volume (except for drives defined as MODEL=IGNORE), the SMC issues message SMC0068 and does not substitute for the original esoteric.

Therefore, to ensure the SMC's ability to perform esoteric substitution, you must define at least one esoteric containing only drives that are compatible with each media type in your library. For example, if your library contains ECART and standard volumes and 9490, 4490 and 4480 drives, you must, at a minimum, define one esoteric containing only drives compatible with the ECART volumes (9490, 4490, and 4480 drives). You can also define other esoterics containing any desired combinations of these drive types.

For best SMC performance, each drive type in each location should have a unique esoteric defined. For example, an esoteric named A09840 could be defined to contain only the T9840 drives located in ACS0.

The following example shows how SETNAME statements can be coded for this configuration. The esoteric unit names specified in the NAMES parameter value list consist of the following:

- CART – all cartridge drives in the environment
- NLCART – all cartridge drives not in a library ACS
- A0CART – all cartridge drives in ACS0
- A1CART – all cartridge drives in ACS1
- ALLxxxx – all cartridge drives of the same device type, xxxx, independent of location
- LIBxxxx – all cartridge drives of the same device type, xxxx, in any library location
- yyxxxx – all cartridge drives of the same device type, xxxx, in location yy.

The generic device type names, such as 3480 or SYS3480R, are also specified in the NAMES lists.

```

* 3480/NONLIBRARY
SETNAME, XTYPE=DEV120, NAMES=(SYS3480R, CART, 3480, NLCART, NL3480)
*
* 3490/NONLIBRARY
SETNAME, XTYPE=DEV0140, NAMES=(SYS3480R, SYS348XR, CART, 3490, NLCART,
                                ALL3490, NL3490)
*
* 9840/NONLIBRARY
SETNAME, XTYPE=DEV0180, NAMES=(SYS3480R, SYS348XR, CART, 3490, NLCART,
                                ALL9840, NL9840)
*
* 4490/ACS0
SETNAME, XTYPE=DEV0220, NAMES=(SYS3480R, SYS348XR, CART, 3490, A0CART,
                                A04490, A0DEVT90)
*
* 9490/ACS0
SETNAME, XTYPE=DEV0240, NAMES=(SYS3480R, SYS348XR, CART, 3490, A0CART,
                                ALL9490, LIB9490, A09490, A0DEVT90)
*
* 9840/ACS0
SETNAME, XTYPE=DEV0280, NAMES=(CART, 3590-1, A0CART, ALL9840, A09840)
*
* 9490/ACS1
SETNAME, XTYPE=ACS19490, NAMES=(SYS3480R, SYS348XR, CART, 3490, A1CART,
                                ALL9490, LIB9490, A19490)
*
* 9940/ACS1
SETNAME, XTYPE=DEV0460, NAMES=(CART, 3590-1, A1CART, ALL9940, A19940)
*
* VIRTUAL DRIVES
SETNAME, XTYPE=DEV0A20, NAMES=(SYS3480R, SYS348XR, CART, 3490, NLCART,
                                VIRT CART)

```

Refer to the appropriate version of the IBM *JES3 Initialization and Tuning Reference* for more information about esoteric unit name values for the SETNAME statement NAMES parameter.

JES3 HWSNAME Initialization Statements

The HWSNAME statements define which esoteric unit names are subsets of other esoteric unit names. Used during JES3 high watermark setup (HWS), these statements determine if a device can be reused from step to step.

The first HWSNAME TYPE parameter specifies the esoteric unit name, known as the *major name*, used during HWS processing. The following esoteric unit names, called *minor names*, can be used as an alternate to the major name.

The order of the minor names listed in the HWSNAME statement is the order in which they can be substituted for the major name. For example:

```
HWSNAME TYPE=(3490,ALL4490,ALL9490,ALL3490)
```

and

```
//STEP1 EXEC PGM...
//DD1 DD UNIT=3490,...
//STEP2 EXEC PGM...
//DD1 DD UNIT=ALL3490,...
//DD2 DD UNIT=ALL4490,...
```

JES3 HWS processing allocates two drives for this job. The Job Summary Table (JST) for the job after HWS shows the following esoterics for each DD allocation request:

- STEP1 DD1 and STEP2 DD2 JST entries contain ALL4490 because ALL4490 appears in the minor name list before ALL3490.
- STEP2 DD1 JST entry contains ALL3490.

Another example shows how HWS names are used when allocating across step boundaries:

```
//STEP1 EXEC PGM...
//DD1 DD UNIT=ALL9490,...
//DD2 DD UNIT=ALL4490,...
//STEP2 EXEC PGM...
//DD1 DD UNIT=3490
```

JES3 HWS begins with DD1 of STEP1 looking for an allocation in STEP2 that can use the same device. DD1 of STEP2 specifies 3490. The HWSNAME above for major name 3490 indicates that ALL9490 is an alternate (or minor) name for 3490. Therefore, STEP1 DD1 and STEP2 DD1 allocate the same drive. The JST entry for DD1 of STEP2 is not updated to reflect a new esoteric. The drive allocated for STEP1 DD2 is freed at the end of STEP1.

The minor names should not contain any devices that are not defined to the major name. For example:

```
HWSNAME TYPE=(A0CART,ALL9840,...)
```

Assume the following:

- A0CART contains drives 220-223, 240-243, and 280-289.
- ALL9840 contains drives 180-189 and 280-289.

ALL9840 contains drives (180-189) not in A0CART. In this case, volumes inside the library requesting a T9840 drive may attempt to allocate to a drive outside the library after HWS processing by JES3.

HWS processing occurs after SMC esoteric unit name replacement. Therefore, the HWSNAME definitions can affect the final allocation decision if JES3 also changes the esoteric unit name as in the first example.

The best solution for this situation is to create unique esoteric unit names (by location and device type) so that the SMC can select an esoteric unit name that has no minor name. See the HWSNAME entries in the following example that have been coded for the sample configuration.

```

* GENERIC MAJOR NAMES
HWSNAME TYPE=(SYS3480R)
HWSNAME TYPE=(SYS348XR)
HWSNAME TYPE=(3480,NL3480)
HWSNAME TYPE=(3490,SYS348XR,
                ALL3490,ALL9490,LIB9490,A0DEVT90,
                A04490,A09490,A19490,NL3490,NL9840)
HWSNAME TYPE=(3590-1, ALL9940,
                A09840,A19940)
*
* ALL DRIVES IN THE COMPLEX
HWSNAME TYPE=(CART,SYS3480R,SYS348XR,3490,3480,3590-1,
                ALL3490,ALL9840,ALL9490,ALL9940,LIB9490,
                A0CART,A1CART,NLCART,A0DEVT90,
                A04490,A09490,A09840,A19490,A19940,
                NL3480,NL3490,NL9840)
*
* DRIVES BY DEVICE TYPE
HWSNAME TYPE=(ALL3490,LIB9490,A0DEVT90,A09490,A19490,NL3490)
HWSNAME TYPE=(ALL9840,A09840,NL9840)
HWSNAME TYPE=(ALL9490,LIB9490,A09490,A19490)
HWSNAME TYPE=(ALL9940,A19940)
*
* DRIVES BY LOCATION
HWSNAME TYPE=(LIB9490,A09490,A19490)
HWSNAME TYPE=(NLCART,ALL3490,ALL3480,3480,
                NL3480,NL3490,NL9840)
HWSNAME TYPE=(A0CART,A04490,A09490,A09840,A0DEVT90)
HWSNAME TYPE=(A1CART,ALL9940,A19940,A19490)
*
* DRIVES BY LOCATION AND DEVICE TYPE
HWSNAME TYPE=(A0DEVT90,A04490,A09490)
HWSNAME TYPE=(NL3480)
HWSNAME TYPE=(NL3490)
HWSNAME TYPE=(NL9840)
HWSNAME TYPE=(A04490)
HWSNAME TYPE=(A09490)
HWSNAME TYPE=(A09840)
HWSNAME TYPE=(A19490)
HWSNAME TYPE=(A19940)
*
* VIRTUAL DRIVES
HWSNAME TYPE=(VIRTCART)

```


Device Prefencing Considerations

The DEVTpref parameter of the HSC and MVS/CSC TAPEREQ statement allows users to request a higher priority for one type of StorageTek 36-track drive during drive prioritization processing. A second or third model of 36-track drive can be specified as alternate choices. This device prefencing is applicable to a library configuration containing a mixture of 4490, 9490 and 9490EE cartridge drives.

To enable this processing, define an esoteric to include all the desired device types by ACS location or in the entire library configuration. In the sample configuration, the esoteric, A0DEVT90, serves this purpose for ACS0.

During drive exclusion, if a TAPEREQ indicated DEVT(9490,4490) for an allocation, the SMC could substitute A0DEVT90 for the original unit name if A0DEVT90 is a subset (e.g., UNIT=3490).



Note: JES3 HWS processing can change this esoteric to A09490 or A04490 when reusing drives across steps.

ZEROSCR Considerations

When specifying the HSC ALLOC command, the MVS/CSC ALTER command, or the MVS/CSC startup parameter ZEROSCR with a value of YES/ON, create esoteric unit names that span ACS boundaries. As an example, the following esoterics could be added to the sample installation:

- CA0A1 – an esoteric containing all drives in ACS0 and ACS1
- A0A1X490 – an esoteric containing all 4490 and 9490 drives in ACS0 and ACS1.

Assume both ACSs contain scratch volumes.

- If the scratch request does not specify media or recording technique, the SMC can substitute CA0A1 for CART.
- If the scratch request asked for 36-track recording technique, the SMC can substitute A0A1X490 for 3490.

In this way, both ACSs remain eligible for the allocation.



Note: Once again, JES3 HWS can alter esoteric unit names after the SMC has selected its choice.

Virtual Drive Definition Considerations

Refer to the VTCS documentation for complete examples of JES3 initialization parameter statements for virtual drives.

SMC Normal Operations

The SMC runs on all processors that are active in a JES3 global and local environment. On both global and local processors, start the SMC and the library subsystem(s), the HSC, and/or MVS/CSC(s) before starting jobs requiring cartridge drives.

When the SMC and the library subsystem have initialized on the global processor and are communicating, the SMC performs drive exclusion, affinity separation, esoteric unit name replacement, fetch message suppression, drive prioritization, and mount deferral for both common and dynamic cartridge drive allocations. If the SMC has not completed initialization before jobs enter the JES3 C/I DSP, this processing is not performed. The PROMPT value on the NOSMC parameter of the SMCEHOOK macro delays one C/I DSP if the SMC has not initialized and reminds the operator to start the SMC.

When the SMC and the library subsystem have initialized on the local processor and are communicating, the SMC performs drive exclusion, affinity separation, and esoteric unit name replacement for dynamic cartridge drive allocations.



Notes:

- See the “Performing JES3 Post-Installation Tasks” appendix in the *NCS Installation Guide* for more information about the SMCEHOOK macro and its parameters.
- See Chapter 8, “Recovery Procedures” on page 161 for recovery procedures related to SMC, library subsystems, and JES3.

JES3 Constraints

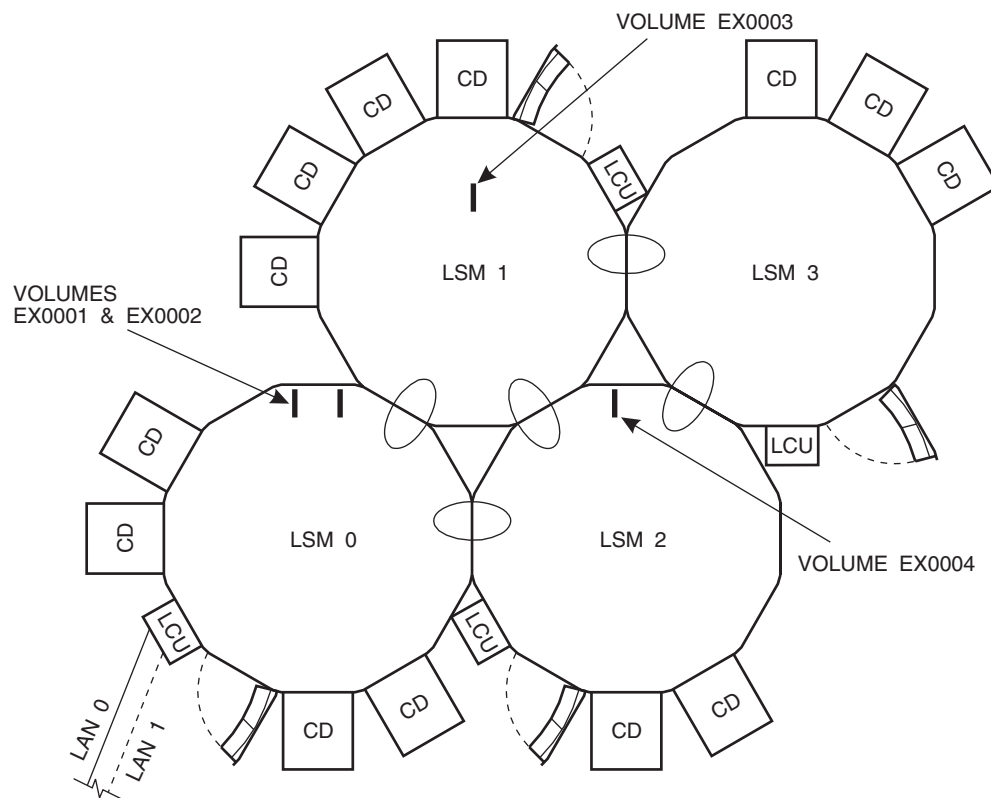
Timing Between C/I and MDS

A time window exists between C/I processing and MDS processing. A requested volume's location or a scratch subpool count can change during the interval between these two processes. When this situation happens, one or more volumes may need to be ejected from or entered into an ACS.

JES3 High Watermark Setup and LSM Pass-Thru Processing

When a job consists of multiple steps, JES3 HWS processing attempts to minimize the number of devices required. Thus a job consisting of multiple steps, each requesting one tape drive, can be allocated a single drive for the entire job. The following example shows the possible effects on pass-thru processing.

Figure 11 shows a library configuration containing four LSMs. All drives in the library are online and available.



C28635

Figure 11. Volume Locations for the Pass-thru Example

The following example shows the JCL for the job:

```
//STEP1 EXEC
//DD1 DD DSN=DSN.IN.LSM0,UNIT=3490,VOL=SER=(EX0001,EX0002)
//*
//STEP2 EXEC
//DD1 DD DSN=DSN.IN.LSM1,UNIT=3490,VOL=SER=EX0003
//*
//STEP3 EXEC
//DD1 DD DSN=DSN.IN.LSM2,UNIT=3490,VOL=SER=EX0004
//*
//STEP4 EXEC
//DD1 DD DSN=DSN.IN.LSM0,UNIT=3490,VOL=SER=(EX0001,EX0002)
```

Volumes EX0001 and EX0002 are in LSM0, EX0003 is in LSM1, and EX0004 is in LSM2 and all volumes are the same media and require the same recording technique. The SMC drive exclusion process picked the same esoteric for the allocation.

After the SMC drive exclusion process completes, JES3 HWS analysis determines that the maximum number of drives required for running the job is one. MDS processing allocates the device. Pass-thru processing occurs as follows:

- If the allocated drive is attached to LSM0, the number of pass-thrus is two (volume EX0003 moves from LSM1, and volume EX0004 moves from LSM2).
- If the allocated drive is attached to LSM1 or LSM2, the number of pass-thrus is three (volumes EX0001 and EX0002 move from LSM0, and either EX0003 or EX0004 moves, depending upon which LSM contains the drive).
- If the allocated device is attached to LSM3, the number of pass-thrus is four (all volumes move to LSM3).

The SMC drive prioritization process uses the pass-thru counts when setting a priority for a drive. However, if the “preferred” drive is not available, other available drives can be selected.

Library Subsystem Selection

In order to support the execution of SMC on a different host from the library subsystem, all HSC and MVS/CSC allocation and message processing user exits are now managed and invoked by the SMC. Therefore, only one version (HSC or MVS/CSC) of each user exit can be executed.

For this reason, Sun Microsystems recommends that you carefully review the contents of your existing user exits to determine the most effective method to direct allocations to the correct library. Specifically, the use of the user exit return code “use non library drives” to direct allocation to another library subsystem is no longer supported. Sun Microsystems recommends that you use TAPEREQ or user exit esoterics to direct allocations to the appropriate library subsystem.

An alternative method to select a library subsystem is to alter your scratch or specific allocation user exit to use the new HSC format, and explicitly select a LIBRARY name to direct a request.

See “SMC Library Subsystem Selection” on page 16 for information on how the SMC selects among multiple library subsystems to determine an “owner” for each allocation request (i.e. each DD in a job step may have a different library subsystem owner).

SMC DFSMS Processing

The SMC interface to DFSMS provides the following capabilities:

- esoteric substitution during MVS JCL interpretation
- at device allocation time, the DFSMS DATACLAS value can request a specific recording technique and/or media.
- at device allocation time, the DFSMS MGMTCLAS value can be used for VTV allocations.
- at volume mount, the DFSMS MGMTCLAS value can be assigned to a VTV.
- at volume mount, the DFSMS DATACLAS value can request a media for a scratch request.

Enabling/Disabling the SMC DFSMS Interface

To enable the SMC DFSMS interface, specify the SMS parameter of the ALLOCDef command as follows:

```
ALLOCDEF SMS=ON
```

To disable the SMC DFSMS interface, specify the SMS parameter of the ALLOCDef command as follows:

```
ALLOCDEF SMS=OFF
```

The SMC DFSMS interface can be selectively enabled or disabled by job or job step by specifying the SMS parameter of the ALLOCJob command.

Tailoring the SMC DFSMS Interface

The SMSDef command can be used to tailor the default SMC DFSMS support for your installation's requirements. This command provides the ability to include or bypass certain SMC DFSMS functions. See "SMSDef Command" on page 147 for more information about the options that can be specified.

DATACLAS, STORCLAS, and MGMTCLAS Specification

DFSMS DATACLAS can be specified by using the DATACLAS parameter on the DD JCL statement or by executing an Automatic Class Selection (ACS) routine. STORCLAS and MGMTCLAS can only be specified by ACS routines.



Note: STORCLAS and MGMTCLAS JCL parameters are not supported by the SMC DFSMS interface due to conflicts with IBM MVS DFSMS. Using the STORCLAS JCL parameter causes a data set to become DFSMS-managed, and the MGMTCLAS JCL parameter requires a DFSMS-managed data set. The data sets assigned STORCLAS and MGMTCLAS values in the STKTAP1 environment are not actually DFSMS-managed.

Invoking ACS Routines

IBM DFSMS invokes ACS routines with the variable `&ACSENVIR` set to `ALLOC` before the SMC invokes the ACS routines with variable `&ACSENVIR` set to `STKTAP1`.

The SMC invokes the ACS routines at the following points in processing:

- SSI55 Interpreter/Dynamic Allocation Exit (IDAX)
- JES3 Converter/Interpreter (C/I)
- SSI23 JES3 Dynamic Allocation
- JES3 Main Device Scheduler (MDS)
- mount message interception.

The ACS routines are invoked in the following order:

1. data class
2. storage class
3. management class
4. storage group.

Management class and storage group ACS routines are called only if a storage class is assigned.

DFSMS Automatic Class Selection (ACS) Routine Environment for SMC

The following list of read-only variables is passed by SMC to DFSMS when the information is available to the SMC. Not all variables are available for every call to the ACS routines. In particular, processes that occur in the JES3 address space, such as MDS, do not provide the SMC access to the MVS control blocks that contain the values for these fields. See the descriptions of each DFSMS interface for exceptions.

- `&ACSENVIR` (equals `STKTAP1` for the SMC interface)
- `&ALLVOL`
- `&ANYVOL`
- `&DATACLAS`
- `&DD`
- `&DSORG`
- `&DSN`
- `&DSTYPE`
- `&EXPDT`
- `&FILENUM`
- `&JOB`
- `&LABEL`
- `&NVOL`
- `&PGM`
- `&RETPD`
- `&SYSNAME`
- `&SYSPLEX`
- `&UNIT`.

In the STKTAP1 environment, the &ANYVOL variable is used only to match a specific VOLSER and does not contain the “REF=xx” values for VOL=REF allocations.

The &DATACLAS field is set when the JCL DD statement specifies this parameter.

If your installation uses the IGDACSXT routine to modify the read-only variables before the DFSMS ACS routine calls, be aware that the following variables, even if initialized, are not passed to the DFSMS ACS routines when &ACSENVIR is set to STKTAP1.

- &ACCT_JOB
- &ACCT_STEP
- &GROUP
- &MSGVP
- &USER
- &XMODE

Refer to the IBM manual *DFSMSdfp Storage Administration Reference* for more information about constraints when using read-only variables.

Esoteric Substitution During IDAX

When a job is in the JCL interpretation phase, the unit name (esoteric) can be changed indiscriminately (for example, you could substitute disk for tape). Considerations:

- Esoteric substitution applies to all newly created data sets. New allocations are signified by specifying DISP=NEW, or in some cases, DISP=MOD in the JCL (refer to the IBM *MVS JCL Reference* manual for a list of requirements when using DISP=MOD to specify a new data set).
- Unit affinity chains are always separated if ACS routines return different esoterics for different members of the chain.
- VOL=REF chains within a job are validated and updated with the head-of-chain esoteric if necessary to ensure that volume references are honored.

When the SMC locates a new allocation that is not DFSMS-managed, the SMC calls the DFSMS ACS routines with the environment variable &ACSENVIR set to STKTAP1. When the ACS routines return a pair of storage group names with the first name being STK1, the second storage group name replaces the original value of the UNIT parameter of the DD statement. For example, the following DD statement allocates a 3490 nonlibrary tape drive (esoteric TNLIB):

```
//DD1 DD DSN=&TEMP.DSN,UNIT=TNLIB,DISP=NEW
```

Assume that your installation wants to move all new tape allocations into a tape library. You can code the DFSMS ACS routines to return storage groups STK1 and a library esoteric such as TACS0, where TACS0 contains all drives in library location ACS0. After esoteric replacement, the only drives considered for the allocation are contained within the TACS0 esoteric. The nonlibrary drives are no longer considered eligible for the allocation.

Defining Storage Groups and Storage Classes

To enable esoteric substitution during IDAX, complete the following steps:

1. Define a storage group of type POOL named STK1.
2. Define a storage group of type POOL for each possible esoteric returned, giving it the same name as the esoteric name.
3. Define at least one volume to each storage group. Sun Microsystems recommends that you use a nonexistent volume for this definition.
4. Create a storage class ACS routine that, when it is invoked by the SMC (i.e., when the environmental variable is `&ACSENVIR=STKTAP1`), returns a storage class to be passed to the storage group ACS routine. Refer to Figure 12 on page 102.
5. Create a storage group ACS routine that, when the environmental variable is `&ACSENVIR=STKTAP1`, returns two storage groups (e.g., `&STORGRP='STK1', 'CART'`). In this case, “STK1” is the first storage group returned, and “CART,” the replacement esoteric, is the second. Refer to Figure 12.
6. Test for the `&ACSENVIR=STKTAP1` read-only variable in the storage class and storage group ACS routines. This enables you to prevent the storage class being assigned when MVS invokes the ACS routines. If MVS invokes the SMC version of the ACS routines, the data set becomes DFSMS-managed and all SMC allocation functions are bypassed.
7. Return both a storage class and a storage group during the SMC invocation of the DFSMS ACS routines. If a storage class is returned but a storage group is not, DFSMS issues a message stating that allocation has failed, but in reality that may not be the case.

Figure 12 on page 102 provides sample storage class and storage group routines.

```

PROC STORCLAS

FILTLIST LOCALDSN INCLUDE (BACKUP*.**,
                           PROD.BKP*.**)

FILTLIST RMTDSN INCLUDE (PROD.OFFSITE.**)
```

IF &ACSENVIR = 'STKTAP1' THEN

```

  SELECT
  WHEN &DSN = &LOCALDSN
    SET &STORCLAS = 'CART'
  WHEN &DSN = &RMTDSN
    SET &STORCLAS = 'RMT CART'
  END
END
```

=====

```

PROC STORGRP

IF &ACSENVIR = 'STKTAP1' THEN
  SELECT
  WHEN &STORCLAS = 'CART'
    SET &STORGRP = 'STK1', 'CART'
  WHEN &STORCLAS = 'RMT CART'
    SET &STORGRP = 'STK1', 'RMT CART'
  END
END
```

Figure 12. Sample Storage Class/Storage Group Routines

Availability of Read-only Variables

During DFSMS STORCLAS/STORGRP ACS routine processing, all read-only variables listed in “DFSMS Automatic Class Selection (ACS) Routine Environment for SMC” on page 99, except &DSORG, are available to the IDAX interface for esoteric substitution. The &DATACLAS field is set when the DD statement in the JCL specifies this parameter.

Validating DFSMS STORCLAS/STORGRP ACS Routine Execution

DFSMS provides the following ways to validate correct ACS routine execution:

- DFSMS WRITE statement in the ACS routines
- DFSMS ISMF test function.

When the SMC invokes the DFSMS ACS routines, all messages issued by DFSMS WRITE statements are routed to the GTF trace file if SMC tracing is active.

Refer to the publication *DFSMS/MVS DFSMSdfp Storage Administration Reference* for further information about writing and testing DFSMS ACS routines.

Retrieving DFSMS Values During Allocation and Volume Mount

Requesting Recording Technique/Media using DFSMS DATACLAS

During allocation processing, the media and/or recording technique can be specified for a DD allocation by coding:

- TAPEREQ statement
- DATACLAS parameter on the DD statement
- DATACLAS ACS routines.

During scratch volume mount processing, only the volume media can be specified.



Note: DATACLAS media and recording technique values override TAPEREQ values.

Media/Rectech Support

The following media and recording techniques are supported:

Media:	
MEDIA1	Standard capacity for both 18-track and 36-track media types (this is the default)
MEDIA2	Enhanced capacity 36-track media
Recording Techniques:	
18TRACK	18-track recording technique
36TRACK	36-track recording technique

Defining Data Classes

To use DATACLAS, customers must define their own data class and specify the recording technique and media. If the recording technique is left blank, the SMC defaults to allowing both 18TRACK and 36TRACK to remain eligible for the allocation request, unlike MVS, which defaults to 36TRACK.

As an example, a customer can define a DATACLAS named STD18 that specifies a recording technique of 18TRACK and standard media.

When specifying a DATACLAS by coding it on the DD statement, the system programmer must have installed the optional Type-1 modification to IATIICM. The Type-1 modification retrieves the DATACLAS, determines its recording technique and media specification, and saves it in the IJSMEDIA field. This information is used during C/I processing (IATIIP1) and then passed to the JSTMEDIA field and used again during MDS processing.

When using an ACS routine to supply a DATACLAS to set recording technique and media, do not specify `&ACSENVIR='STKTAP1'`. The DATACLAS ACS routine is invoked both by the SMC and MVS, and the resulting recording technique and media should be consistent.

The following figure provides a sample data class routine.

```
PROC DATACLAS

FILTLIST RMTDSN INCLUDE(PROD.OFFSITE.**)

  SELECT
  WHEN &DSN = &RMTDSN
    SET &DATACLAS = 'ECART36'
  END
```

Figure 13. Sample Data Class Routine

In the example above, assume DATACLAS ECART36 is defined with a recording technique of 36-track and a media type of MEDIA2. This data class routine example assigns ECART36 to data sets with names that start with “PROD.OFFSITE.”

By default, any DATACLAS value specified by the DFSMS ACS routine overrides the DATACLAS parameter specified on the JCL statement. Your installation can change this behavior by adding the following statements to your DFSMS DATACLAS ACS routine.

```
  WHEN (&DATACLAS NE ' ')
  DO
    SET &DATACLAS = &DATACLAS
  EXIT
  END
```



Note: During JES3 C/I POSTSCAN processing, JES3 MDS processing, and IAT5210 mount message processing, the DATACLAS name specified on the JCL statement is not available and is not passed to the ACS routines. Thus, the statements above may not produce the expected results.

Requesting Management Class Using DFSMS ACS Routines

During allocation and mount processing, a management class name can be specified for a virtual allocation by coding any of the following:

- TAPEREQ statement
- MGMTCLAS ACS routine.

The MGMTCLAS JCL parameter is not supported by the SMC DFSMS interface.



Note: The MGMTCLAS management class name overrides a TAPEREQ management class name.

When writing a management class routine, keep in mind:

- The management class routine is invoked only when a storage class is assigned.

- The management class routine must test for the &ACSENVIR='STKTAP1' read-only variable value.

DFSMS issued messages for JES3 static allocations are routed to the SMC GTF trace file.

The following figure provides a sample management class routine.

```

PROC STORCLAS
FILTLIST CART INCLUDE ('CART')

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
      WHEN (&UNIT EQ &CART)
        SET &STORCLAS = 'VIRTAPE'
      WHEN (&UNIT NE &CART)
        SET &STORCLAS = 'STKDFLT'
    END
  END

=====

PROC STORGRP

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
      WHEN (&STORCLAS = 'VIRTAPE')
        SET &STORGRP = 'STK1', 'VDRIVES'
    END
  END

=====

PROC MGMTCLAS

FILTLIST LOCAL INCLUDE(BACKUP*.*,
                       PROD.BKP*.)

FILTLIST REMOTE INCLUDE(PROD.OFFSITE.*)

  IF &ACSENVIR = 'STKTAP1' THEN
    SELECT
      WHEN (&DSN = &LOCAL)
        SET &MGMTCLAS = 'INVTAPE'
      WHEN (&DSN = &REMOTE)
        SET &MGMTCLAS = 'OFFVTAPE'
    END
  END

```

Figure 14. Creating Management Class Routines



Note: In this example, the STORCLAS routine assigns a storage class to every call. This ensures that the MGMTCLAS routine is also driven at mount time. Only the VTCS component of NCS currently uses the management class value.

Availability of Read-only Variables

During DFSMS ACS routine processing, the SMC sets the values for all read-only variables when the information is available. Not all information is available for each process for which the SMC calls ACS routines.

During JES3 C/I POSTSCAN processing, the following read-only variables are **not** available to the SMC:

- &DATACLAS (if specified in JCL)
- &EXPDT
- &PGM
- &RETPD.

During SSI23 dynamic allocation processing, all read-only variables are available to the ACS routines.

During JES3 MDS processing, the following read-only variables are **not** available to the SMC:

- &DATACLAS (if specified in JCL)
- &EXPDT
- &PGM
- &RETPD.

During IAT5210 mount message processing, the following read-only variables are **not** available to the SMC:

- &DATACLAS (if specified in JCL)
- &EXPDT
- &PGM
- &RETPD
- &UNIT (does not contain the original esoteric but does contain the selected device number, such as 0A10).

When processing mount message IEF233A for a dynamic allocation, the &DATACLAS value specified by the SVC99 text unit DADACL is not available.

When processing mount message IEC501A, the SMC sets the &UNIT read-only variable to the generic unit type, such as 3490.



Note: Each SMC interaction with DFSMS invokes all levels of ACS routines. The variable availability listed above applies to all ACS routines.

Validating DFSMS ACS Routine Execution

DFSMS provides the following ways to validate correct ACS routine execution:

- DFSMS WRITE statement in the ACS routines
- DFSMS ISMF test function.

When the SMC invokes the DFSMS ACS routines, all messages issued by DFSMS WRITE statements or by DFSMS processing are routed to the GTF trace file if SMC tracing is active. During SSI23 Dynamic Allocation processing and MVS mount message processing, these messages are also issued to the SYSMSG data set of the job.

Refer to the publication *DFSMS/MVS DFSMSdfp Storage Administration Reference* for further information about writing and testing DFSMS ACS routines.

GDG ALL Processing

GDG ALL processing is the referencing of all generations of a generation data group, using the JCL GDG base entry instead of referencing each individual generation by relative generation number.

There is a GDG ALL processing condition that the SMC cannot properly influence. This condition occurs when a step indicates that a new generation will be created, but fails to execute after being conditionally bypassed. A subsequent step then performs the GDG ALL processing.

Possible scenarios include, but are not limited to the following:

1. The new generation would have been created on a different device type than the current generation, and the creation of the new generation is bypassed.

As a result, the current generation is allocated to a device type that cannot support the media that the current generation resides on.

2. The last generation of the GDG resides on a different media than all other generations and the creation of the new generation is bypassed.

As a result, the last generation is not influenced and is requested to be mounted on a device type that does not support its media.

3. In a multiple VTSS environment, the new generation would have been created in one VTSS but the current generation resides in another VTSS.

As a result, the allocation is processed in the first VTSS and an automatic migrate/recall occurs.

For example, if the new generation would have been created in VTSS1 and either the first, last or both generations exist in VTSS2, these generations are transferred into VTSS1 during the GDG ALL processing step.

SMC recommendations for these scenarios are:

- Separate these steps into two jobs.
- Ensure that all generations use the same media/device type.
- If you are using multiple VTSSs, ensure that all generations have access to the same ACS to enable the transfer of the generations that may exist in a different VTSS.

Interaction with Other Software

Automated Operations

Customers who use an automated operations product should review the following WTORs for possible auto-ops rule changes:

- SMC0110 - issued during SMC swap processing

Open Type J

The SMC allocation enhancements may not operate if you use the MVS Open Type J macro. Since that macro allows you to change volume serial numbers or data set names at open time, information available at Job Step Allocation time may be incorrect as interpreted by the SMC.



Note: Some vendor software products use MVS Open Type J. If you are experiencing unexpected allocation results using a vendor software product, check with the vendor to determine if Open Type J is used and follow the recommendations below.

SMC allocation may be influencing MVS allocation erroneously based on information that may have changed at open time. To prevent this problem when using the Open Type J macro, specify the appropriate esoteric in the JCL or specify an appropriate esoteric in an applicable TAPEREQ control statement.

SAMS: DISK (DMS)

Sterling Software's SAMS: DISK (DMS) has two methods of allocating transports:

- allocates transports at session startup, holds onto the transports throughout the session and uses Open Type J (refer to the section about "Open Type J").
- uses dynamic allocation (DYNALLOC) to allocate transports when required.

The SMC allocates correctly when dynamic allocation is used. Therefore, the latter method of allocating transports is recommended.

MVS Security Packages

Ensure that your MVS security package (e.g. RACF, TopSecret) is configured to grant SMC the necessary authority to respond to MVS swap messages.

Chapter 6. SMC Message Handling

Overview

The SMC intercepts specific MVS, JES3, and Tape Management System (TMS) messages related to mount, dismount, and swap operations. When the intercepted message includes a drive defined to a library subsystem (HSC or MVS/CSC), the SMC directs the owning library subsystem to perform the requested operation.

Messages that are intercepted by the SMC are listed in Appendix A, “Intercepted Messages” on page 241.



Note: The IATUX71 user exit must be installed to allow JES3 mount messages to be processed. Refer to the *NCS Installation Guide* for more information.

User Directed Message Handling

If the SMC does not currently support the Tape Management System at your installation, you can still direct the SMC to intercept the specific messages issued by your TMS. Use the USERMsg operator command to define these additional messages. See “USERMsg Command” on page 160 for more information.

User Exit 01 (SLSUX01 and SCSUX01) allows you to change or enhance actions taken on intercepted messages, and can direct the SMC to perform actions on messages that are not in the intercepted message list.

The SMC calls the library subsystem user exit for each intercepted message. This includes the default messages listed in Appendix A, “Intercepted Messages” and all messages defined using the USERMsg command.



Notes:

- Only messages intercepted by the SMC are passed to the user exit.
- The SMC does not support the user exit return code of REPLY.

Message Handling Policies

The SMC honors the following MVS, HSC and MVS/CSC policies related to mount, dismount, and swap message handling:

MVS Policies

The System Authorization Facility (SAF) can be used to protect tapes at the volume level (CLASS=TAPEVOL) using your current security software. If defined, the SMC honors the policies defined through the SAF interface regarding the write-protect requirement for a volume mounted on a library transport. The SMC invokes the SAF interface by issuing a RACROUTE macro, and protects read-only volumes through the ACS Virtual Thumbwheel (VTW) support.

HSC Policies

The LIBGEN delete disposition policy is honored by the SMC to control whether volumes dismounted by MVS with the delete disposition are to be scratched.

The MNTD MOUNTmsg(Roll|Noroll) policy is honored by the SMC to determine whether mount messages handled by the SMC are allowed to roll off the operator console screen before the mount request is satisfied.

The MNTD VOLWatch(Off|ON) policy is honored by the SMC to determine whether the SMC issues an SMC0106 message whenever a library volume is requested to be mounted on a non-library device.

The AMPND startup parameter is honored by the SMC to determine whether pending mounts for HSC owned drives are automated. This occurs when the SMC recognizes that an HSC library has become active at FULL service level. This occurs only when the SMC first establishes communication with the HSC, through an MVS mount event, or when the operator issues the SMC RESYNChronize command. See “RESYNChronize Command” on page 141 for more information.

MVS/CSC Policies

The ALTER DELDisp(SCRTCH|NOSCRTCH) policy is honored by SMC to control whether volumes dismounted by MVS with the delete disposition are to be scratched.

The ALTER WTOdesc(YES|NO) policy is honored by the SMC to determine whether mount messages handled by the SMC are allowed to roll off the operator console screen before the mount request is satisfied.

The AMPND startup parameter is honored by SMC to determine whether pending mounts for MVS/CSC owned drives are automated. This occurs when the SMC recognizes that an MVS/CSC library has become active. This occurs only when the SMC first establishes communication with the MVS/CSC, through an MVS allocation or mount event, or when the operator issues the SMC RESYNChronize command. See “RESYNChronize Command” on page 141 for more information.

Tape Management System Support

The SMC interprets Mount, Dismount, and Swap messages from the following Tape Management Systems:

- CA-1
- CA-DYNAM/TLMS
- DFSMSrmm
- AutoMedia (Zara)
- CONTROL-T

For tape management systems that supply a subpool, the subpool is interpreted by the SMC and used as the requested subpool name, unless overridden by user exit 01 or a TAPEREQ statement. Related messages include:

- CTS002
- CTT101A
- CTT104A
- TMS002

SMC Swap Processing

The SMC automates the swap process (I/O error or operator initiated) in the same manner as HSC enhanced swap processing. This eliminates the need for operators to find a compatible “swap-to” device when DDR has chosen an incompatible device. If the SMC cannot find a compatible swap-to device, or if all compatible devices are busy, the SMC issues a message and control is returned to DDR processing with no further SMC involvement. Note that enhanced swap processing is the only mode supported in SMC.

On JES3 systems, the SMC does not influence the swap-to device selection. The SMC does not issue SMC0107 or SMC0110. JES3 can select a compatible drive itself based on proper initialization deck definitions. The SMC still automates the swap when the IGF502E is received.

The SMC swap process begins when one of the following messages is issued:

```
IGF500I SWAP XXX1 TO XXX2 - I/O ERROR  
IGF503I ERROR ON XXX1, SELECT NEW DEVICE  
IGF509I SWAP XXX1 - I/O ERROR
```

If device *XXX1* is known to the SMC as a library or nonlibrary defined device, the SMC suppresses the message and begins the automatic swap process.

The SMC issues one of two messages:

```
SMC0108  No compatible drive found for SWAP processing
```

Or when a compatible drive can be selected by the SMC:

```
SMC0107  SWAP volser from XXX1 to XXX2
```

Device *XXX2* is the SMC-selected device that has been determined to be compatible for the swap. The SMC next suppresses the MVS IGF500D or IGF509D message and replaces the message with:

```
SMC0110  Allow swap of volser from XXX1 to XXX2;  
Reply 'Y', or 'N' or DEVICE
```

The operator may approve the device selected, cancel the swap or choose a different device. If the operator selects a different device, the SMC accepts the device with no further compatibility checking.

If the reply is 'Y' or a new device, MVS issues the following message:

```
IGF502E  PROCEED WITH SWAP OF XXX1 TO XXX2
```

If *XXX1* is a library owned device, the dismount of the volume is automated.

If *XXX2* is a library owned device, the mount of the volume is automated.



Note: Ensure that your MVS security package (e.g. RACF, TopSecret) is configured to grant SMC the necessary authority to respond to MVS swap messages IGF500D and IGF509D.

HSC Mount-Related Messages

Certain mount-related messages may still be issued by the HSC due to error conditions.

- SLS0088D is issued when, due to an error condition, a repeated mount for the same volume is required.
- SLS1075D is issued when a dismounted volume encounters an I/O or other type of error.

In addition, the following HSC messages are replaced by SMC messages:

- SLS4306I and SLS4308I are replaced by SMC0106.
- SLS4310D is replaced by SMC0110.

Managing HSC Mounts from the SMC Client

The SMC 6.0 client server architecture provides the ability to perform management of certain mount/dismount exception conditions from the client console rather than from the server console. Conditions handled by the SMC include the following:

- Displaying messages on the drive and console indicating the volume and its location for manual mounts when an LSM is offline.
- When a drive is found to be loaded with another volume during a mount request from a job in execution or JES3 setup processing, the dismount is forced and the mount is automatically redriven based on the UCB outstanding mount status (or JES3 SETUNIT).



Note: When a mount or dismount message is intercepted by SMC and directed to an HSC server, the SLS0107D message is not issued on the server or client. When a loaded drive condition is detected by the HSC for an SMC directed dismount, the HSC waits one minute for the drive to unload, and then terminates the dismount. If the dismount is performed as part of a mount request from the SMC, the loaded drive status is returned to the SMC, and the SMC then redrives the mount request if the mount is still pending.

- When a drive or volume is in use by another process, verifying that the mount is still pending, periodically retrying the request, and allowing the operator to cancel or retry the request at any time.
- When the LIBrary WTORdest Client option is selected, certain WTOR messages are not issued on the HSC server but are sent directly to the client console instead. The operator response to the message is then transmitted back to the server, as SMC redrives the mount or dismount to the server. The following HSC mount/dismount WTOR messages are currently supported with this option:
 - SLS0134
 - SLS0905
 - SLS2126
 - SLS2905
 - SLS2984
 - SLS0109
 - SLS4084

Refer to the *HSC Messages and Codes Guide* for more information about these messages.

Chapter 7. Operator Commands

Overview

This chapter provides information for the following SMC commands:

- ALLOCDef
- ALLOCJob
- CMDDef
- LIBrary
- LIst
- MSGDef
- MSGJob
- READ
- RESYNChronize
- SERVer
- SMSDef
- TCPip
- TRace
- UEXit
- USERMsg

Issuing SMC Commands

SMC operator commands can be issued using either the MVS Modify command, or a user-defined command prefix defined in the SMC CMDDef command.

- Issuing SMC commands using the MVS Modify command
- Issuing SMC commands using the SMC command prefix
- Specifying SMC commands in the SMCPARMS or SMCCMDS data set.

Issuing SMC Commands Using the MVS Modify Command

The following information is included when an SMC command is issued using the MVS Modify (F) command:

- MVS Modify command (F)
- SMC started task name
- command name
- parameters (optional or required).

SMC commands are entered in the following format:

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

The following example illustrates the MSGDef command:

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



Notes:

- 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 can be separated with any number of spaces, and may optionally include an equal (=) sign. Values may also be included in parentheses.

Issuing SMC Commands Using the SMC Command Prefix

SMC operator commands can be issued using an SMC command prefix. This prefix is defined using the SMC CMDDef command. See “CMDDef Command” on page 127 for more information about this command.

Specifying SMC Commands in the SMCPARMS or SMCCMDS Data Set

SMC operator commands can be specified in the SMCPARMS or SMCCMDS data set. These commands are automatically processed at startup. See “Control Statements” on page xxiv for syntax conventions used when specifying commands in the SMCPARMS or SMCCMDS data set.

- The SMCCMDS data set can be reprocessed while the SMC is active using the READ operator command. The following is a sample SMCCMDS member entry:

```
MSGDEF CASE(MIXED) LVL(4)
ALLOCDEF MINLVL=4 SEPLVL=MIN
```

- The SMCPARMS data set is used for user-configured items that cannot be changed while the SMC is active. The SMCPARMS data set **cannot** be reprocessed using the READ command. The following is a sample SMCPARMS member entry:

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



Note: Sun Microsystems recommends 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.

Operator commands that allow specification of JOBname, STEPname, or PROCstep are evaluated 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*.

SMC Command Descriptions

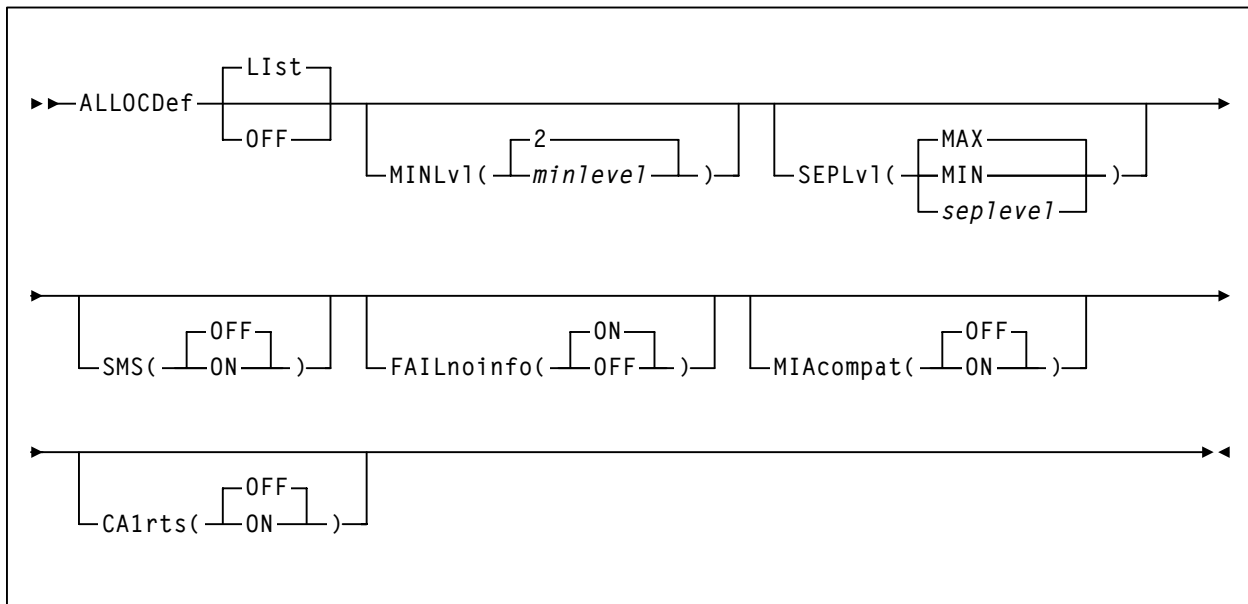
This section describes the SMC operator commands. Command syntax, parameter descriptions, and examples are included for each command.

See “Syntax Flow Diagrams” on page xix for syntax flow diagramming conventions.

ALLOCDDef Command

The SMC ALLOCDDef command is used to specify system default parameters used within the SMC subsystem to control allocation.

Syntax



Command Name

ALLOCDDef

initiates the ALLOCDDef command.

Parameter Descriptions

List

optionally, lists current default allocation settings.

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

OFF

optionally, resets all default allocation values to original SMC default settings. It **cannot** be specified with any other parameter.

MINLvl

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.

minlevel

indicates the desired minimum allocation exclusion level. Valid values are 1-8. See Chapter 4, “SMC Allocation in a JES2 Environment” or Chapter 5, “SMC Allocation in a JES3 Environment” for more information regarding SMC exclusion levels.



Note: 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.



Note: 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.

MIN

do **not** separate chains beyond the minimum level.

MAX

separate chains for conflicting exclusion criteria whenever sufficient drives are available. This is the default value.

seplevel

indicates a separation level between the minimum and maximum levels. When sufficient drives exist, the SMC attempts to separate chains at this level. See Chapter 4, “SMC Allocation in a JES2 Environment” or Chapter 5, “SMC Allocation in a JES3 Environment” for more information regarding SMC exclusion levels.



Note: The *seplevel* value **cannot** be less than the *minlevel* value.

SMS

enables and disables the DFSMS interface. When the DFSMS is enabled, SMC invokes the DFSMS ACS routines. The returned constructs are used to influence device allocation of data sets.

OFF

disables the DFSMS interface. This is the default value.

ON

enables the DFSMS interface.

FAILnoinfo

optionally, specifies whether or not the SMC fails a job during allocation when a communication failure prevents the retrieval of volume information from a library subsystem or library server, or if no libraries are available.

OFF

do not fail the job when a communication failure occurs during the volume lookup process or if no libraries are available. This is the default value.

ON

if a communication failure occurs during volume lookup, or if no libraries are available, mark all devices ineligible and fail the job.

MICompat

optionally, specifies whether or not the EDL is to be modified 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.

OFF

do not update the EDL at SSI24 time. This is the default value.

ON

update the EDL at SSI24 time. Specify this value if you use Unicenter CA-MIA.

CA1rts

optionally, specifies whether 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.

OFF

do not update DEFER status at SSI24 time. This is the default value.

ON

update DEFER status at SSI24 time. Specify this value if you use the Real Time Stacking feature of CA-1.

Example

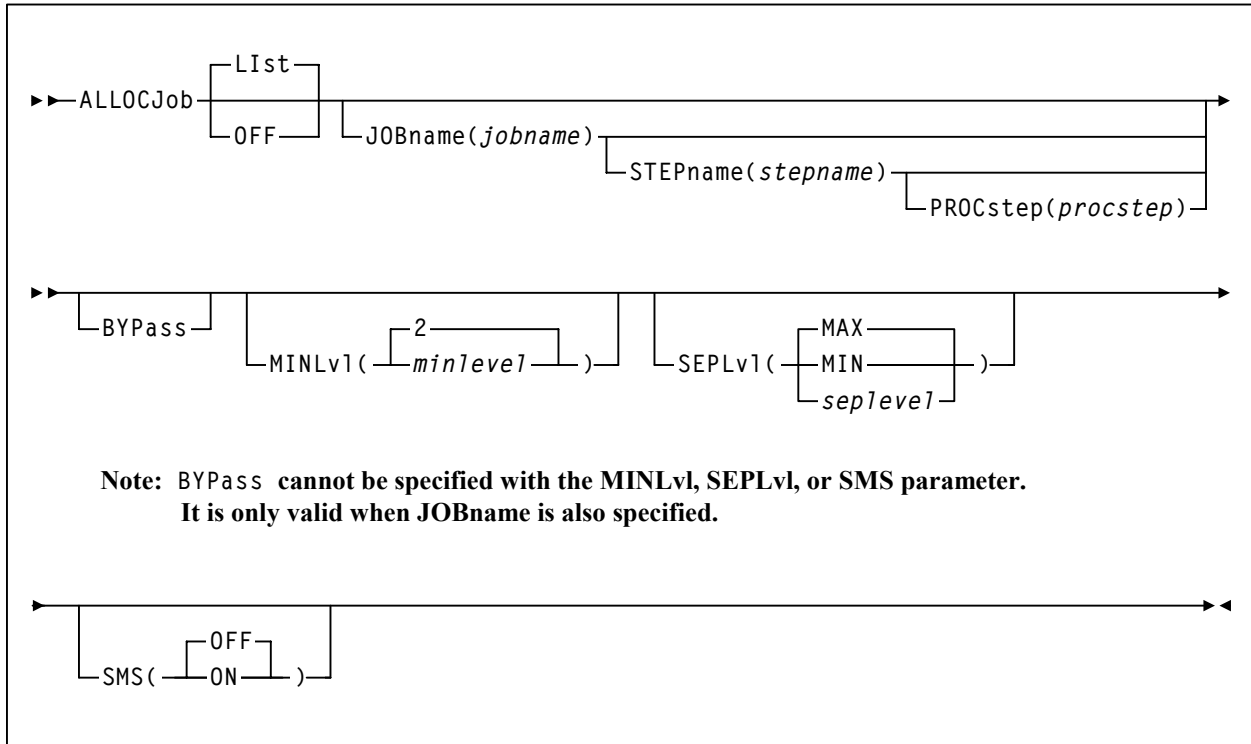
In the following example, the ALLOCDef command sets the minimum level of allocation exclusion to 4 and specifies that unit affinity and GDGALL chains are separated based on levels 1-4.

```
ALLOCDEF MINLVL=4 SEPLVL=MIN
```

ALLOCJob Command

The SMC ALLOCJob command is used to override SMC default allocation parameters by job name, step name, and PROC step.

Syntax



Command Name

ALLOCJob

initiates the ALLOCJob command.

Parameter Descriptions

List

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

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

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

optionally, specifies a job name.

This parameter is required unless LList 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

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

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 (job step, step name, PROC step).

This parameter **cannot** be specified with the MINLvl, SEPLvl, or SMS parameter. It is **only** valid when JOBname is also specified.

MINLvl

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 drive exclusion level. Valid values are 1-8. See Chapter 4, “SMC Allocation in a JES2 Environment” or Chapter 5, “SMC Allocation in a JES3 Environment” for information regarding SMC exclusion levels.



Note: 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.

MIN

do **not** separate chains beyond the minimum level.

MAX

separate chains for conflicting exclusion criteria whenever sufficient drives are available. This is the default value.

seplevel

indicates a level between the minimum and maximum levels. When sufficient drives exist, the SMC attempts to separate chains at this level. See Chapter 4, “SMC Allocation in a JES2 Environment” or Chapter 5, “SMC Allocation in a JES3 Environment” for more information regarding SMC exclusion levels.



Note: The *seplevel* value **cannot** be less than the *minlevel* value.

ALLOCJob

SMS

enables and disables the DFSMS interface. When the DFSMS is enabled, SMC invokes the DFSMS ACS routines. The returned constructs are used to influence device allocation of data sets.

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

OFF

disables the DFSMS interface. This is the default value.

ON

enables the DFSMS interface.

Example

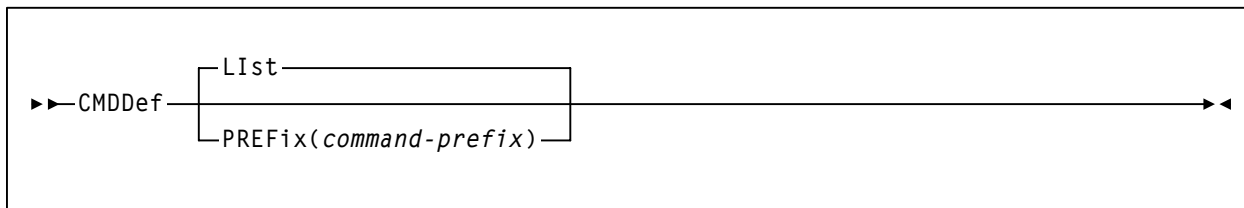
In the following example, the ALLOCJob command specifies that jobs with stepnames beginning with “STEP1” are bypassed during allocation influencing. These are the **only** steps that are bypassed.

```
ALLOCJOB JOB=PRODJOB1 STEP=STEP1* BYPASS
```

CMDDef Command

The SMC CMDDef command is used to assign an SMC command prefix.

Syntax



Command Name

CMDDef

initiates the CMDDef command.

Parameter Descriptions

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

optionally, specifies a command prefix for the SMC subsystem.



Note: This parameter may **only** be specified in the SMCPARMS data set.

command-prefix

indicates the desired command prefix characters.

The value entered **must** be one to eight characters in length and **must** 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.

CMDDef

Example

In the following example, the CMDDef command is specified in the SMCPARMS data set as follows:

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

As a result, SMC operator commands can be entered as follows:

```
B@F$MSGDEF CASE(MIXED)
```

LIBrary Command

The SMC LIBrary command defines a specific StorageTek tape hardware configuration, normally represented by a single CDS.

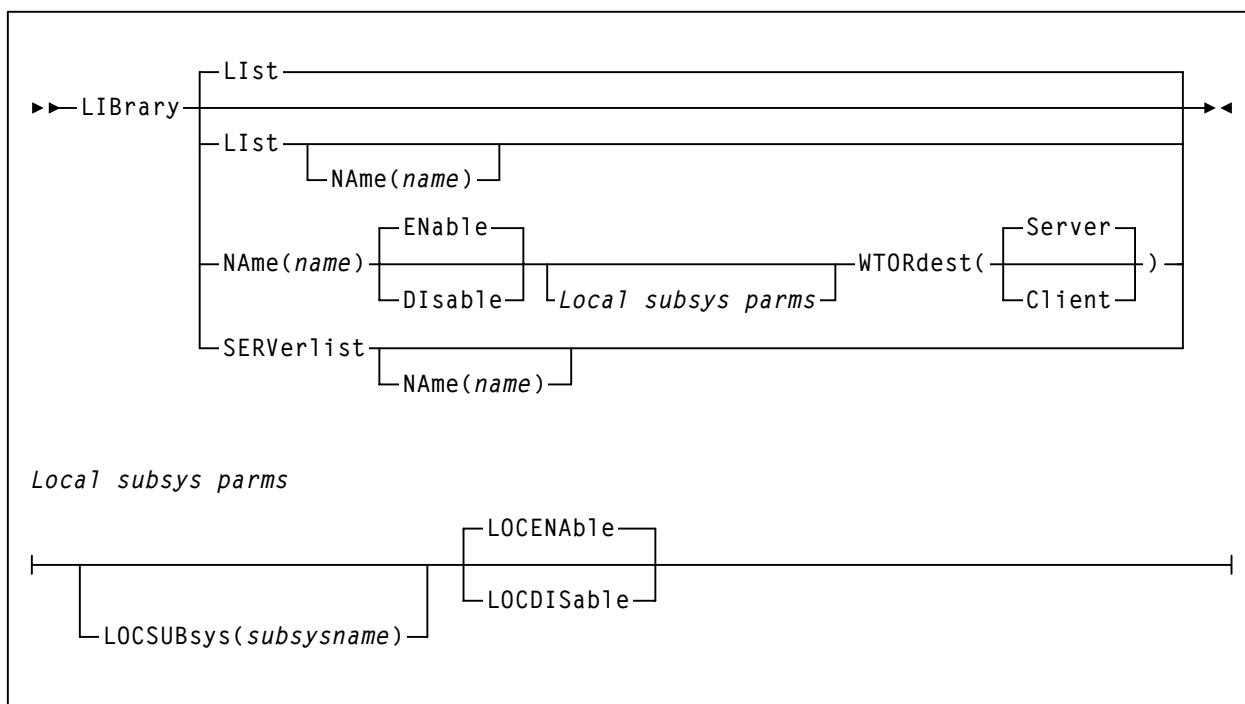


Notes:

- If a LIBrary command is specified, then all libraries must be explicitly defined.
- If the LIBrary command is not specified, the SMC uses active HSC and MVS/CSC subsystems defined in the host's SSCVT chain.

The LIBrary command can also be used to list libraries that the SMC tries to communicate with, and report their status.

Syntax



LIBrary

Command Name

LIBrary
initiates the LIBrary command

Parameter Descriptions

Llist
optionally, lists the specified library. If no library name is specified, all libraries are listed.

NAME
specifies the library name to be listed

name
indicates the library name

NAME
specifies the library name to be defined or modified.

name
indicates the library name. The following rules apply:

- The first character must be an alpha character.
- The last character must be either an alpha character or digit.
- Any character between the first and last must be either an alpha character, digit, or hyphen.

ENable
enables the specified library to be selected for allocation or mount requests.

DISable
disables the specified library. The library is not used for any allocation or mount requests.

LOCSUBsys
specifies a local HSC and/or MVS/CSC subsystem

subsysname
indicates the subsystem

LOCENable
enables the specified HSC or MVS/CSC subsystem to be used as a local path to the library.

LOCDISable
disables the specified HSC or MVS/CSC subsystem. LOCDISable can be used to force the SMC to ignore the local subsystem and access the library via a remote server.

WTORdest

optionally, specifies the console to which HSC WTOR messages are directed. See “Managing HSC Mounts from the SMC Client” on page 115 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.



Note: If WTORdest Client is specified but SMC is communicating with a local server, this parameter is ignored.

Server

WTOR messages are issued on the HSC server console. This is the default.

SERVerlist

lists defined libraries, their attributes and associated servers. The SERVerlist parameter may also be specified with the NAME parameter to limit the display to a single library.

NAME

specifies the library name for which servers are to be listed

name

indicates the library name

Example

In the following example, the LIBRARY command defines a local HSC subsystem executing on the same host as the SMC.

```
LIBRARY NAME(DENVER) LOCSUB(HSC0)
```

In the following example, the LIBRARY command defines a local MVS/CSC subsystem executing on the same host as the SMC.

```
LIBRARY NAME(CSCLIB) LOCSUB(CSCA)
```

LIBrary

In the following example, the LIBrary command specifies that the remote subsystem (OMAHA) return selected WTORs to the SMC client without the remote HSC issuing the WTOR message.

```
LIBRARY NAME(OMAHA) WTORDEST(CLIENT)
SERVER NAME(OMAHA1) LIBRARY(OMAHA) HOST(OMA1) PORT(8888)
SERVER NAME(OMAHA2) LIBRARY(OMAHA) IP(11.22.33.44) +
    PORT(7777)
```

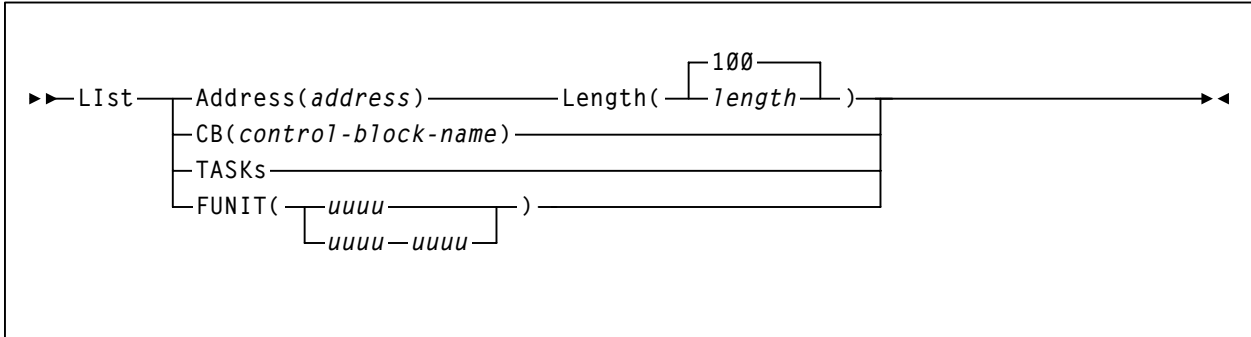


Note: The SERVer command must be specified to define the path to the remote library.

Llist Command

The SMC Llist command is used to display storage accessible from the SMC started task address space.

Syntax



Command Name

Llist

initiates the Llist command.

Parameter Descriptions

Address

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

optionally, specifies the length of SMC memory (bytes) to be listed.

length

indicates the length. 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

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 Sun Microsystems StorageTek Software Support.

List

TASKs

Lists the tasks in the SMC subsystem address space.

FUNIT

Lists SMC control blocks associated with the specified unit address(es).

uuuu

indicates the unit address(es).

Example

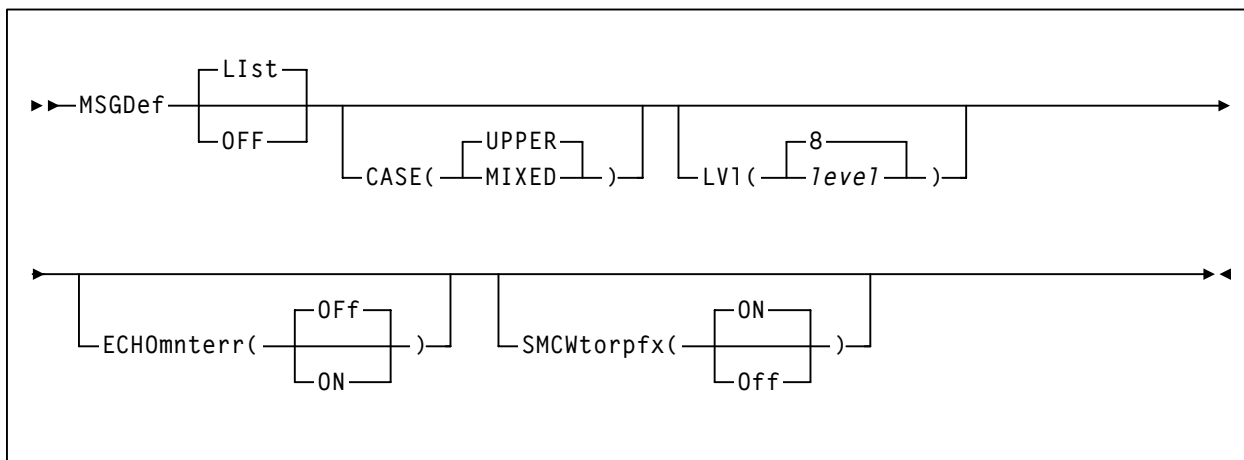
In the following example, the Llst command displays the given length (FFF) of accessible storage beginning with address 01FE00.

```
LIST A=01FE00 L=FFF
```

MSGDef Command

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

Syntax



Command Name

MSGDef

initiates the MSGDef command.

Parameter Descriptions

List

optionally, lists current default SMC message settings.

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

OFF

optionally, resets all MSGDef values to original SMC default settings. It **cannot** be specified with any other parameter.

CASE

optionally, specifies the message case. Valid values are UPPER or MIXED. If no value is specified, the default value UPPER is used.

LVI

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

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 value if the MSGDef parameter is not specified.

If no value is specified, the default value of 8 is used.



Note: Levels higher than 8 are used for diagnostic purposes and should **only** be specified as directed by Sun Microsystems 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 directly to the console for the SMC client.



Note: This parameter is only valid for remote servers.

Off

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

SMCWtorpfx

optionally, specifies whether the SMC 0137 prefix is displayed for HSC WTOR messages generated by WTORDEST(CLIENT) on the LIBRARY command.

ON

The SMC 0137 prefix is displayed for HSC WTOR messages. This is the default.

Off

The SMC 0137 prefix is not displayed for HSC WTOR messages.

Example

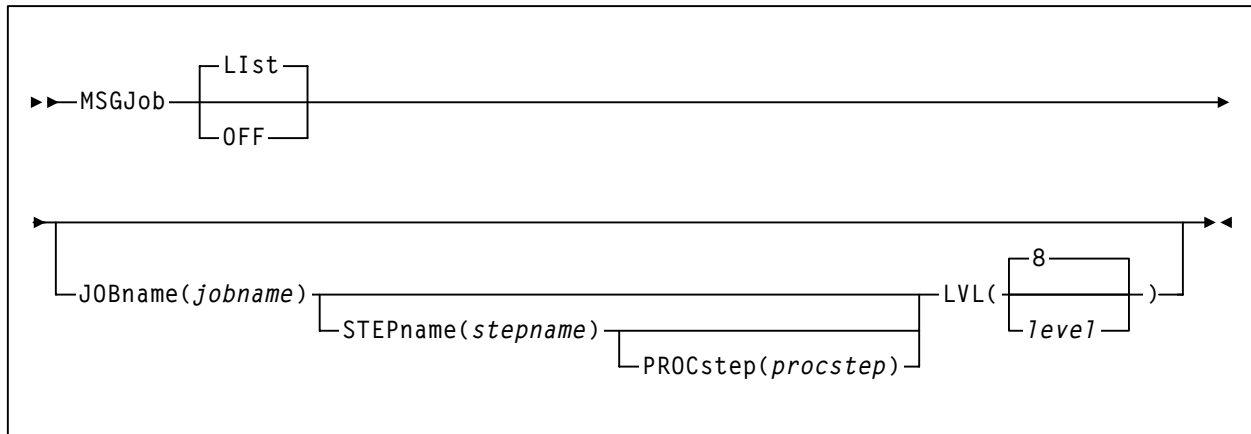
In the following example, the MSGDef command specifies that SMC messages are displayed in mixed case. The command also specifies that **only** error and warning messages from the SMC subsystem are displayed.

```
MSGDef CASE(MIXED) LVL(4)
```

MSGJob Command

The SMC MSGJob command allows an override of the default message level by job name, step name, and PROC step.

Syntax



Command Name

MSGJob

initiates the MSGJob command.

Parameter Descriptions

List

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

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

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.

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

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

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

LVL

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.

If no value is specified, the default value of 8 is used.



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

Example

In the following example, the MSGJob command specifies that **only** error and warning messages are displayed for job PRODJOB1, step STEP3, PROC step REPORT.

```
MSGJOB JOB(PRODJOB1) STEP(STEP3) PROC(REPORT) LVL(4)
```

READ

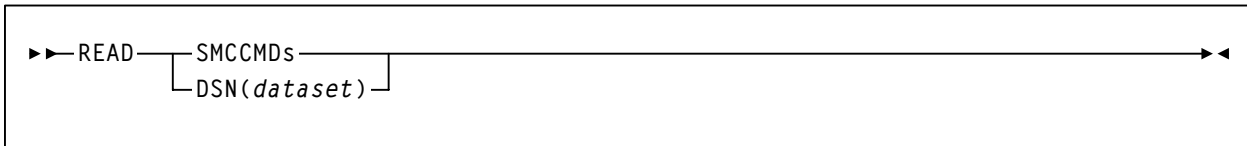
READ Command

The SMC READ command is used to enter a series of commands using an input data set instead of console commands.



Note: This command can **only** be issued from the console.

Syntax



Command Name

READ

initiates the READ command.

Parameter Descriptions

SMCCMDs

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

DSN

processes commands contained in the specified data set.

dataset

indicates the specified data set.



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

Example

In the following example, the READ command is used to process commands included in the MYCMDS member of the MY.COMMAND.PDS data set.

```
READ DSN('MY.COMMAND.PDS(MYCMDS)')
```

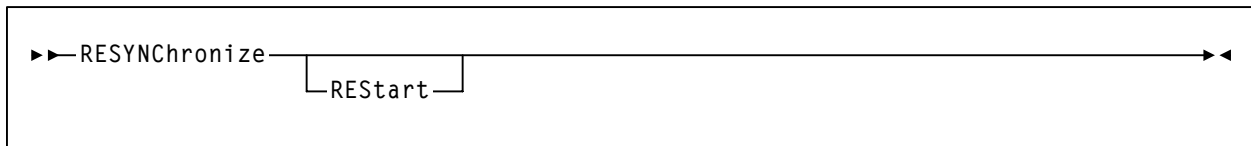

RESYNChronize Command

The SMC RESYNChronize command is used to re-establish connections to all defined libraries for the purpose of acquiring policies from all libraries, and to establish library ownership and determine drive type information for each tape drive.

This action is automatically performed when the SMC first activates a new remote path to a library, or whenever an active local path to a library is accessed during an MVS allocation or mount event. However, the operator **must** issue the RESYNChronize command in the following instances:

- When ALLOC or UNITDEF commands are issued on a remote HSC library
- 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 HSC or MVS/CSC library, 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.

Syntax



Command Name

RESYNChronize
initiates the RESYNChronize command.

Parameter Descriptions

REStart
starts the RESYNChronize attempt at the first SERVER, regardless of the last active path. REStart also attempts to redrive any outstanding mounts.

Notes:

- The SMC RESYNChronize command is not required when communicating with a local HSC, but may be used to redrive outstanding mounts that may have failed due to hardware or communication errors with **either** a local or remote HSC.
- Use of the RESYNChronize command to automate pending mounts for newly active HSC or MVS/CSC systems requires that the corresponding HSC or MVS/CSC systems be started with the AMPND startup parameter.
- Use of the RESYNChronize command to automate pending mounts for newly active HSC systems should be issued after the HSC system has reached FULL service level.

SERVer

SERVer Command

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

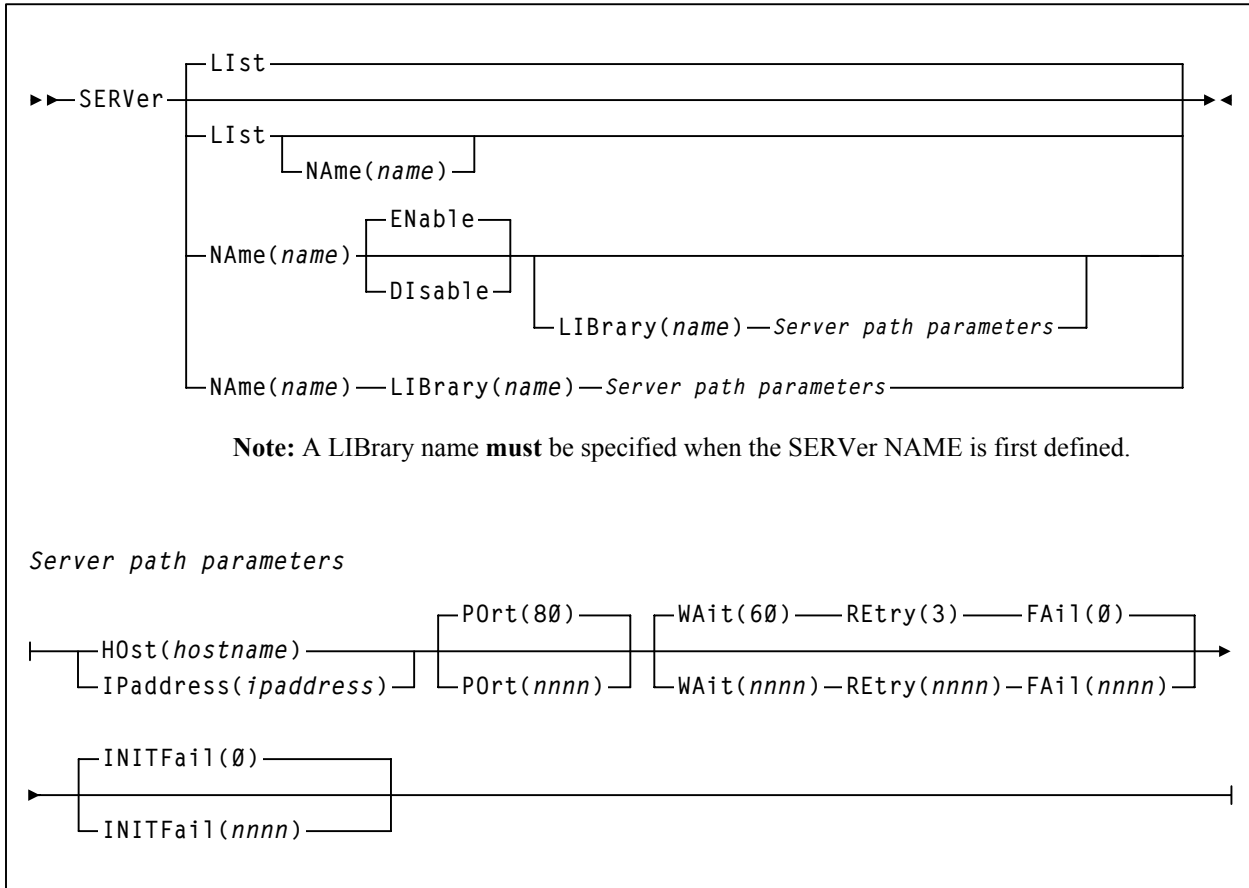


Notes:

- Before a SERVer is defined, the LIBRARY that it references must be defined using a LIBRARY command. The library name associated with a SERVer **cannot** be changed. See “LIBRARY Command” on page 129 for more information.
- You **cannot** define a SERVER if the corresponding LIBRARY defines a LOCSUBSYS for an MVS/CSC subsystem. For example, the following is **not** valid (assuming the subsystem CSC1 is an MVS/CSC system).

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

Syntax



Command Name**SERVer**

initiates the SERVer command.

Parameter Descriptions**List**

optionally, displays status information for all server library server paths.

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

Name

optionally, specifies a server path for which status is displayed.

name

indicates the server path

Name

optionally, specifies the communication path or route to the server.

name

indicates an identifier for the path parameters. This name is reported in any communication error messages. The following rules apply:

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

ENable

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 library, the library is unavailable for allocation or mount requests.

LIBrary

optionally, specifies the library name associated with an actual ACS hardware configuration. The LIBrary parameter **must** be specified when a new server is defined.

name

indicates the library name. This name is reported in any library error messages. The following rules apply:

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



Note: If you are using the StorageTek HTTP server, there can be multiple NAMED paths to a single LIBRARY configuration. See Chapter 2, “SMC and StorageTek Library Management” for more information.

HOst

optionally, specifies the host on which the subsystem resides.

hostname

indicates the host name. The following arpanet rules apply:

- The first character must be an alpha character.
- The last character must be either an alpha character or digit.
- Any character between the first and last must be either an alpha character, digit, hyphen, or period.
- The host name can be a maximum of 128 characters.

IPaddress

optionally, specifies the subsystem IP address

ipaddress

indicates the IP address

POrt

optionally, specifies the server port.

nnnn

indicates the server port. The default is 80.

WAIt

optionally, specifies the maximum wait time for any single request made over the network.

nnnn

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



Note: The default for a mount or dismount request is 10 minutes (600 seconds) or more if the specified WAIT time is greater than 600 seconds.

REtry

optionally, specifies the number of retries to be attempted 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.

FAil

optionally, specifies the number of failures (**after** communication is established) before the server path is disabled or placed out of service.



Note: The FAil limit count **only** applies **after** successful communication has been established on this SERVER path.

nnnn

indicates the number of failures. The default is 0.

If 0 is specified, the named SERVER is never automatically disabled due to communications errors. Specify this value when there are no backup SERVER paths to a named LIBRARY.

After successful communication is established with a LIBRARY, the error severity and SERVER FAil count are used to determine whether a SERVER path should be automatically disabled.

INITFail

optionally, specifies the number of failures (**before** communication is established) before the server path is disabled or placed out of service.



Note: The INITFail limit count **only** applies **before** successful communication has been established on this SERVER path.

nnnn

indicates the number of failures. The default is 0.

If 0 is specified, the named SERVER is never automatically disabled due to communications errors.

Examples

In the following example, two libraries (DENVER and OMAHA) are defined

The first library, DENVER, has both a local HSC (HSC0, on the SMC host) and a backup remote path on MVS system DEN1. SMC uses the local path unless the local subsystem is inactive or is disabled using the command LIB NAME(DENVER) LOCDIS. If the backup remote path is used, selected HSC messages are displayed on the client host.

```
LIBRARY NAME(DENVER) LOCSUBSYS(HSC0) WTORDEST(CLIENT)
SERVER NAME(DENBKUP) LIBRARY(DENVER) HOST(DEN1) PORT(8888)
```

The second library, OMAHA, is always accessed using a remote server. The SMC attempts to use the first defined server, OMAHA1. If this server becomes inactive, inaccessible, or is disabled, the SMC uses the next defined server, OMAHA2. If the SMC is unable to communicate with either server, no library requests are directed to the OMAHA library.

```
LIBRARY NAME(OMAHA) WTORDEST(CLIENT)
SERVER NAME(OMAHA1) LIBRARY(OMAHA) HOST(OMA1) PORT(8888)
SERVER NAME(OMAHA2) LIBRARY(OMAHA) IP(11.22.33.44) +
PORT(7777)
```

MVS hosts DEN1, OMA1, and the host represented by the IP address 11.22.33.44 must each have an instance of the StorageTek HTTP server executing on them in order for SMC to communicate with the HSC on these hosts. The StorageTek HTTP server port must be defined as 8888 for the first two hosts and 7777 for the third host.

SMSDef Command

The SMSDef command is used to tailor default SMC DFSMS support.

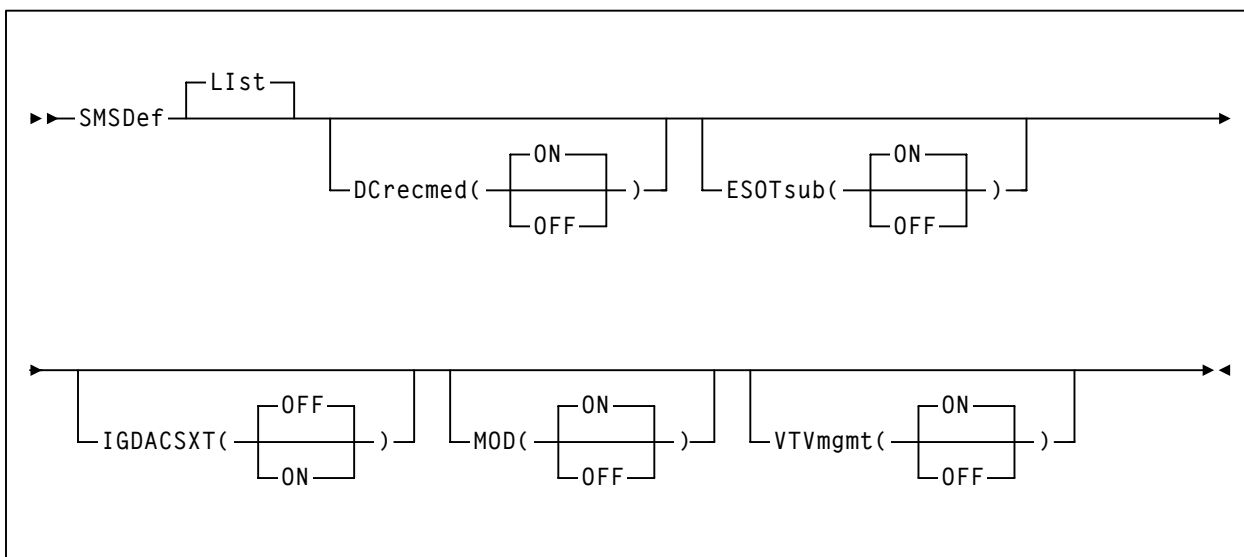


Note: This command takes effect when SMC DFSMS support is activated by setting the ALLOCDef or ALLOCJob SMS parameter value to ON.

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.

- DFSMS data class constructs specify a limited set of recording techniques and media supported by Sun Microsystems. Conversely, TAPEREQ statements can specify all recording techniques and media supported by Sun Microsystems. By default, DFSMS data class construct recording technique and media values override the TAPEREQ values. However, by setting the SMSDef DCRecmed parameter to OFF, the TAPEREQ recording technique and media values are always used.
- If VTCS is installed in your environment but TAPEREQ statements are used to assign a management class instead of DFSMS ACS routines, the calls to the DFSMS ACS routines can be bypassed by setting the SMSDef VTVmgmt parameter to OFF.
- If your environment does not support esoteric substitution using DFSMS storage group assignment, these DFSMS ACS routine calls can be bypassed by setting the SMSDef ESOTsub parameter to OFF.
- If your environment does not support DFSMS functions, the ALLOCDef SMS parameter can be set to or allowed to default to OFF. The SMC bypasses all DFSMS ACS routine calls.

Syntax



Command Name

SMSDef

initiates the SMSDef command.

Parameter Descriptions

List

optionally, lists the current SMS settings.

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

DCrecmed

specifies whether the SMC requests the data class construct be returned when a data class is assigned either during DFSMS ACS data class routine invocation or by using the DATACLAS parameter on the DD statement. The data class construct contains the recording technique and media value of the associated data class.

ON

the SMC retrieves the data class construct of a data class specified via JCL or DFSMS ACS data class routine. The recording technique and media, if defined to the data class, are retrieved during allocation and mount processing.

OFF

the SMC does not retrieve the data class construct.



Note: If DCrecmed is set to OFF, the DFSMS ACS data class routine is still retrieved from the DFSMS ACS data class routine or from the DATACLAS JCL parameter if the ESOTsub parameter or VTVmgmt parameter is set to ON.

ESOTsub

specifies whether the SMC invokes the DFSMS ACS routines during IDAX (Interpreter/Dynamic Allocation Exit) processing. The IDAX DFSMS interface can be used to provide a different unit name for new allocations.

ON

the SMC invokes all four (data class, storage class, management class and storage group) DFSMS ACS routines during IDAX processing.

OFF

the SMC does not invoke any DFSMS ACS routines during IDAX processing.

IGDACSXT

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 manual *DFSMS Installation Exits* for more information about this exit.

ON

the SMC invokes exit IGDACSXT before calling 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 SMC does not invoke IGDACSXT before calling the DFSMS ACS routines.

MOD

optionally, specifies whether SMS esoteric substitution (if specified) applies to DISP=MOD data sets as well as DISP=NEW data sets.

OFF

The SMC SMS esoteric substitution (if specified) does not apply to DISP=MOD data sets. This is the default value.

ON

The SMC SMS esoteric substitution (if specified) applies 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 that during SMS IDAX substitution, the SMC **cannot** determine whether a MOD data set exists.

VTVmgmt

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

when SMC determines that a list of eligible drives contain a virtual volume or a mount request specifies a virtual drive, the SMC invokes the data class, storage class and management class DFSMS ACS routines. If a management class name is returned, it is provided to VTCS during volume lookup and volume mount processing.

OFF

the SMC does not invoke the DFSMS ACS storage class or management class routines during virtual volume allocation or virtual volume mount processing.



Note: The DFSMS ACS data class routine is still invoked if the DCCredmed parameter value is set to ON.

SMSDef

Examples

In the following example, the SMSDef command is issued to disable the esoteric substitution function during IDAX processing.

```
SMSDEF ESOTSUB(OFF)
```

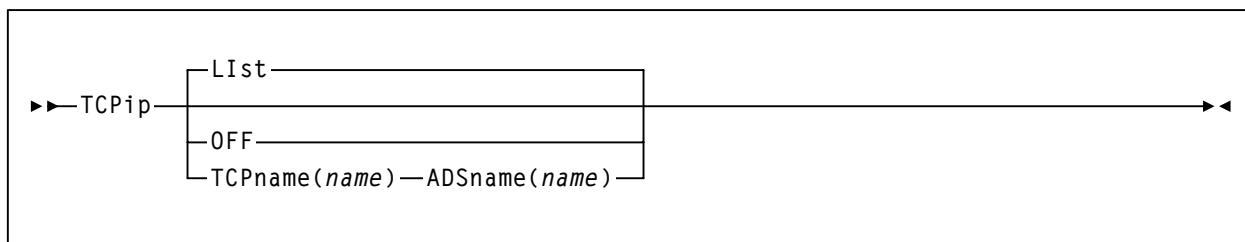
In the next example, the SMSDef command is used to bypass all DFSMS ACS routine calls during virtual volume allocation and virtual volume mount processing. Setting VTVMgmt to OFF bypasses the storage class and management class ACS routines. Setting DCrcmed to OFF bypasses the data class ACS routine.

```
SMSDEF DCRECMED(OFF) VTVMGMT(OFF)
```

TCPip Command

The SMC TCPip command alters or lists the current TCPNAME and ADSNAME defaults for your TCP/IP communications environment. It allows you to direct TCP/IP requests to a specific TCP/IP stack on an MVS host. The TCPip command can be issued at any time.

Syntax



Command Name

TCPip
initiates the TCPip command.

Parameter Descriptions

LIst
optionally, lists current TCPNAME and ADSNAME TCP/IP defaults.

OFF
optionally, specifies that system defaults are used for TCP/IP communications.

TCPname
optionally, specifies the TCPNAME value for a specific TCP/IP stack on an MVS host.

name
indicates the TCPNAME value

ADSname
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.

TRace

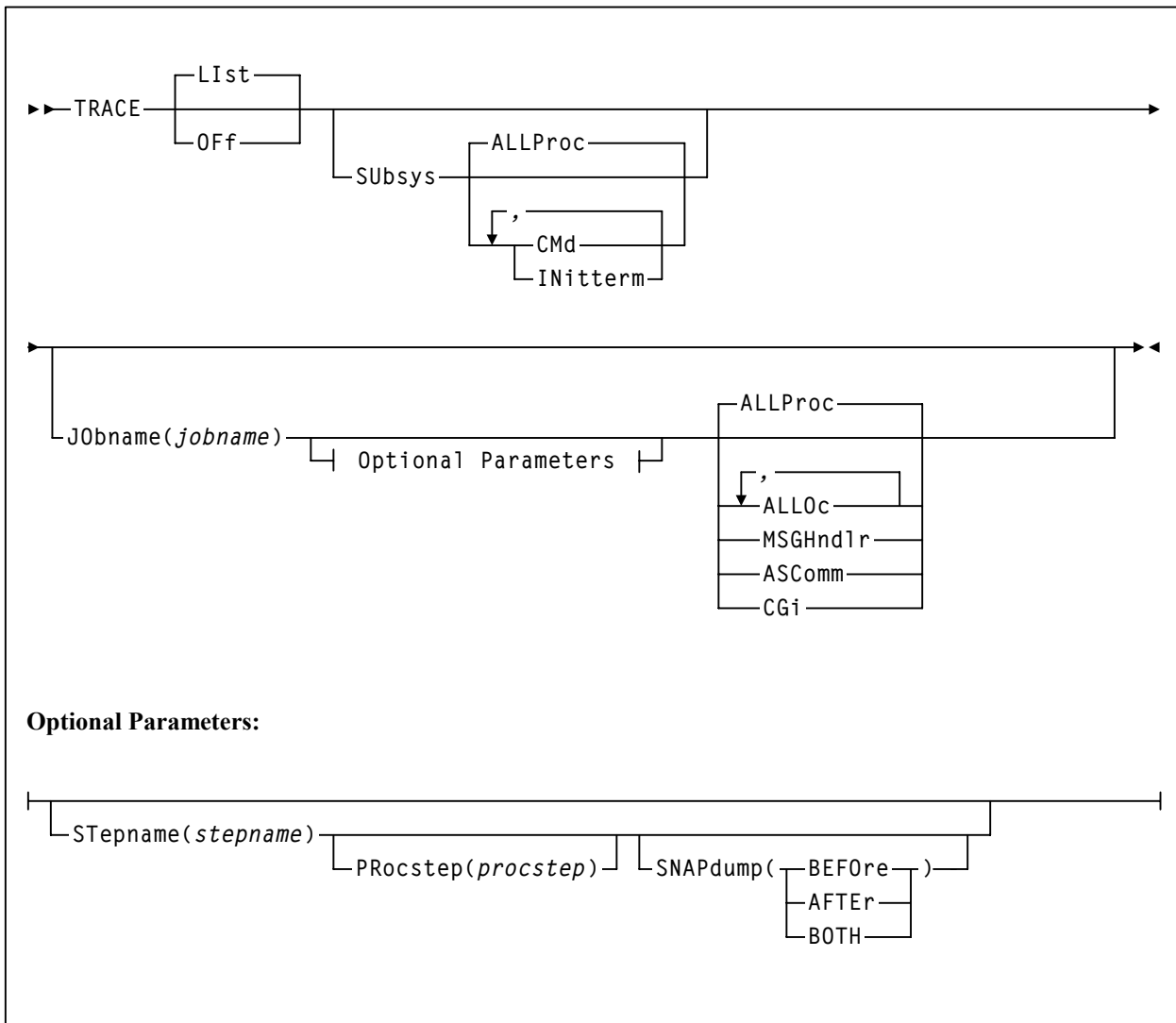
TRace Command

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.



Warning: This command may impact system performance and should be used **only** as directed by Sun Microsystems StorageTek Software Support.

Syntax



Command Name**TRace**

initiates the TRace command.

Parameter Descriptions**LlSt**

optionally, lists current SMC trace settings for both the SMC subsystems and SMC components by job name, step name, and PROC step.

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. In either case, all SMC subsystems being traced are listed.

This is the default if no parameters are specified.

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.

SUBsys

optionally, enables or disables SMC tracing for one or more SMC subtasks.

ALLProc

enables tracing for all SMC subtasks.

OFF

disables tracing for all SMC subtasks.

CMd

enables tracing for the SMC subtask that processes all SMC operator commands.

INitterm

enables tracing for the SMC subtask that performs SMC initialization and termination.

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, all jobs whose job names match the characters preceding the asterisk are traced or listed. To trace all jobs, specify TRACE JOB(*).

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. It is **not** valid in JES3 environments with TAPE SETUP processing.

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. It is **not** valid in JES3 environments with TAPE SETUP processing.

SNAPdump

optionally, requests a snap dump. This parameter is **not** valid in JES3 environments with TAPE SETUP processing.



Note: A snap dump may impact system performance, and should be used **only** as directed by Sun Microsystems StorageTek Software Support.

BEFORe

perform a snap dump before allocation influencing.

AFTEr

perform a snap dump after allocation influencing.

BOTH

perform a snap dump both before and after allocation influencing.

ALLProc

enables tracing for all SMC components. This is the default if no parameters are specified.

ALLOc

optionally, enables tracing for the SMC allocation component that performs drive exclusion and drive prioritization.

MSGHndlr

optionally, enables tracing for the SMC message handler component that processes all MOUNT, DISMOUNT, and SWAP messages and issues robotic movement commands to the StorageTek tape library.

ASComm

optionally, enables tracing for the SMC address space communication component that transfers library requests and configuration information between address spaces.

CGi

optionally, enables tracing for the StorageTek HTTP server component for requests originating on this host.

Examples

In the following example, tracing is enabled for any job whose name begins with “TESTJOB”.

```
TRACE JOB=TESTJOB*
```

In the following example, the TRace command specifies that tracing is disabled for all SMC components and subsystems.

```
TR OFF
```

In the following example, tracing is enabled for SMC allocation and message handling components for any job with jobname=ABC.

```
TR JOBN(ABC) ALLOC MSGHN
```

In the following example, tracing is enabled for the SMC command subsystem.

```
TR SUBSYS CMD
```

TRace

In the following example, tracing is disabled for all SMC subsystems.

```
TR SUBSYS OFF
```

In the following example, tracing is disabled for all jobs whose job name begins with “PROD”.

```
TR OFF JOB(PROD*)
```


UExIt Command

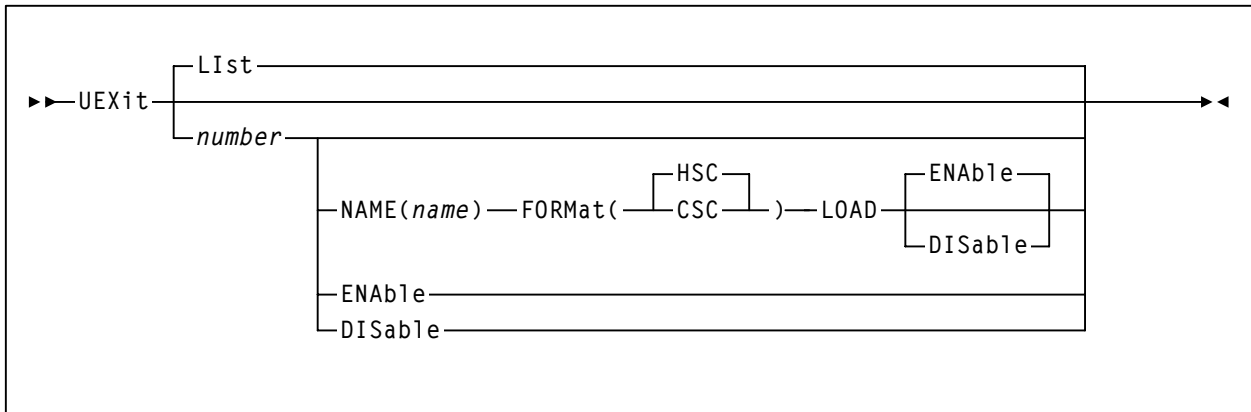
User exits 1, 2, 4, 8, 9, 10, 11, 12, and 13 are invoked and managed by the SMC. The UExIt command defines which load module is invoked for a specific user exit, and whether the user exit is enabled or disabled. Both HSC and MVS/CSC user exit formats are supported. The UExIt command specifies the desired format.



Notes:

- 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 version of a given user exit can be invoked, regardless of the number of libraries the SMC interacts with. The desired version (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 (SMC STEPLIB, JOBLIB, or your system's LINKLIB concatenation)
- Refer to the *NCS User Exit Guide* for information about each user exit.
- Sun Microsystems recommends that you use the HSC user exit formats, as support for MVS/CSC user exit formats will end in a future SMC release.

Syntax



Command Name

UExit

Initiates the UExit command.

Parameter Descriptions

List

optionally, displays status information for all user exits. This is the default if no parameters are specified.

number

optionally, specifies the user exit number. Valid values are 1, 2, 4, 8, 9, 10, 11, 12, and 13.

NAME

optionally, specifies the user exit's load module name

name

indicates the load module name (e.g. SLSUX01)

FORMat

optionally, specifies the format of user exit to be enabled or disabled.

HSC

indicates the HSC user exit format.

CSC

indicates the MVS/CSC user exit format.

LOAD

optionally, loads the specified user exit, making it available for use.



Notes:

- 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.

Example

In the following example, the UExit command is used to load the HSC format (SLSUX01) of user exit 01.

```
UEXIT 01 NAME(SLSUX01) FORMAT(HSC) LOAD
```

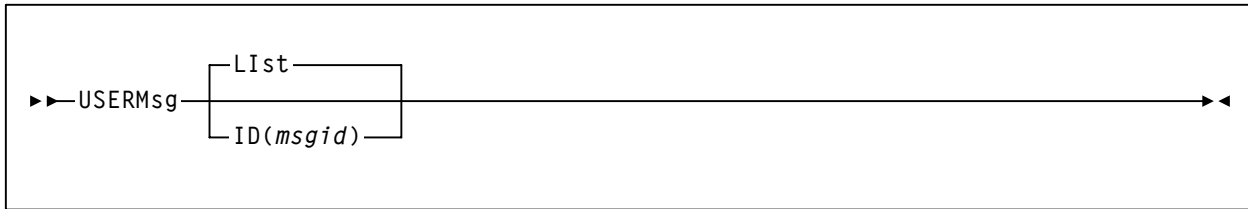
SLSUX01 is loaded from the SMC STEPLIB, JOBLIB, or your system's linklib concatenation.

USERMsg

USERMsg Command

The USERMsg command allows you to specify additional message identifiers for messages to be intercepted and passed to the library subsystem's user exit 01. This command can also be used to list the message identifiers of those messages that have been defined using the USERMsg command.

Syntax



Command Name

USERMsg
initiates the USERMsg command.

Parameter Descriptions

LIst

optionally, lists the message identifiers in the customer-defined message table. This is the default if no parameters are specified.

ID

optionally, specifies a message to be added to the customer-defined message table. This message is passed to any active library subsystem user exit 01 (SLSUX01/SCSUX01).



Note: This parameter may **only** be specified in the SMCPARMS data set.

msgid

indicates the message identifier for the message to be added.

Example

In the following example, the USERMsg command is used to list message identifiers for messages included in the customer-defined message table.

```
USERMsg LI
```

Chapter 8. Recovery Procedures

Overview

This chapter describes recovery procedures used when the SMC, a library subsystem (HSC or MVS/CSC), or JES3 becomes inactive. Procedures for both JES2 and JES3 are included.

SMC Recovery Procedures (JES2)

This section describes recovery procedures for the following problem scenarios:

- Inactive SMC - Active library subsystem
- Active SMC - Inactive library subsystem
- Automating mount requests for inactive libraries
- Lost MVS mount requests for active libraries

Inactive SMC - Active Library Subsystem

When the SMC fails while a library subsystem remains active, the following functions are **not** performed:

- allocation processing
- automation of mount/dismount/swap messages

When this occurs, re-start the SMC.

Certain software products allow you to suspend processing that may require dynamic allocation. For example, if Data Facility Hierarchical Storage Manager (DFHSM) is installed on the local processor, you can issue commands to prevent this type of processing without stopping DFHSM.

Common allocations can be postponed by holding the job queue or purging all initiators. Refer to the appropriate IBM publication for more information about JES2 operator commands.

If the HSC and MVS/CSC were started with the AMPND parameter, outstanding mount messages are re-driven. Refer to the *MVS/CSC Configuration Guide* for more information about the AMPND parameter.

Active SMC - Inactive Library Subsystem

When a library subsystem fails or is terminated, volumes and drives owned by that subsystem become unknown to SMC. The following functions are **not** performed:

- allocation processing
- automated mount processing

When this occurs, re-start the library subsystem. If AMPND is specified, The SMC will automatically re-establish communication with the library and automate any outstanding mounts. See “Automating Mount Requests for Inactive Libraries” below for more information.

Certain software products allow you to suspend processing that may require dynamic allocation. For example, if Data Facility Hierarchical Storage Manager (DFHSM) is installed on the local processor, you can issue commands to prevent this type of processing without stopping DFHSM.

Common allocations can be postponed by holding the job queue or purging all initiators. Refer to the appropriate IBM publication for more information about JES2 operator commands.



Note: You can provide a backup path to a remote library that is automatically activated when the local HSC is discovered to be inactive.

Automating Mount Requests for Inactive Libraries

MVS mount requests for drives owned by inactive library subsystems can be automatically redriven when the corresponding HSC or MVS/CSC library subsystems are activated. The SMC redrives pending mount requests when the corresponding HSC or MVS/CSC AMPND startup parameter is in effect for those subsystems. This requires that the SMC process a new MVS allocation or mount event. Alternatively, the SMC RESYNChronize command can be used to force a connection to the newly active library subsystem and redrive the pending mounts immediately.

Lost MVS Mount Requests for Active Libraries

An MVS mount request may be lost when an LMU error occurs. Use this procedure if you suspect lost mounts.

1. Issue the following MVS command on the system requesting the mount to determine if any drives have a mount request pending:

```
D R,L
```

2. Issue the following MVS command on the same system to determine which VOLSER to mount:

```
D U,,,uuuu,1
```

3. If the drive is defined to an HSC library, issue the HSC Mount command for the volume on the MVS system on which the HSC is active.

SMC Recovery Procedures (JES3)

This section describes recovery procedures for the following problem scenarios:

- Inactive SMC - Active library subsystem
- Active SMC - Inactive library subsystem
- Inactive JES3 on a local processor
- Inactive JES3 on a global processor
- Automating mount requests for inactive libraries
- Lost JES3 mount requests for active libraries
- Lost MVS mount requests for active libraries

Inactive SMC - Active Library Subsystem

When the SMC fails while a library subsystem (HSC and/or MVS/CSC) remains active, the following functions are **not** performed:

- allocation processing
- automation of mount/dismount/swap messages

When this occurs, re-start the SMC.

Certain software products allow you to suspend processing that may require dynamic allocation. For example, if Data Facility Hierarchical Storage Manager (DFHSM) is installed on the local processor, you can issue commands to prevent this type of processing without stopping DFHSM.

To postpone the C/I process for batch jobs while SMC is inactive, use the following modify command:

```
*F X,D=POSTSCAN,MC=00
```

After the SMC is re-started, restore the maximum count to its original value, xx:

```
*F X,D=POSTSCAN,MC=xx
```

If the HSC and MVS/CSC were started with the AMPND startup parameter, outstanding mount messages are re-driven when SMC is re-started and an MVS allocation or mount event occurs. Alternatively, the SMC RESYNChronize command may be issued to redrive pending mounts under these circumstances.

Active SMC - Inactive Library Subsystem

When a library subsystem fails or is terminated, volumes and drives owned by that subsystem become unknown to SMC. The following functions are **not** performed:

- allocation processing
- automated mount processing

When this occurs, re-start the library subsystem. If AMPND is specified, The SMC will automatically re-establish communication with the library and automate any outstanding mounts. See “Automating Mount Requests for Inactive Libraries” on page 164 for more information.

Certain software products allow you to suspend processing that may require dynamic allocation. For example, if Data Facility Hierarchical Storage Manager (DFHSM) is installed on the local processor, you can issue commands to prevent this type of processing without stopping DFHSM.



Note: You can provide a backup path to a remote library that is automatically activated when the local HSC is discovered to be inactive.

Inactive JES3 on a Local Processor

When JES3 fails on a local processor, jobs executing at the time that do not require JES3 services continue to execute. Drive exclusion still occurs for dynamic allocation requests.

To recover, restart JES3 (LOCAL start).

The SMC continues processing and requires no recovery.

Inactive JES3 on a Global Processor

When JES3 fails on a global processor, jobs executing at the time that do not require JES3 services continue to execute. Drive exclusion still occurs for dynamic allocation requests.

To recover, restart JES3 or invoke Dynamic System Interchange (DSI) processing.

You can use DSI to reassign the JES3 global function to a JES3 local processor when the global processor becomes inactive or requires maintenance. One of the JES3 local processors becomes the new JES3 global processor. By reassigning the global function to a local processor, the JES3 environment continues processing. The SMC continues processing and requires no recovery.

Refer to the *HSC System Programmer's Guide* or *MVS/CSC System Programmer's Guide* for more information about cross host recovery.

Automating Mount Requests for Inactive Libraries

MVS mount requests for drives owned by inactive library subsystems can be automatically redriven when the corresponding HSC or MVS/CSC library subsystems are activated. The SMC redrives pending mount requests when the corresponding HSC or MVS/CSC AMPND startup parameter is in effect for those subsystems. This requires that the SMC process a new MVS allocation or mount event. Alternatively, the SMC RESYNChronize command can be used to force a connection to the newly active library subsystem and redrive the pending mounts immediately.

Lost JES3 Mount Requests for Active Libraries

A JES3 mount request may be lost when an LMU error occurs. Use this procedure if you suspect lost mounts.

1. Issue the following JES3 command to determine which jobs are awaiting a volume mount:

```
*I,S,V
```

2. Issue the following JES3 command to determine how long a job has been waiting:

```
*I,J=nnnn,W
```

3. Issue the following JES3 command to determine the volume and drive the job is waiting on:

```
*CALL,DISPLAY,J=nnnn
```

4. If the drive with a pending mount is defined to an HSC library, issue the HSC Mount command for the volume on the MVS system on which the HSC is active.

Lost MVS Mount Requests for Active Libraries

An MVS mount request may be lost when an LMU error occurs. Use this procedure if you suspect lost mounts.

1. Issue the following MVS command on the system requesting the mount to determine if any drives have a mount request pending:

```
D R,L
```

2. Issue the following MVS command on the same system to determine which VOLSER to mount:

```
D U,,,uuuu,1
```

3. If the drive is defined to an HSC library, issue the HSC Mount command for the volume on the MVS system on which the HSC is active.

Chapter 9. SMC Utilities

Overview

This chapter provides information for the following SMC utilities:

- Tape Management Extract utility, used to support scratch synchronization between the SMC client host and the HSC CDS.
- JES3 Configuration Report utility, used to generate JES3 configuration data which can be used to help verify JES3 initialization deck behavior.

SMC Utility Return Codes and Messages

SMC utilities may return the following return codes:

- A return code of 0 is returned when the utility program completes successfully with no errors or warnings.
- A return code of 4 is returned when an inconsistency is detected.
- A return code of 8 is returned when an error that prevents the utility from completing is detected.

For a listing of SMC messages that may be returned by SMC utilities, see page 226. These messages are identified by the SMCU prefix.

Tape Management Extract Utility

The SMC utility SMCUDBX supports scratch synchronization between the SMC client host and the HSC CDS. Using common tape extract modules (SLUDRCA1, SLUDRTL, SLUDRRMM, and SLUDRZAR) with HSC and MVS/CSC, the SMC accesses the client tape management system to produce an extract file that can be read as input by the HSC scratch tape conversion module SLUCONDB. This process is illustrated in the following figure:

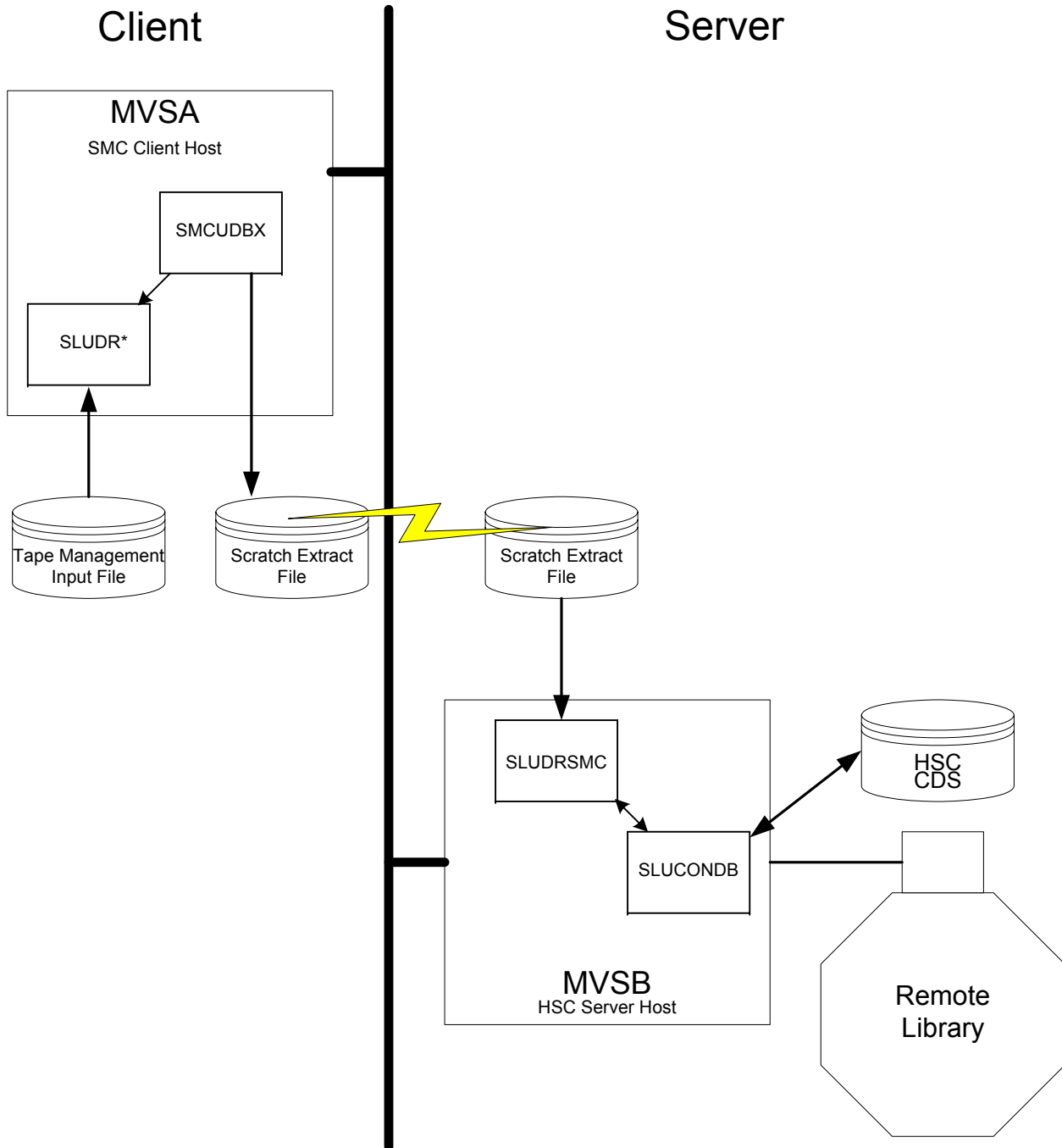


Figure 15. Scratch Synchronization Between SMC Client host and HSC CDS

If the client and server share a tape management system, the HSC utility SLUCONDB may be run directly using the common tape management system data base as input.

If, however, the client host has a tape management system that is not accessible to the HSC host, the SMC utility SMCUDBX can be used to extract scratch volumes from the tape management system using label and date parameters. The SMCUDBX 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 SMCUDBX 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.

SMCUDBX 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, SMCUDBX calls the appropriate tape management system database READ routine. For Zara, SMCUDBX invokes the SLUDRZAR module, 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.

The following list describes the READ routines that are called for the CA-1, CA-DYNAM/TLMS, DFSMSrmm, and Zara tape management systems.

SLUDRCA1

Called to read the CA-1 TMC when CA-1 is defined as the tape management system. This routine uses CA-1 macros to map the layout of the TMC.

SLUDRTLMS

Called to read the CA-DYNAM/TLMS VMF when CA-DYNAM/TLMS is defined as the tape management system. This routine uses a CA-DYNAM/TLMS COPY member to map the layout of the VMF.

SLUDRRMM

Called to read the DFSMSrmm report extract file when DFSMSrmm is defined as the tape management system. This routine uses the DFSMSrmm EDGRVEXT macro to map the layout of the report extract file.

SLUDRZAR

Called to invoke an API to read Zara's database when Zara is defined as the tape management system.



Note: Both source and load module versions of the SLUDR* tape management extract routines are provided as part of the HSC product installation. Source modules are provided in the HSC SAMPLIB file and load modules are provided in the HSC SLULINK distribution library.

If your installation uses a different tape management system, you can develop a program modeled on one of the supplied examples SLUDRCA1, SLUDRTL, SLUDRRMM, or SLUDRZAR, or you can write your own program to produce an extract file in the following 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'.

SMCUDBX, SLUDRCA1, SLUDRTL, 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 HSC SAMPLIB file on the base installation tape contains the sample source code.



Notes:

- If there are local modifications to your tape management system, you must reassemble the associated READ routine.
- If your tape management system is CA-1 Release 4.9 or below, you must modify and reassemble the SLUDRCA1 routine. See “Using SMCUDBX with CA-DYNAM/TLMS Release 5.3 or Below” for information about modifying and reassembling this routine.
- If your tape management system is CA-DYNAM/TLMS Release 5.3 or below, you must reassemble the SLUDRTL routine. See “Using SMCUDBX with CA-DYNAM/TLMS Release 5.3 or Below” for information about reassembling this routine.
- If your tape management system is Zara, the SMCUDBX 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 SMCUDBX 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 SMCUDBX with CA-1 Release 4.9 or Below

To use SMCUDBX 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.

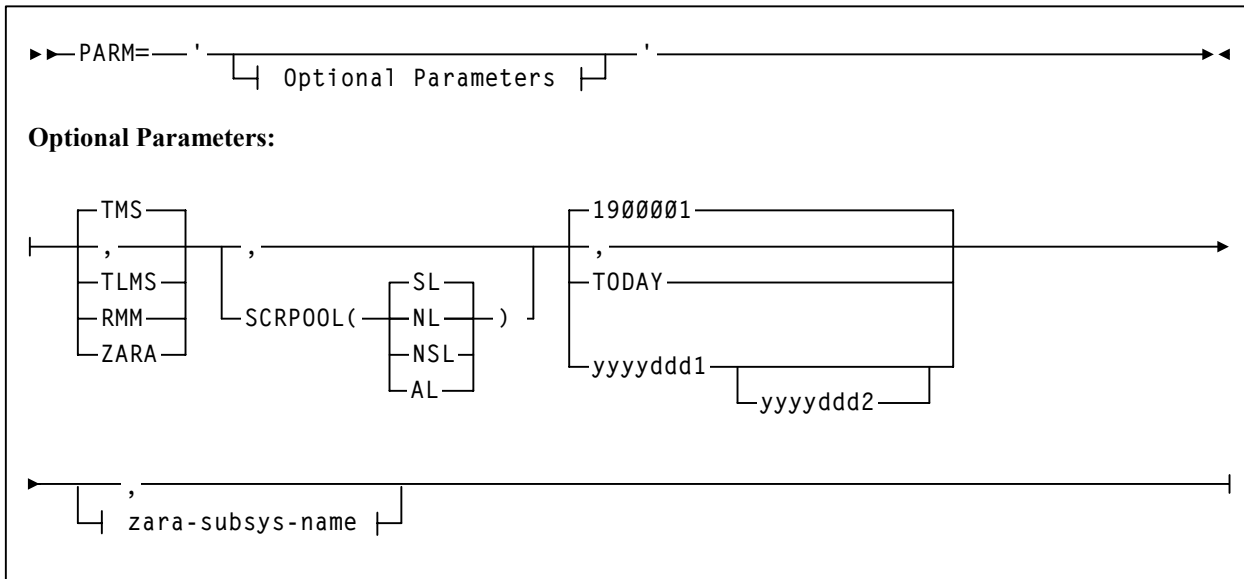
Using SMCUDBX with CA-DYNAM/TLMS Release 5.3 or Below

Perform the following steps to use SMCUDBX with CA-DYNAM/TLMS Release 5.3 or below:

1. Verify that the CA-DYNAM/TLMS Release 5.3 MACLIB containing the VMFBASE MACRO is accessible for the re-assembly of the SLUDRTLTM routine.
2. Reassemble the SLUDRTLTM routine; no modifications to the source code are required.

Specifying Values for the PARM Parameter

You can specify the following values for the PARM parameter on the JCL EXEC statement when running the Scratch Extraction utility.



The keywords for the PARM parameter are positional. If the PARM parameter is not specified, default values are used.

Parameter Descriptions

This section describes the values you can specify for the PARM parameter on the JCL EXEC statement.

TMS

Specifies the CA-1 tape management system.

This is the default value.

TLMS

Specifies the CA-DYNAM/TLMS tape management system.

RMM

Specifies the DFSMSrmm tape management system.

ZARA

Specifies the Zara tape management system.

SCRPOOL

Specifies the label type for the scratch volume selection. Acceptable values are:

SL specifies standard label. This is the default value.

NL specifies non-labeled.

AL specifies ANSI label.

NSL specifies non-standard label.

1900001

Specifies January 1, 1900 as the date. This is the default date.



Note: You must specify a four-year format for year 2000 dates. For example, specify 2000001 for January 1, 2000.

TODAY

Specifies the current system date.

yyyyddd1

Specifies a single date in four-year format. This date becomes the *from* date. The *to* date is the current system date (which is not specified). All dates must be specified in Julian notation (year-day).

yyyyddd1-yyyddd2

Specifies a range of dates in which the volumes became scratch. Neither date specified can be greater than the current system date, and the first date cannot be greater than the second date.



Note: A comma (,) or no keyword (blank) results in the system default date of January 1, 1900 (1900001).

MIXED

Specifies to print all report headings and messages in mixed-case characters (uppercase and lowercase).



Note: A comma (,) or no keyword (blank) results in headings and messages being printed in uppercase characters.

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.

Sample JCL

The following definitions apply to the JCL used for the SMC Scratch Extract utility.

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
           or
//JOB LIB DD DISP=SHR, DSN=your.zaraapi1.library
```

Figure 16 shows sample JCL used to invoke the Scratch Extraction utility to process the CA-DYNAM/TLMS VMF for all tapes scratched in 2003.

```
//yourstd JOB card
//STEP0 EXEC PGM=SMCUBX,PARM='TLMS,,2003001-2003365'
//STEPLIB DD DISP=SHR,DSN=your.SMCLINK
//          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=*
/*
```

Figure 16. Sample JCL for Scratch Extraction Utility

Sample Output

Output resulting from the execution of the Scratch Extraction utility includes:

- Messages associated with error conditions resulting from an unsuccessful attempt to execute scratch extraction processing.
- Messages indicating errors or actions occurring during processing.
- An output data set containing Scratch Update utility control statements.

Figure 17 shows sample output from the SMCUDBX utility.

```
SMCUDBX 6.0.0      SMC TMS DB SCRATCH EXTRACT      Page 0001
TIME 15:26:41                                           Date 2003-08-14

Total records extracted for scratch volumes = 977
```

Figure 17. Scratch Extraction Utility Sample Output-Messages

JES3 Configuration Report Utility

The SMC utility SMCUPJS 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:

- A list of XTYPE numbers in hexadecimal order with their corresponding XTYPE names.
- A list of XTYPEs in hexadecimal order with all esoterics in which the XTYPE appears.
- A 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.
- A 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.
- A 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 JCL sample executes the SMCUPJS utility to produce the five individual reports listed above.

```
//yourstd JOB card
//STEP EXEC PGM=SMCUPJS
//STEPLIB DD DSN=yoursmc.LINKLIB,DISP=SHR
//STDOUT DD DSN=yourstd.printout.dataset,DISP=SHR
```

Figure 18. JCL to Produce the JES3 Configuration Report



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 (6.0.0)          SMC JES3 CONFIGURATION REPORT          PAGE 0001
TIME 15:03:42          XTYPE NUMBER TO XTYPE NAME (EXTON)    DATE 2003-01-09

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 (6.0.0)          SMC JES3 CONFIGURATION REPORT          PAGE 0001
TIME 15:03:42          XTYPE NUMBER TO ESOTERIC (EXTOE)    DATE 2003-01-09

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 (6.0.0)          SMC JES3 CONFIGURATION SUPPORT          PAGE 0001
TIME 15:03:42          ESOTERIC TO XTYPE NUMBER (EETOX)       DATE 2003-01-09

  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 library subsystem
- device location
- device recording technique.

The device location is one of the following:

- in a library, in which case "*libraryname* LSM AA:LL" is displayed, where *libraryname* is the name of the library containing the HSC where the nonlibrary device is defined via the NONLIB esoteric or UNITATTR.
- in a VTSS, in which case "*libraryname* VTSS (*nnnnnnnn*)" is displayed
- in the nonlibrary esoteric, in which case "nonlib" is displayed
- neither in the library or nonlibrary esoteric, in which case "not lib or nonlib" is displayed

The recording technique for non-virtual devices is displayed using the same values used in the RECTECH parameter of the HSC UNITATTR control statement. For devices outside the library, if no UNITATTR statement was defined by the HSC, the recording technique is defaulted according to the device type: 3480 defaults to 18TRACK, 3490 defaults to 36ATRACK, and 3590 defaults to STK1RA35.

```
SMCUPJS (6.0.0)          SMC JES3 CONFIGURATION REPORT          PAGE 0001
TIME 15:03:42          XTYPE TO DEVICE NUMBER (EXTOD)        DATE 2003-01-09
```

```
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  HSCQ LSM 00:00, rectech 18TRACK
0AA0 0AA1 0AA2 0AA3
```

...

```
XTYPE(48) DEV9000  contains 64 DEVICES  HSCQ 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 library subsystem
- device location
- device recording technique.

The device location and recording technique are displayed as in the XTYPE to Device Number report described above.

```
SMCUPJS (6.0.0)          SMC JES3 CONFIGURATION REPORT          PAGE 0001
TIME 15:03:42          DEVICE TO XTYPE NUMBER (EDTOX)        DATE 2003-01-09
```

```
DEVICE/XTYPE - 0120/3C  not lib or nonlib, rectech 36ATRACK
DEVICE/XTYPE - 0121/3C  not lib or nonlib, rectech 36ATRACK
```

...

```
DEVICE/XTYPE - 0AA0/27  HSCQ LSM 00:00, rectech 18TRACK
```

```
DEVICE/XTYPE - 0AA1/27  HSCQ LSM 00:00, rectech 18TRACK
```

```
DEVICE/XTYPE - 0AA2/27  HSCQ LSM 00:00, rectech 18TRACK
```

```
DEVICE/XTYPE - 0AA3/27  HSCQ LSM 00:00, rectech 18TRACK
```

...

Chapter 10. SMC Messages

Overview

This chapter provides information about SMC messages. This information is provided to help system programmers and operators:

- initialize the SMC
- monitor SMC activity
- diagnose and correct SMC problems

See Chapter 11, “StorageTek HTTP Server Messages” for information about messages issued by the StorageTek HTTP Server for OS/390 and z/OS.

Message Format

Messages are displayed on the console in the following format:

SMCnnnn message-text

where:

- SMC identifies the Storage Management Component (SMC).
- *nnnn* is the four-character message number.
- *message-text* is the actual text displayed on the job log or system log.

Message Descriptions

This chapter provides a description for each SMC message. In addition to the message number and message text, this description includes the following:

Message Level

The message level represents a message category. Using the MSGDEF or MSGJOB operator command, the message level is specified in order to control which categories of messages are issued. See “MSGDef Command” on page 135 for more information.

Explanation

The explanation describes the message.

System Action

The system action describes how the SMC reacts when a message-triggering event occurs.

User Response

The user response describes how the user should respond to the message. In many cases, no response is required.

Variable Definitions

Italicized text indicates variable data that is replaced by actual values when messages are issued. Message specific variable data is symbolized as follows:

Table 7. Variable Data Definitions

Variable Data	Definition
<i>AA</i>	ACSid
<i>n</i>	decimal value
<i>X</i>	hexadecimal value
various letters (<i>C, N, K</i> , etc.)	variable information (character data)
numbers (<i>1, 2</i> , etc.) (combined with letters)	related variable information i.e. commands <i>CCCCCCC1</i> and <i>CCCCCCC2</i>
{ }	multiple choices
[]	optional field (may not appear in message)

SMC System Message Listing

The following SMC messages are listed numerically.

- SMC0001** SMC subsystem initializing
- Message Level:** 0
- Explanation:** The MVS start command was entered for the SMC, and the SMC subsystem initialization process has begun.
- System Action:** None.
- User Response:** None.
- SMC0002** CCCCCCCCCCCCCC failed; return code=XXXX, reason code=XXXX
- Message Level:** 0
- Explanation:** MVS facility or macro CCCCCCCCCCCCCC completed with the specified non-zero return code XXXX and reason code XXXX.
- System Action:** Depending upon the type of error, initiation/termination may try to continue. If the MVS facility is listed as *Requested SDUMP*, the error occurred during the TRACE SNAP process, and processing will continue without producing the requested SDUMP.
- User Response:** Look for IBM related messages in the SYSLOG or job log, and refer to IBM documentation for the explanation.
- SMC0003** SMC subsystem SSSS terminating
- Message Level:** 0
- Explanation:** The MVS stop command was entered for the SMC, and the SMC subsystem termination process has begun.
- System Action:** None.
- User Response:** None.
- SMC0004** MVS release is down-level
- Message Level:** 0
- Explanation:** An attempt was made to initialize the SMC on an MVS system that does not support the necessary services required by this version of SMC.
- System Action:** The SMC subsystem terminates.
- User Response:** Upgrade MVS to the required release level.

SMC0005 Invalid command CCCCCCCC [at line nnnn of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: An undefined command, CCCCCCCC, was encountered by the SMC.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0006 SMC subsystem not dynamic

Message Level: 0

Explanation: The SMC was defined as a non-dynamic subsystem using the positional form of the IEFSSNxx parmlib member. The SMC must run as a dynamic subsystem.

System Action: The SMC subsystem terminates.

User Response: Use the keyword format of the subsystem definition in the IEFSSNxx parmlib member to define the SMC subsystem, or remove the SMC subsystem from the IEFSSNxx parmlib member.

SMC0007 SMCCVT incompatible with previous version; defaulting to COLD start

Message Level: 4

Explanation: During initialization, the SMC subsystem detected that the prior SMC subsystem of the same name was an incompatible version. The COLD start parameter was not specified.

System Action: Initialization continues, but in COLD start mode (the SMC subsystem CVT is rebuilt).

User Response: None.

SMC0008 Not running from an authorized library; SMC subsystem terminating

Message Level: 0

Explanation: The SMC subsystem initialization module, SMCBINT, was executed from an unauthorized library.

System Action: The SMC subsystem terminates.

User Response: Ensure that the SMC link library is APF authorized.

SMC0009 Job *JJJJJJJJ* JOB*nnnn* active at termination

Message Level: 4

Explanation: The SMC subsystem received the MVS stop ('P') command, but there is an active job, *JJJJJJJJ*, in tape allocation. The SMC0012 message was issued previously, but the job allocation has not completed after waiting one minute.

System Action: Termination continues. No allocation influencing is performed for JOB*nnnn*.

User Response: None.

SMC0010 Unable to acquire storage for *CCCCCCCC*; return code=*XXXX*

Message Level: 0

Explanation: During initialization, the SMC subsystem could not acquire sufficient storage for the specified dynamic control block or module, *CCCCCCCC*.

System Action: The SMC subsystem terminates.

User Response: Ensure that there is sufficient CSA storage available. Refer to IBM documentation for the explanation of return code *XXXX*.

SMC0011 Load failed for module *MMMMMMMM*

Message Level: 0

Explanation: The SMC subsystem could not load the required module *MMMMMMMM*.

System Action: The SMC subsystem terminates.

User Response: Ensure that the SMC startup procedure has access to all SMC distributed load libraries in its steplib concatenation.

SMC0012 Termination waiting for job *JJJJJJJJ* JOB*nnnn* [in allocation|in message handling]

Message Level: 0

Explanation: The SMC subsystem received the MVS stop command, but there is an active job *JJJJJJJJ* in tape allocation or message handling.

System Action: The SMC waits for 30 seconds or until all active processes are complete.

User Response: Ensure that there are no jobs performing tape allocation or message handling when the SMC is terminated. Respond to any MVS allocation recovery messages.

SMC0013 TRACE SUBSYS SSSSSSSS

Message Level: 0

Explanation: The TRACE command was specified with the LIST keyword. The SMC0013 message lists the SMC subsystems that are traced.

System Action: None.

User Response: None.

SMC0014 Unmatched [quote|or invalid parenthesis] detected; command ignored
[at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command containing an unterminated quoted string, or invalid or unmatched parenthesis.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0015 Invalid keyword *KKKKKKKK* for the *CCCCCCCC* command [at line *nnnn* of
SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command that specified an invalid keyword *KKKKKKKK*.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0016 Invalid value *VVVVVVVV* for keyword *KKKKKKKK* of the *CCCCCCCC* command
[at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command that specified keyword *KKKKKKKK* with an invalid value *VVVVVVVV*.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0017 Keyword *KKKKKKKK* of the *CCCCCCCC* command requires a value [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command that specified keyword *KKKKKKKK* without an accompanying value (required by most keywords).

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0018 Keyword *KKKKKKKK* of the *CCCCCCCC* command is not allowed for *EEEEEEEE* [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command that specified keyword *KKKKKKKK*, which is not valid in the current operating environment *EEEEEEEE*. For example, some keywords or keyword=value pairs may be invalid depending upon whether the user is executing JES2 or JES3.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the specified keyword is valid in your environment.

SMC0019 Duplicate keyword *KKKKKKKK* specified for the *CCCCCCCC* command [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command that specified the same keyword, *KKKKKKKK*, more than once.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0020 Keyword *KKKKKKK1* of the *CCCCCCCC* command is mutually exclusive with keyword *KKKKKKK2* [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command that specified multiple keywords, two of which (*KKKKKKK1* and *KKKKKKK2*), are mutually exclusive.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0021 {COLD|WARM} start failure

Message Level: 0

Explanation: During initialization, the SMC subsystem detected an error.

System Action: The SMC subsystem terminates.

User Response: Look for SMC related messages in the SYSLOG or job log. Associated messages may be (but are not limited to) SMC0002, SMC0004, SMC0006, SMC0008, SMC0010, or SMC0011.

SMC0022 Invalid format or missing keywords for the *CCCCCCCC* command [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command *CCCCCCCC* that contained either too many or too few keywords in the command line.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0023 *CCCCCCCC* command successfully processed [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The *CCCCCCCC* command was successfully validated and processed by the SMC.

System Action: None.

User Response: None.

SMC0024 SSSS subsystem initialization complete

Message Level: 0

Explanation: The SMC initialization process is complete. The SMC is ready to begin normal operations.

System Action: None.

User Response: None.

SMC0025 No {CCCCCCCC|control block} entries to display [(no JES3 tape setup)]

Message Level: 0

Explanation:

One of the following occurred:

- Command CCCCCCCC was specified with the LIST keyword. However, no entries were found in the SMC queue of the specified command.
- A LIST command was specified with a control block keyword. However, no control block entries were found.
- A LIST command was specified for an SMC JES3 control block name. However, the system was not operating with JES3 tape setup.

System Action: None.

User Response: None.

SMC0027 Keyword KKKKKKK1 of the CCCCCCCC command requires keyword KKKKKKK2 [at line nnnn of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command that specified keyword KKKKKKK1, but not the required co-requisite keyword, KKKKKKK2.

System Action: Processing continues. The command is ignored.

User Response: Ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0028 STOP command received

Message Level: 0

Explanation: The MVS stop ('P') command was received by the SMC, and the SMC subsystem termination process is set to begin.

System Action: The SMC subsystem begins termination processing.

User Response: None.

SMC0029 CCCCCCCC command processing error; [matching entry not found|command line truncated; will be ignored|parameter truncated; command ignored] [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: An error was found processing the CCCCCCCC command.

System Action: Processing continues. The command is ignored.

User Response: Use the LIST keyword to list the current SMC queue of entries; then ensure that the syntax in the command data set is correct, or enter the corrected command.

SMC0030 TRACE JOBNAME=JJJJJJJJ [STEPNAME=SSSSSSSS]
[PROCSTEP=PPPPPPPP] [SNAPdump DDDDDDDD] CCCCCCCC

Message Level: 0

Explanation: The TRACE command was specified with the LIST keyword. The SMC0030 message lists the jobs, step, and PROC steps and the associated processes which have been specified for tracing.

System Action: None.

User Response: None.

SMC0031 No SMC start mode specified; defaulting to WARM start

Message Level: 4

Explanation: During SMC initialization, the subsystem detected that neither a WARM or COLD start was specified in the initialization program's execution parameters.

System Action: Initialization continues in WARM start mode.

User Response: None.

- SMC0032** Number of SMC startup parameters specified exceeds maximum of *n*
- Message Level:** 0
- Explanation:** During initialization, the SMC subsystem initialization program encountered an execution parameter string containing too many parameters.
- System Action:** The SMC subsystem terminates.
- User Response:** Correct the subsystem initialization startup procedure to specify the correct execution parameter string.
- SMC0033** SMC startup parameter *PPPPPPP* may not have a value
- Message Level:** 0
- Explanation:** During initialization, the SMC subsystem initialization program encountered a valid execution parameter, but it was specified as a keyword=value pair, where no value is allowed.
- System Action:** The SMC subsystem terminates.
- User Response:** Correct the subsystem initialization startup procedure to specify the correct execution parameter string.
- SMC0034** SMC startup parameter *PPPPPPP* must have a value
- Message Level:** 0
- Explanation:** During initialization, the SMC subsystem initialization program encountered a valid execution parameter, but it was not specified as a keyword=value pair, and a value is required.
- System Action:** The SMC subsystem terminates.
- User Response:** Correct the subsystem initialization startup procedure to specify the correct execution parameter string.
- SMC0035** Error processing SMC startup parameter *PPPPPPP; CCCCCCCCCC*
- Message Level:** 0
- Explanation:** During initialization, the SMC subsystem initialization program encountered an error in the execution parameter string. The string *CCCCCCCCC* indicates the type of error encountered.
- System Action:** The SMC subsystem terminates.
- User Response:** Correct the subsystem initialization startup procedure to specify the correct execution parameter string.

SMC0036 SMC startup parameter *PPPPPPP* successfully processed

Message Level: 4

Explanation: During SMC initialization, the execution parameter *PPPPPPP* was successfully verified and processed.

System Action: None.

User Response: None.

SMC0037 Invalid SMC startup parameters; subsystem terminating

Message Level: 0

Explanation: During initialization, the SMC subsystem initialization program detected an error processing the execution parameter string.

System Action: The SMC subsystem terminates.

User Response: Look for SMC related messages in the SYSLOG or job log. Associated messages may be (but are not limited to) SMC0032 SMC0033, SMC0034, or SMC0035.

SMC0038 Another SMC system *SSSS* is already active

Message Level: 0

Explanation: During initialization, the SMC subsystem initialization program detected a different SMC subsystem, *SSSS*, already active on the system.

System Action: The initializing SMC subsystem terminates.

User Response: Only one SMC can be active on a system at a time.

- If SMC *SSSS* is active, and is preventing the current SMC from initializing, terminate SMC *SSSS*.
- If SMC *SSSS* was terminated abnormally and is not truly active, restart the current SMC subsystem with the RESET execution parameter.

- SMC0039** Identically named subsystem SSSS is already active
- Message Level:** 0
- Explanation:** During initialization, the SMC subsystem initialization program detected an identically named SMC subsystem SSSS already active on the system.
- System Action:** The initializing SMC subsystem terminates.
- User Response:** Only one SMC can be active on a system at a time. If SMC subsystem SSSS was terminated abnormally and is not truly active, then restart SMC SSSS with the RESET execution parameter.
- SMC0040** SMC subsystem SSSS is already active; RESET specified; startup continuing
- Message Level:** 4
- Explanation:** During initialization of SMC subsystem SSSS, the initialization program detected that the prior SMC subsystem terminated abnormally, but the user specified the RESET execution parameter to ignore this condition.
- System Action:** None.
- User Response:** None.
- SMC0041** {Command|Comment} beginning at line *nnnn* of {SMCCMDS|SMCPARMS} is unterminated
- Message Level:** 4
- Explanation:** A command or comment beginning at line *nnnn* of an input command file ended with a continuation character (+), but no continuation was found.
- System Action:** Processing continues. The command containing the unterminated string is ignored.
- User Response:** Ensure that the syntax in the command data set is correct.
- SMC0042** Job *JJJJJJJJ* step *SSSSSSSS* not allocatable before SMC modification
- Message Level:** 4
- Explanation:** During execution of job *JJJJJJJJ* step *SSSSSSSS*, the SMC allocation component detected that the job step was not allocatable before any allocation influencing was performed.
- System Action:** None. The job is failed by MVS or express-canceled by JES3.
- User Response:** Correct the JCL.

SMC0043 Job *JJJJJJJJ* step *SSSSSSSS* not allocatable at MINLVL=*nn*; failing DD
DDDDDDDD

Message Level: 4

Explanation: During execution of a job with tape allocation, the SMC allocation component detected that the job is not allocatable at the specified minimum exclusion level (MINLVL).

System Action: The job is failed by MVS or express-canceled by JES3.

User Response: Correct the JCL, or change the MINLVL for the specified job.

SMC0044 SMC subsystem termination in progress; no allocation influence for job *JJJJJJJJ* step *SSSSSSSS*

Message Level: 8

Explanation: During execution of job *JJJJJJJJ* step *SSSSSSSS*, the SMC allocation component detected that the SMC subsystem was being terminated.

System Action: No allocation influencing is performed for the job.

User Response: None.

SMC0045 Conflicting exclusion criteria for job *JJJJJJJJ* step *SSSSSSSS* DD
DDDDDDDD

Message Level: 8

Explanation: Messages SMC0045 and SMC0046 are always produced together. See the explanation for message SMC0046 for more information.

System Action: None.

User Response: None.

SMC0046 CCCCCCCCCCCCCCCC would have excluded all devices; not honored

Message Level: 8

Explanation: When the SMC allocation component attempted to apply the exclusion criteria CCCCCCCCCCCCCCCC for job JJJJJJJJ step SSSSSSSS DD DDDDDDDD, no devices remained in the current exclusion level EDL. The SMC0045 and SMC0046 messages are always produced together.

For information about exclusion criteria, refer to the keyword column of the “JES2 Drive Exclusion Levels” and “JES3 Drive Exclusion Levels” tables in the *SMC Configuration and Administration Guide*.

System Action: None. The criteria that would have eliminated all devices is ignored.

User Response: None.

SMC0047 Esoteric EEEEEEEE contains no known devices; ignored for job JJJJJJJJ step SSSSSSSS DD DDDDDDDD

Message Level: 8

Explanation: During execution of job JJJJJJJJ step SSSSSSSS, the SMC allocation component encountered a user policy esoteric, EEEEEEEE (from TAPEREQ or user exit), that did not contain any tape devices in the current EDL.

System Action: The specified esoteric is ignored.

User Response: None.

SMC0048 ACS ID aa contains no known devices; ignored for job JJJJJJJJ step SSSSSSSS DD DDDDDDDD

Message Level: 8

Explanation: During execution of job JJJJJJJJ step SSSSSSSS, the SMC allocation component encountered an user exit policy requesting exclusion to ACS ID aa. However, the EDL at the current exclusion level does not include tape devices in the requested ACS.

System Action: The ACS ID policy request is ignored.

User Response: None.

SMC0049 No eligible libraries for job *JJJJJJJJ*

Message Level: 8

Explanation: During execution of job *JJJJJJJJ*, the SMC allocation component did not find any libraries at the correct release level.

System Action: If ALLOCDEF FAILnoinfo(ON) is specified, the SMC marks all devices ineligible. Otherwise, no allocation influencing is performed for the job.

User Response:

- If libraries are implied using the MVS SSCVT chain or via the LIBrary command LOCSubsys parameter, ensure that the HSC or MVS/CSC(s) on the host are active and at the correct level.
- If libraries on another host are defined using the LIBrary command, ensure that the HSC on the remote host defined by the SERVer command is active and at the correct level. Also ensure that the HTTP server is active on the remote host.

SMC0051 SMC subsystem interface error for job *JJJJJJJJ*; reason=*SSSSSSSS*
function=*CCCCCCCC*

Message Level: 4

Explanation: SMC processing in an MVS initiator address space attempted to communicate with the SMC subsystem in order to process an allocation or mount request but encountered an error. The reason for the failure is *SSSSSSSS* while the function attempted was *CCCCCCCC*.

System Action: The SMC will not be able to continue processing the allocation or mount request.

User Response: Investigate the cause of the error by looking for related SMC subsystem, or MVS error messages.

SMC0052 User exit *nn* {inactive|abended and disabled}

Message Level: 8

Explanation: During execution of a job with tape allocation, the SMC allocation component invoked the user exit *nn* from an HSC or MVS/CSC subsystem. However, the user exit is currently inactive, or has abended and is now disabled for both the SMC and the owning HSC or MVS/CSC subsystems.

System Action: Allocation influencing continues for the job.

User Response: Correct the user exit.

SMC0053 **** SMC U1099 ABEND at CCCCCCCCn ****

Message Level: 0

Explanation: An SMC task has abended in module CCCCCCCC at abend sequence number *n*.

System Action: If the abend occurs in the address space of a tape allocation job, the SMC subsystem does not influence the job's allocation. If the abend occurs in the SMC started task address space, a restart of the SMC subsystem may be required.

User Response: Look for SMC or IBM related messages in the SYSLOG or job log. Save the associated logs, dump data set, and JCL, and contact Sun Microsystems StorageTek Technical Support.

SMC0054 MSGJOB LVL=*nn* for {DEFAULT SETTING|JOBNAME=JJJJJJJJ}
[STEPNAME=SSSSSSSS][PROCSTEP=PPPPPPPP]

Message Level: 0

Explanation: A MSGJOB command was specified with the LIST keyword. Each unique job, job step, and PROC step entry found in the MSGJOB queue is displayed in a separate SMC0054 message, followed by the SMC defaults in a final SMC0054 message. *nn* indicates the message level, and any messages at that indicated level or lower are produced on that job's job log in JES2 or on the system log in JES3.

System Action: None.

User Response: None.

SMC0055 ALLOCJOB {BYPASSEd|MINLVL=*nn*}[SEPLVL=MIN|MAX|*nn*][SMS=OFF|ON] for
{JOBNAME=JJJJJJJJ}[STEPNAME=SSSSSSSS][PROCSTEP=PPPPPPPP][SNAPdump
DDDDDD] PPPPPPPP

Message Level: 0

Explanation: An ALLOCJOB command was specified with the LIST keyword. Each unique job, job step, and PROC step entry found in the ALLOCJOB queue is displayed in a separate SMC0055 message. The BYPASSED message indicates that the specified job, job step, or PROC step will not have any of its tape allocations influenced by the SMC. MINLVL=*nn* indicates the desired minimum exclusion level at which affinity and GDG chains are separated. The SMS value indicates whether the SMC DFSMS interface is enabled for this entry.

System Action: None.

User Response: None.

SMC0056 Listing *nnnn* bytes at address *XXXXXXXX*

Message Level: 0

Explanation: The SMC encountered a valid LIST command.

System Action: Following the SMC0056 message, the SMC displays multiple lines listing the storage in translated hexadecimal format, each line listing the next 16 bytes of storage.

User Response: None.

SMC0057 No {SMCPARMS|SMCCMDS} DDNAME statement found

Message Level: 8

Explanation: During SMC initialization, the specified SMCPARMS DD or SMCCMDS DD was not present in the SMC startup procedure.

System Action: Initialization continues.

User Response: None.

SMC0058 Error opening {DDNAME|DSNAME} {SMCPARMS|SMCCMDS|*datasetname*}

Message Level: 0

Explanation: The SMC encountered a READ command, but the specified DDNAME or DSNAME could not be opened.

System Action: The READ command is ignored.

User Response: Look for IBM related messages in the SYSLOG or job log, and refer to IBM documentation for more information.

SMC0059 Identically named subsystem *SSSS* is initializing

Message Level: 0

Explanation: During initialization, the SMC subsystem initialization program detected an identically named SMC subsystem, *SSSS*, already being initialized (but not yet fully active).

System Action: The currently initializing SMC subsystem terminates.

User Response: Only one SMC can be active on a system at a time. If the prior SMC subsystem, *SSSS*, was terminated abnormally, and is not truly active, then restart SMC *SSSS* with the RESET execution parameter.

SMC0060 I/O error reading [DDNAME {SMCPARMS|SMCCMDS}|DSNAME *datasetname*]

Message Level: 0

Explanation: The SMC received an I/O error attempting to read the SMCPARMS or SMCCMDS data set specified in the SMC started procedure or a data set specified on a READ command.

System Action: The indicated data set is not processed.

User Response: Determine the cause of the error. If the input data set is a PDS, ensure that a member name was specified.

SMC0061 Command beginning at line *nnnn* of {SMCCMDS|SMCPARMS} is too long; input ignored

Message Level: 4

Explanation: The SMC encountered a multi-line command beginning at line *nnnn* of the specified file. This command exceeds 1024 characters in length.

System Action: Processing continues. The entire multi-line is ignored.

User Response: Ensure that the command data set has the correct syntax.

SMC0062 Command *CCCCCCC* [with parameter *PPPPPPP*] is not allowed [{from console|at line *nnnn* of SMCCMDS|SMCPARMS}]

Message Level: 0

Explanation: The SMC encountered a command or a command parameter that is not supported for the indicated command origin. For example, the READ command is not supported if encountered during processing of another READ command.

System Action: The command is ignored.

User Response: Issue the command from a valid command origin.

SMC0063 MSGDEF LVL=*nn* CASE={UPPER|MIXED} ECHOmnterr={OFF|ON}
SMCWtorpfx={ON|OFF}

Message Level: 0

Explanation: A MSGDEF or MSGJOB command was specified with the LIST keyword. The SMC0063 message lists the default message settings for the SMC subsystem. *nn* indicates the message level, and any messages at that indicated level or lower will be produced.

System Action: None.

User Response: None.

SMC0066 ALLOCDEF MINLVL=*nn* SEPLVL={MIN|MAX|*nn*} SMS={OFF|ON}
FAILNOINFO{ON|OFF} MIACOMPAT={OFF|ON} CA1RTS={OFF|ON}

Message Level: 0

Explanation: An ALLOCDEF or ALLOCJOB command was specified with the LIST keyword. The SMC0066 message lists the default allocation settings.

- MINLVL indicates the exclusion level at which a job step must have drives remaining in order not to fail allocation.
- SEPLVL indicates the exclusion level at which affinity chains will be separated.
- SMS indicates whether or not the SMC DFSMS interface is enabled.
- FAILNOINFO indicates whether the SMC intentionally fails a job during allocation if a volume lookup failure occurs, or if no libraries are available.
- MIACOMPAT indicates whether or not the EDL is marked for excluded devices at SSI24 time.
- CA1RTS indicates whether DEFER processing will be performed at SSI24 time

System Action: None.

User Response: None.

SMC0067 SMSDEF IGDACSXT=[ON|OFF] ESOTSUB=[ON|OFF] DCRECMED=[ON|OFF]
VTVMGMT=[ON|OFF]

Message Level: 0

Explanation: The SMSDEF command was specified with the LIST keyword. The SMC0067 message lists the default SMS settings for the SMC subsystem.

System Action: None.

User Response: None.

SMC0068 Cannot substitute for esoteric EEEEEEE at MINLVL=*n* job JJJJJJJJ
step SSSSSSSS DD DDDDDDDD

Message Level: 8

Explanation: During execution of job JJJJJJJJ step SSSSSSSS on JES3, the SMC allocation component determined that the job step was not allocatable at the specified minimum level (MINLVL).

For example, if you are executing at the default minimum exclusion level 2, this message indicates that the SMC is unable to select an esoteric containing only drives compatible with the media type of a specific volume.

System Action: None. No esoteric substitution is performed. The job may later fail because a volume is requested on an incompatible drive.

User Response: Refer to “JES3 SETNAME Initialization Statements” in Chapter 5 for an explanation of esoteric definition requirements.

SMC0069 SMCERSLV release level LLLLLL1 does not match SMC release level
LLLLLL2

Message Level: 0

Explanation: Module SMCERSLV is at release level LLLLLL1 which does not match SMC release level LLLLLL2.

System Action: The SMC does not initialize.

User Response: Reassemble SMCERSLV at the correct release level and restart the SMC.

SMC0070 SMC maintenance has been applied; reassemble SMCERSLV

Message Level: 0

Explanation: SMC maintenance was applied and affected the module SMCERSLV, but SMCERSLV was not reassembled.

System Action: The SMC does not initialize.

User Response: Reassemble SMCERSLV using the correct level of SMC macros and restart the SMC.

SMC0071 SMCERSLV JES3 release level *LLLLLL1* does not match JES3 release level *LLLLLL2*

Message Level: 0

Explanation: Module SMCERSLV was assembled using a different JES3 release level *LLLLLL1* than the active JES3 release level *LLLLLL2*.

System Action: The SMC does not initialize.

User Response: Reassemble SMCERSLV using the correct level of JES3 macros and restart the SMC.

SMC0073 JES3 C/I waiting for SMC to initialize; Start SMC or reply "GO" to continue

Message Level: 0

Explanation: JES3 has begun scanning the JCL of jobs requiring tape mounts and the SMC is not initialized and cannot influence allocation.

System Action: One JES3 C/I process waits until the SMC has been started or the operator has replied "GO".

User Response: Start the SMC or reply "GO" to proceed without SMC allocation influence.

SMC0074 Unsupported virtual label type for job *JJJJJJJJ* step *SSSSSSSS* DD *DDDDDDDD*

Message Level: 8

Explanation: All virtual devices were previously excluded for the DD because of an unsupported label type (NL). A subsequent exclusion criteria requesting virtual drives could not be honored. Message SMC0046 describes the criteria not honored.

System Action: The DD is allocated to a non-virtual drive. Non-library drives are preferred over library drives.

User Response: Change the JCL to request a supported label type, or change the policy to direct the allocation to a different device type.

SMC0075 SEPLVL cannot be less than MINLVL on the *CCCCCCC* command [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The SMC encountered a command that specified either a SEPLVL that is lower than the applicable MINLVL or a MINLVL higher than the applicable SEPLVL.

System Action: Processing continues. The command is ignored.

User Response: Ensure that command MINLVL and SEPLVL values are correct in the command data set, or enter the corrected command.

SMC0076 Xtype *CCCCCCC* (*XX*) has inconsistent {device type|location type|VTSS|ACS} between device *DDDD1* and device *DDDD2*

Message Level: 4

Explanation: In a JES3 environment, the SMC encountered an XTYPE that contains non-homogeneous devices, where *CCCCCCC* represents the JES3 SETNAME name. Use of this XTYPE may result in allocation to an incorrect device based on media, recording technique, or location.

System Action: Allocation is performed based on the device characteristics of the first device encountered in the XTYPE.

User Response: Refer to Chapter 3, “Starting the SMC” for JES3 initialization parameter requirements.

SMC0077 Subtask *CCCCCCC* terminating at retry count *n*; please start and restart SMC subsystem

Message Level: 0

Explanation: The specified SMC subsystem task abnormally terminated *N* times, and could not be restarted. The subsystem is now operating without a required service.

System Action: Processing continues. However, specific allocation or command facilities may be affected.

User Response: Stop and restart the SMC subsystem. If the named subtask is SMCOCMD, use the MVS CANCEL command to terminate the subsystem.

SMC0078 No command prefix defined

Message Level: 0

Explanation: No CMDDEF command specified a command prefix in the SMCPARMS data set.

System Action: Processing continues.

User Response: If a command prefix is desired, add the CMDDEF command to the SMCPARMS data set. The command is processed the next time SMC is initialized.

SMC0079 The command prefix is *PPPPPPPP*

Message Level: 0

Explanation: The command prefix for the subsystem is set to *PPPPPPPP*.

System Action: The SMC now accepts commands prefixed with *PPPPPPPP*.

User Response: None.

SMC0080 Command prefix value *VVVVVVVV* contains invalid character *C* at line *nnnn* of SMCPARMS

Message Level: 0

Explanation: The command prefix value *VVVVVVVV* of the CMDDEF command contained an invalid character *C*.

System Action: Processing continues. The command is ignored.

User Response: Review the list of valid characters for the command prefix listed with the CMDDEF command description. Update the CMDDEF command in the SMCPARMS data set with the new prefix value. The command is processed the next time the SMC is initialized.

SMC0081 Command prefix not added; [prefix is not unique|CPF system error]

Message Level: 0

Explanation: The command prefix was disallowed by the CPF facility because the prefix was not unique or a system error occurred.

System Action: Processing continues. The command is ignored.

User Response:

- If the prefix is not unique, issue the MVS DISPLAY OPDATA command and compare the prefix with those of the other subsystems. 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.
- If a CPF error occurred, look for IBM related messages in the SYSLOG and refer to IBM documentation for the explanation.

SMC0082 Command prefix already set

Message Level: 0

Explanation: The command prefix can be specified only once while the SMC is initializing and cannot be changed during execution.

System Action: Processing continues. The command is ignored.

User Response: To change the command prefix, update the CMDDEF statement in the SMCPARMS data set and recycle the SMC.

SMC0083 Unable to locate {JES3 SETNAME table|JES3 SETUNIT table}

Message Level: 0

Explanation: During SMC subsystem initialization on a JES3 system, the indicated JES3 control structures could not be located. The SMC could not initialize.

System Action: The SMC subsystem terminates.

User Response: Reassemble SMCERSLV with the current level of JES3 macros. Refer to the *NCS Installation Guide* for more information.

SMC0084 *MMM DD YYYY HH:MM:SS SSSS active on hostid HHHH*

Message Level: 4

Explanation: The date (*MMM DD YYYY*), time (*HH:MM:SS*), subsystem name (*SSSS*) and MVS hostid (*HHHH*) are displayed once a day at midnight and during SMC initialization.

System Action: None.

User Response: None.

SMC0085 *SMCBPREI: IEFSSI failed RC=xx RS=xxxx*

Message Level: 0

Explanation: The SMC subsystem pre-initialization routine (SMCBPREI) encountered an error.

System Action:

- If the Return Code (RC) is 4 or less, the SMC subsystem was defined.
- If the Return Code (RC) is 8 or higher, the SMC subsystem was not defined.

User Response: Investigate the Return Code (RC) and Reason (RS) and take the necessary action. The return code and reason are documented in the IBM manual *MVS Programming: Authorized Assembler Service Reference, Volume 2*.

SMC0086 *A(Sub) A(Main) Type Program Stat Jobname JobId Last*

SMC0087 *AAAAAAAA BBBB BBBB C-D P P P P P P P P SSSS JJJJJJJJ I I I I I I I I T T T T T T*

Message Level: 0

Explanation: An SMC LIST TASK command was entered. The SMC0086 heading line is followed by multiple SMC0087 detail lines displaying the status of each task in the SMC subsystem address space. An SMC0094 total line will also be displayed.

System Action: None.

User Response: None.

SMC0088 Unable to [acquire/release] resource *name*; attempt by *jobname*
ID=*key1* owned by ID=*key2*

Message Level: 0

Explanation: A shared SMC resource could not be acquired or freed successfully. Job *jobname* is the task currently attempting to acquire or free the resource, but cannot because another task holds the resource.

System Action: Jobname *jobname* may not be processed correctly.

User Response: Contact Storage Technology technical support.

SMC0089 Unable to start subtask CCCCCCCC

Message Level: 0

Explanation: During initialization, the indicated SMC subsystem task could not be successfully attached.

System Action: The SMC subsystem terminates.

User Response: Look for related MVS or SMC messages in the SYSLOG or SMC job log.

SMC0090 Unable to restart subtask CCCCCCCC

Message Level: 0

Explanation: During subsystem processing the indicated SMC subsystem task abended and could not be restarted.

System Action: Processing continues. However, subsystem processing or command facilities may be affected.

User Response: Stop and restart the SMC subsystem. If the indicated subtask is SMCOCMD, use the MVS CANCEL command to terminate the subsystem.

SMC0091 Could not allocate job JJJJJJJJ step SSSSSSSS after applying all exclusion levels; backing up until allocatable

Message Level: 8

Explanation: The indicated job step would not allocate when all exclusion criteria were applied to all DDs in the step. This means that the remaining set of drives is insufficient for each DD to be assigned a unique drive.

System Action: The SMC backs out exclusion criteria on selected DDs in the step until the set of remaining drives is sufficient to allocate to all DDs in the step.

User Response: None.

SMC0093 TCP/IP TCPNAME=*tcpname* ADSNAME=*adsname*

Message Level: 0

Explanation: A TCP/IP LIST command was executed. The current values of the TCPNAME and ADSNAME are displayed. If the SMC is using the default settings, "default" is displayed.

System Action: None.

User Response: None.

SMC0094 SMCQWRK executing tasks=*NN* waiting tasks=*NN* requests=*NNNNNN*

Message Level: 0

Explanation: An SMC LIST TASK command was entered. The SMC0094 message displays the total task and request counts.

System Action: None.

User Response: None.

SMC0095 UX01 function code UX01RPLY not supported

Message Level: 4

Explanation: The function code return of UX01RPLY (reply to a WTOR message) is not supported by SMC.

System Action: None. The user exit is ignored.

User Response: None.

SMC0096 Invalid UX01 function code X'xx'

Message Level: 4

Explanation: A user exit 01 returned an invalid function code of X'xx'.

System Action: All data returned by this invocation of UX01 is ignored.

User Response: Correct the UX01 to return only function codes documented in the *MVS/CSC* or *HSC System Programmer's Guide*.

- SMC0097** UX01 function code X'xx' not valid for message *MMMMMMMM*
- Message Level:** 4
- Explanation:** The SMC detected that the action code returned for a message (mount, dismount, or swap) did not match the action for an SMC default message.
- System Action:** All data returned by this invocation of UX01 is ignored.
- User Response:** Correct the UX01 to return a function code compatible with the message.
- SMC0098** UX01 drive *XXXX* does not match message drive *XXXX*, using UX01 drive
- Message Level:** 4
- Explanation:** The device address returned from UX01 does not match the address specified in the message.
- System Action:** The device address returned from UX01 is used in the message processing.
- User Response:** Ensure that UX01 is functioning as desired.
- SMC0099** Drive *XXXX* not a library drive
- Message Level:** 4
- Explanation:** A user exit 01 returned a drive *XXXX*, which is not a library or virtual drive.
- System Action:** All data returned by this invocation of UX01 is ignored.
- User Response:** Ensure that UX01 is functioning as desired.
- SMC0100** Invalid UX01 drive *XXXX*
- Message Level:** 4
- Explanation:** A user exit 01 returned an invalid drive address of *XXXX* which is not defined as a valid tape device on this system.
- System Action:** All data returned by this invocation of UX01 is ignored.
- User Response:** Correct the UX01.

SMC0101 Invalid UX01 or TAPEREQ subpool SSSSSSSSSSSSS

Message Level: 4

Explanation: A user exit 01 or TAPEREQ specified an invalid subpool name.

System Action: The default subpool 0 is used.

User Response: Correct the UX01 or TAPEREQ to specify a valid subpool.

SMC0102 Invalid UX01 volser *volser*

Message Level: 4

Explanation: A user exit 01 returned a volume serial containing invalid characters.

System Action: The volume serial returned by this invocation of UX01 is ignored.

User Response: Correct the UX01 to return a valid volume serial.

SMC0103 Unrecoverable mount error on device *XXXX* volser *volser* for JOB
JJJJJJJJ

Message Level: 0

Explanation: The SMC detected a volume mount error in response to an IAT5310 message for a mount requested in message IAT5210.

System Action: The SMC breaks the mount loop and issues a dismount to the indicated device *XXXX*. The job remains in the MDS VERIFY queue.

User Response: Refer to the IAT5310 message for the cause of the mount error and take corrective action.

SMC0104 Default recording technique set for device *XXXX*

Message Level: 0

Explanation: During initialization of the SMC drive table, a device with an unrecognized recording technique was detected.

System Action: A recording technique is defaulted based on the UCB device type. Processing continues.

User Response: Verify that any maintenance for new device types has been applied to all NCS products, including SMC.

SMC0106 Mount of HSC volser *volser* on non-HSC device *XXXX*

Message Level: 0

Explanation: An HSC subsystem on the host specified a VOLWATCH parameter on the MNTD command. The SMC has detected that volume resident in the HSC library is being requested to be mounted on a drive outside the HSC library.

System Action: None.

User Response: Cancel the job requesting the mount, or eject the volume to satisfy the mount request.

SMC0107 Swap volser from *XXXX1* to *XXXX2*

Message Level: 4

Explanation: The SMC has intercepted DDR swap processing. *XXXX2* is either the original device selected to swap to or a device selected by SMC that more closely matches the device characteristics of *XXXX1*.

System Action: DDR swap processing continues.

User Response: None.

SMC0108 No compatible drive found for SWAP processing

Message Level: 0

Explanation: The SMC has intercepted DDR swap processing. The original device chosen to swap to is not compatible with the swap from device and the SMC could not locate a compatible alternate device.

System Action: DDR swap processing continues.

User Response: If no compatible device is available, reply NO to message IGF500D or IGF509D to stop the swap process. If a compatible device is offline, vary it online and reply with its device number to message IGF500D or IGF509D.

SMC0109 The SMC subsystem is running in key *n*; results are unpredictable; reply 'Y' to continue or 'N' to terminate

Message Level: 0

Explanation: During initialization SMC detected that it is running in key *n*, not 1 through 7.

System Action: The SMC waits until a reply is received.

User Response: A reply of N stops SMC. A reply of Y causes SMC to continue initialization, though results are unpredictable. Possible problems are S0C1 and S0C4 ABENDs. To prevent this message, update the Program Properties Table (PPT) with "PPT PGMNAME(SMCBINT),SYST,KEY(*n*)", where *n* is between 1 and 7, inclusive.

SMC0110 Allow swap of *volser* from *XXXX1* to *XXXX2*; Reply 'Y', 'N' or DEVICE

Message Level: 0

Explanation: The SMC has intercepted DDR swap processing. The SMC is awaiting operator approval to allow the swap.

System Action: The SMC continues processing; however, the swap cannot complete until an operator reply is entered.

User Response: To allow the swap to proceed using the selected device *XXXX2*, reply Y. To select a different swap to device, reply with its device address. The SMC does not validate a new device address. To cancel the swap, reply N. If an I/O error on device *XXXX1* caused the swap, a reply of N causes the job to fail.

SMC0111 USER defined message ID: *MMMMMMMM*

Message Level: 0

Explanation: The SMC has listed the messages that were added using the USERMsg command.

System Action: Both SMC default and USERMsg added messages are sent to user exit 01 for each library subsystem. SMC processes USERMsg added messages according to the response from user exit 01.

User Response: None.

SMC0112 Cannot add duplicate message ID *MMMMMMMM* at line *nnnn* of SMCPARMS

Message Level: 0

Explanation: The USERMsg command was entered to add a new message ID that SMC will intercept. The supplied message ID *MMMMMMMM* is a duplicate of a message already defined to SMC.

System Action: The message is not added.

User Response: None.

SMC0113 *servername* LIBRARY=*library*
STATUS=[disabled|active|inactive|never active]

Message Level: 0

Explanation: The SERVER List command was executed. This message, as well as SMC0114 and SMC0115, is produced for each server defined.

System Action: None.

User Response: None.

SMC0114 *servername* Location [HOST=*host*|IPADDRESS=*address*[PORT=*port*]]

Message Level: 0

Explanation: The SERVER List command was executed. This message, as well as SMC0113 and SMC0115, is produced for each server defined.

System Action: None.

User Response: None.

SMC0115 *servername* WAIT=*wait* RETRY=*retry* FAIL=*fail* MSGS=*msgs*
RETRIES=*retries* ERRORS=*errors* INIT ERRORS=*errors*

Message Level: 0

Explanation: The SERVER LIST command was executed. This message, as well as SMC0113 and SMC0114, is produced for each server defined.

- MSGS indicates the number of requests directed to the server.
- RETRIES indicates the total number of retries attempted.
- ERRORS indicates the number of communication failures recorded **after** successful communication was established. ERRORS are counted against the SERVER FAIL limit, if the SERVER FAIL limit is not specified as 0. When the number of ERRORS exceeds the SERVER FAIL limit, the SERVER is automatically disabled by the SMC.
- INIT ERRORS indicates the number of communication failures recorded **before** successful communication was established.

System Action: None.

User Response: None.

SMC0116 Cannot find LIBRARY *libraryname* for SERVER=*name* [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The user specified a SERVER command with a LIBRARY name that was not previously defined.

System Action: The SERVER is not added or updated.

User Response: Specify a LIBRARY command to define the library, then specify the SERVER command.

SMC0117 Cannot change LIBRARY name for existing SERVER *servername* [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The user specified a SERVER command with the NAME of an existing SERVER and a LIBRARY name, but the LIBRARY name of the existing server did not match the library name in the new command.

System Action: The command is rejected.

User Response: Omit the library name, change the library name to match the existing server, or change the SERVER name to add a new server to the specified library.

SMC0118 No LOCSUBSYS for [LOCENABLE|LOCDISABLE] parameter on LIBRARY command [at line nnnn of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The user entered an LOCENABLE or LOCDISABLE parameter on a LIBRARY command, but the library was not defined with a LOCSUBSYS subsystem name.

System Action: This message is a warning. The LIBRARY will be added or modified, but the LOCENABLE or LOCDISABLE parameter is ignored.

User Response: None.

SMC0119 [SERVER|LOCSUBSYS] *name* now disabled

Message Level: 0

Explanation: The SMC detected TCP/IP errors in excess of the FAIL count or detected a fatal error for a local subsystem. See the preceding SMC0128/SMC0129 messages for the reason for the disable.

System Action: None. If there are no additional server paths defined for the associated library, the library hardware is no longer accessible from this host.

User Response: Correct the problem with the TCP/IP network, SMC server, host operating system, or the HSC subsystem, and re-ENABLE the SERVER or LOCSUBSYS.

SMC0120 NAME must be specified to add or modify a SERVER [at line nnnn of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The user entered a SERVER command to add a new SERVER or modify an existing SERVER path, but the SERVER path NAME was not specified.

System Action: None.

User Response: Enter the SERVER command specifying the SERVER path NAME.

SMC0121 UEXIT *nn* is now [ENABLED/DISABLED]

Message Level: 0

Explanation: The user entered a UEXIT *nn* ENABLE or DISABLE command, or the SMC detected an abend in the specified user exit. The specified user exit is now ENABLED or DISABLED.

System Action: None.

User Response: If the SMC automatically disabled the user exit due to an abend, the exit cannot be re-enabled unless it is also reloaded.

SMC0122 Uexit *nn* STATUS=[ACTIVE|INACTIVE|DISABLED|ABENDED]
FORMAT=[HSC|CSC] NAME=nnnnnnnn SEQ=*ss*

SMC0123 Uexit *nn* LOAD *load date time* CHANGE *change date time*

Message Level: 0

Explanation: The user entered a UEXIT *nn* LIST command. The indicated user exit's status, format (HSC or CSC), module name and load sequence are listed.

The status may be one of the following:

- ACTIVE - the user exit has been loaded and enabled
- INACTIVE - the user exit inactivated itself via a return code
- DISABLED - the user exit has been disabled by an operator command
- ABENDED - the user exit abnormally abended.

The load sequence is the number of times this user exit has been loaded.

The load date time is the date and time that this version of the exit was loaded.

The change date time is the date and time that the user exit's status was altered.

System Action: None.

User Response: None.

SMC0124 ABENDED or INACTIVE UEXIT *nn* must be reloaded to ENABLE [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The user entered a UEXIT *nn* ENABLE command for a user exit that had previously abended, or inactivated itself via a return code.

System Action: None.

User Response: Use the UEXit command with the LOAD keyword to reload the exit.

SMC0125 UEXIT *nn* already [ENABLED/DISABLED][at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The user entered a UEXIT *nn* ENABLE or DISABLE command but the exit was already in the specified state.

System Action: None.

User Response: Verify the exit number you are trying to alter.

SMC0126 UEXIT *nn* not valid for [JES2/JES3] [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: The user entered a UEXIT command for a user exit number not valid for this configuration.

System Action: None.

User Response: Verify the user exit number for your configuration.

SMC0127 Cannot resolve HOST name *name* [at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: A SERVER command was specified using the HOST keyword specification. However, the SMC could not resolve the IP address using the specified HOST *name*.

System Action: The SERVER command is ignored.

User Response: Enter the correct HOST name or use the IPADDRESS keyword instead.

SMC0128 {Communication|Initialization} error num [of *max*] on SERVER=*name* for job=*jobname* | Library interface error on local subsystem=*SSSS* for job=*name*



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

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

SMC0129 Communication *error reason*

Message Level: 4

Explanation: A communication error was encountered on the SERVER path *name* or *local subsystem name* for job *jobname*. The error *num* indicates the cumulative error count on the server path, while *max* is the maximum allowed before the named SERVER is disabled by the SMC. The SMC0129 message lists the type of error encountered. Examples include:

- Specific TCP/IP function error
- Data error (incomplete or invalid data sent or received)
- Subsystem inactive or not found, or at incorrect release level
- Subsystem function error
- HTTP server not authorized

System Action: Allocation or mount may not be processed by the SMC.

User Response: Use the specified error reason to determine the cause of the problem. If communicating with a remote HSC, verify the HTTP server is active.



Note: Certain error conditions may result in the display of multiple SMC0129 messages. When these errors are related to unexpected data received from the HTTP server, the entire response may be displayed as in the following example:

```
SMC0129 REASON=DATA ERROR; SMCCGIRE HEADER NOT FOUND
SMC0129 HTTP RESPONSE TEXT=
SMC0129 HTTP/1.0 401 UNAUTHORIZED
```

These SMC0129 messages indicate that the SMC message was rejected by the StorageTek HTTP server.

SMC0130 LIBRARY *libraryname* has same LOCSUBSYS=*locsubsysname* as LIBRARY *libraryname2*[at line *nnnn* of SMCCMDS|SMCPARMS]

Message Level: 0

Explanation: A LIBRARY command was entered with a different library name *libraryname* but the same LOCSUBSYS name as an existing library *libraryname2*.

System Action: The LIBRARY command is not executed.

User Response: Correct the error and resubmit the command.

SMC0131 Cannot find active [LIBRARY|SUBSYSTEM]=CCCCCCCC returned by user exit

Message Level: 4

Explanation: User exit 02/04/08/13 returned with data indicating that SUBSYSTEM CCCC or LIBRARY CCCCCCCC is to own the allocation event. However, the specified LIBRARY or SUBSYSTEM is not active or cannot be found.

System Action: The user exit data relating to event ownership is ignored. However, other data from the exit, such as subpool, and preferencing data is honored.

User Response: .Ensure that the specified SUBSYSTEM or LIBRARY is active.

SMC0132 {Mount error reason|TAPEREQ error reason} for volser=VVVVVV device=DDDD from LIBRARY=LLLLLLLL SERVER=SSSSSSS

Message Level: 4

Explanation: When SMC mount or mount TAPEREQ retrieval processing determines that an MVS mount is still pending and should be retried, it waits before attempting the retry. After the mount or mount policy retrieval retry limit is reached, the SMC0132 message is displayed.

System Action: The SMC will not continue to process the mount request after this message is issued.

User Response: Investigate the cause of the HSC mount problem. After resolving the problem, issue the SMC RESYNChronize REStart command or the HSC MOUNT command if necessary.

SMC0133 *libraryname* STATUS={disabled|active|inactive|never active}
[LOCSUBSYS={locsubsysname|not specified}] [LOCDISABLE]

Message Level: 0

Explanation: A LIBRARY command was entered with the LIST or SERVERLIST keyword. A LIBRARY has a DISABLED status only if disabled via a command. An enabled library is either active or inactive. When SMC successfully communicates with a library, it is marked as active.

System Action: None.

User Response: None.

SMC0134 *libraryname* SERVER=*servername* STATUS=[disabled|active|inactive]

Message Level: 0

Explanation: A LIBRARY command was entered with the SERVERLIST keyword. A SERVER has a DISABLED status only if disabled via a command, or if disabled because its error limit was reached. An enabled server is either active or inactive. Each LIBRARY has at most one active server.

System Action: None.

User Response: None.

SMC0135 {Mount/dismount message|Client response WTOR message|WTOR message issued by server} from LIBRARY=*library* SERVER=*servername*

SMC0136 *HSC mount or dismount message*

Message Level: 4

Explanation: A mount or dismount was performed on a remote HSC server but did not complete successfully. The SMC0135 message indicates the LIBRARY name and SERVER name where the mount or dismount was requested. The SMC0136 merely echoes the HSC server mount or dismount message on the SMC client.

System Action: None.

User Response: Correct the problem indicated in the HSC mount or dismount message.



Note: Messages SMC0135 and SMC0136 are issued for all mount errors if the message level is 12 or greater, and for all mount and dismount activity if the message level is 16 or greater.

SMC0137 *HSC mount or dismount WTOR message***Message Level:** 4

Explanation: A mount or dismount was performed on a remote HSC server with the LIBRARY WTORdest=Client set. HSC issued a WTOR that is to be handled by the client. The SMC0135 message indicates the LIBRARY name and SERVER name where the mount or dismount was requested.

System Action: The SMC redrives the mount or dismount to the server, transmitting the specified message response.

User Response: See the *HSC Messages and Codes Guide* for the corresponding HSC message, and respond as directed.



Note: The MSGDef command SMCWtorpfx parameter can be used to suppress the SMC0137 portion of the message.

xxxxxxx SMC0140 DFSMS message**Message Level:** 0

Explanation: The DFSMS ACS routine has issued the DFSMS message. xxxxxxxx will be the JCL statement number or the DDNAME if it is a dynamic allocation.

System Action: None.

User Response: None.

xxxxxxx SMC0141 UNIT= value yyyyyyy replaced by zzzzzzz**Message Level:** 0

Explanation: The DFSMS ACS routines has changed the esoteric.

- xxxxxxxx is the JCL statement number or the DDNAME if it is a dynamic allocation.
- yyyyyyy is the esoteric that was in the JCL. If a UNIT parameter was not used in the JCL, this will be NULL-UNIT.
- zzzzzzz is the esoteric that was supplied by the DFSMS ACS routines.

System Action: Use the new esoteric supplied by the DFSMS ACS routines for the allocation.

User Response: None.

SMC0142 Invalid reply to prior message from LIBRARY=*libraryname*
SERVER=*servername*

Message Level: 0

Explanation: The WTORDEST(CLIENT) option was specified for LIBRARY *libraryname*, and SMC has communicated a mount or dismount request to HSC via server *servername*. When SMC attempted to redrive the request, HSC returned an indication that the response was invalid.

System Action: SMC redisplay the HSC message.

User Response: Refer to the HSC Messages and Codes Guide for the valid responses for the indicated message.

SMC0143 Response {timeout|wait cancelled by SMC STOP} for job=*jobname*
volser=*volser* device=XXXX from LIBRARY=*libraryname*
SERVER=*servername*

Message Level: 0

Explanation: SMC has communicated a request to LIBRARY *libraryname* via SERVER *servername*. Either the response timed out, or the SMC subsystem received an MVS STOP command.

System Action: None.

User Response: Check the network and server subsystem for problems. If the mount was not performed, issue a manual mount command to the HSC.

SMC0144 Mount of volser=*volser* on device=XXXX; drive is in use; reply 'C'
to cancel mount or 'R' to retry

Message Level: 0

Explanation: SMC has communicated a mount or dismount request to HSC. HSC has responded with a message indicating that the drive is in use.

System Action: If the mount is still outstanding, the SMC automatically redrives the mount every 30 seconds. The message is DOMed if the mount is no longer pending, or the drive becomes available.

User Response: Wait for the drive to become available, or reply 'C' to cancel the request, or 'R' to retry immediately.

SMC0145 Mount of volser=*volser* on device=*XXXX*; volume is in use; reply 'C' to cancel mount or 'R' to retry

Message Level: 0

Explanation: SMC has communicated a mount or dismount request to HSC. HSC has responded with a message indicating that the drive is in use.

System Action: The SMC verifies that the mount is still outstanding. If the mount is still outstanding, the SMC automatically redrives the mount every 30 seconds. The message is DOME if the mount is no longer pending, or the volume becomes available.

User Response: Wait for the volume to become available, or reply 'C' to cancel the request, or 'R' to retry immediately.

SMC0146 All devices marked ineligible for job *JJJJJJJJ* step *SSSSSSSS* due to volume lookup failure

Message Level: 4

Explanation: When SMC allocation attempted to acquire volume information, a communication failure with the library server occurred. The ALLOCDEF option FAILNOINFO(ON) was specified.

System Action: The SMC marks all devices as ineligible for allocation. The job is failed by MVS or express-canceled by JES3.

User Response: Investigate the cause of the communication failure and resubmit the job. In a multiple library environment, where one or more libraries are unavailable, you can use the LIBRARY DISABLE command to prevent the SMC from trying to access the unavailable library.

SMC0170 {Request timeout|SMC subsystem terminating}; request to subsystem *SSSS* aborted

Message Level: 4

Explanation: The SMC issued a request to subsystem *SSSS*, but no response was received within the time limit, or the SMC subsystem was terminated.

System Action: The SMC continues processing without waiting for the HSC response. In addition, if a request timeout is indicated, the SMC SERVER path to the specified subsystem is disabled.

User Response: Check the status of the subsystem that is not responding to SMC, and correct the problem. After correcting the problem, enable the appropriate SERVER path.

SMC0171 Allocatability test matrix retries exceeded

Message Level: 0

Explanation: The SMC could not determine allocatability of the job step using its test matrix in the calculated number of retries.

System Action: Normally, processing continues, with SMC “backing out” exclusion conditions until the job step is allocatable.

During JES3 CI esoteric substitution, if SMC is unable to solve its allocatability matrix after reverting to the original list of eligible devices, an abend results, and no esoteric substitution is performed.

User Response: If possible, rerun the job with allocation trace turned on. Save trace and log output and contact Sun Microsystems StorageTek Software Support.

SMC0174 ASCOMM ACK timeout for job=JJJJJJJJ (DS=nnnn-nnnn)

Message Level: 0

Explanation: The SMC replied to a request from job *JJJJJJJJ*, but no acknowledgment was received within the time limit.

System Action: The SMC continues processing without waiting for the acknowledgment.

User Response: None.

SMC0175 Communication initialized on LIBRARY=*name* {SERVER=*name*}

Message Level: 8

Explanation: The SMC has successfully communicated with the specified LIBRARY for the first time. Additionally, if the communication path selected was a remote SERVER, then the SERVER is also displayed.

System Action: Processing continues.

User Response: None.



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

SMC0230 IEFJFRQ exit smcxJFRQ is inactive. SMC is unable to influence tape allocation

Message Level: 0

Explanation: Exit routine smcxJFRQ has reached its error threshold and has been made inactive by the operating system.

The smcx in smcxJFRQ will be replaced with the subsystem name belonging to SMC. IBM message CSV430I was issued when the routine was made inactive. An SVC dump of the job most likely occurred along with message CSV430I.

The dump title will be:

```
DUMP TITLE=COMPON=SSI,COMPID=5752SC1B6,ISSUER=IEFJSARR,
MODULE=IEFJRASP,ABEND=aaaa,REASON=rrrrrrrr
```

System Action: SMC is unable to direct tape allocation to the correct drive. SMC will delete the SMC0230 each minute and re-check the state of exit routine. If the exit routine is still inactive SMC0230 will be re-issued.

User Response: Investigate the reason that the exit became inactive. To display the exit use the following MVS command:

```
DISPLAY PROG,EXIT,EXITNAME=IEFJFRQ,DIAG
```

The state of the exit can be made active by one of the following methods:

1. Use the MVS SETPROG command:

```
SETPROG EXIT,MODIFY,EX=IEFJFRQ,MOD=smcxJFRQ,STATE=ACTIVE
```

2. Stop and re-start SMC.

Retain the SVC dump and job log of the failing job. Contact Sun Microsystems Software support for analysis of the failure.

SMC9999 *MMMMMMMM* Variable text

Message Level: 12, 16, 20, 24, 28

Explanation: SMC9999 messages are intended for Sun Microsystems StorageTek Technical Support problem determination and resolution. *MMMMMMMM* is the name of the issuing module.

System Action: None.

User Response: None. A message level (LVL) of 12 or higher should generally be specified only when directed by Sun Microsystems StorageTek Technical Support.

SMC Utility Message Listing

SMCU0001 SMC is not active or not JES3; unable to produce report

Explanation: The SMCUPJS utility was submitted on a system without an active SMC subsystem, or on a system that is not JES3.

System Action: Report processing terminates with a return code of 8.

User Response: If the host system is JES3, start the SMC subsystem and resubmit the utility job.

SMCU0002 Utility release level *n.n* is incompatible with SMC release *n.n*

Explanation: The SMCUPJS utility load module is not at the same release level as the SMC subsystem on the host.

System Action: Report processing terminates with a return code of 8.

User Response: Resubmit the utility job with the release level matching the SMC subsystem.

SMCU0003 No active libraries; unable to determine drive characteristics

Explanation: The SMCUPJS utility was submitted on a system without an active HSC or MVS/CSC subsystem, and no defined active remote library, or the SMC subsystem has not yet processed an allocation or message request.

System Action: Report processing terminates with a return code of 8.

User Response: Start the HSC or MVS/CSC subsystem, or define a remote HSC subsystem using the LIBRARY and SERVER commands. Then issue the RESYNC command and resubmit the job.

SMCU0004 *** WARNING: HCD esoteric not found

Explanation: The SMCUPJS utility found an esoteric defined by the JES3 SETUNIT statement that had no corresponding HCD esoteric.

System Action: Report processing continues, but a return code of 4 is returned.

User Response: Research and correct the discrepancy.

- SMCU0005** *** WARNING: HCD esoteric does not match JES3 esoteric
- Explanation:** The SMCUPJS utility found an esoteric that contains different devices in its HCD and JES3 definitions.
- System Action:** The report lists JES3 devices missing from the HCD esoteric and HCD devices missing from the JES3 esoteric. Report processing continues, but a return code of 4 is returned.
- User Response:** Research and correct the discrepancy.
- SMCU0006** *** WARNING: XTYPE contains inconsistent location or drive characteristics
- Explanation:** Drives contained within an XTYPE do not have the same location type (library, virtual, nonlibrary, or unknown), the same location (ACS or VTSS), or the same recording technique.
- System Action:** Report processing continues, but a return code of 4 is returned.
- User Response:** Review the Device to XTYPE report to determine the inconsistency, and correct the discrepancy.
- SMCU0007** *** WARNING: XTYPE contains unknown or MODEL(IGNORE) devices
- Explanation:** One or more of the drives within an XTYPE are either UNKNOWN, MODEL(IGNORE), or both. However, one or more drives in the XTYPE are NOT UNKNOWN or MODEL(IGNORE).
- System Action:** Report processing continues, but a return code of 4 is returned.
- User Response:** Verify that the XTYPEs are defined as intended.
- SMCU0008** SMCUDBX parameter error
- Explanation:** An error was detected in the input PARM for the SMCUDBX utility. A detailed description of the error follows.
- System Action:** The utility processing terminates with a return code of 8.
- User Response:** Correct the indicated error and resubmit the job.

SMCU0009 Unable to load TMS interface routine
{SLUDRCA1|SLUDRTL|SLUDRRMM|SLUDRZAR}

Explanation: Explanation: Based on the input TMS parameter, the SMCUDBX utility attempted to load the corresponding tape management access routine, but the load failed.

System Action: The utility processing terminates with a return code of 8.

User Response: Ensure that a load library containing the appropriate SLUDR* routine for your tape management system is available to the SMCUDBX utility through a JOBLIB, STEPLIB, or MVS LINKLIST library.

SMCU0010 Error opening file DDNAME {DBTMS|DBEXTFIL|

Explanation: Explanation: The SMCUDBX program or tape management extract program was unable to open the tape management data base file or extract file.

System Action: System Action: The utility processing terminates with a return code of 8.

User Response: User Response: Correct the execution JCL to provide the required DD statement.

SMCU0011 Error processing ZARA interface for subsystem CCCC

Explanation: The user requested an extract from the ZARA tape management system with subsystem ID CCCC. A detailed description of the error follows the message.

System Action: The utility processing terminates with a return code of 8.

User Response: Correct the indicated error and resubmit the job.

SMCU0012 Unexpected return code XXXX from TMS interface

Explanation: An unexpected return code XXXX was received from the TMS extract routine.

System Action: The utility processing terminates with a return code of 8.

User Response: If user modifications have been made to the TMS extract routine, correct the routine. Otherwise, contact Sun Microsystems StorageTek Software Support for assistance.

SMCU0013 No VLF control record found by SLUDRTLM

Explanation: The user specified TLMS as the SMCUDBX tape management system, but the input file did not contain a CA-DYNAM/TLMS VLF record.

System Action: Utility processing terminates with a return code of 8.

User Response: Verify that the correct input file was specified on the DBTMS DD statement, and resubmit the job.

Chapter 11. StorageTek HTTP Server Messages

Overview

This chapter provides information about StorageTek HTTP server messages. See Chapter 2, “SMC and StorageTek Library Management” for more information about the HTTP Server.

Message Format

Messages are displayed on the console in the following format:

SKYnnnn message-text

where:

- *SKY* identifies the StorageTek HTTP server.
- *nnnn* is the four-character message number.
- *message-text* is the actual text displayed on the job log or system log.

StorageTek HTTP Server Message Listing

The following HTTP Server messages are listed numerically.

- SKY001I** *name Server version starting on system running MVS level*
Explanation: This is the HTTP server startup message
- SKY002E** Error binding server socket - terminating
Explanation: The server cannot bind to the specified network port. The server shuts down.
- SKY003I** *name ready to accept requests*
Explanation: The server name is ready for normal operation.
- SKY004E** Error opening server socket - terminating
Explanation: The server cannot open a socket connection to the required network port. The server shuts down.
- SKY005E** Server select failed rc=*ret* err=*msg*
Explanation: An error occurred while listening for a client connection.
- SKY006I** TCP/IP connection terminated
Explanation: The client connection was terminated.
- SKY007E** Accept error - shutting down
Explanation: There was an error in network connection accept processing. The server shuts down.
- SKY008E** Error reading request
Explanation: A network error occurred while reading a client request.
- SKY013I** Shutdown command from operator acknowledged
Explanation: The server has received a console command to shut down.

SKY016I name Server *ver* started at *tttt*
 "requests received: *nnn*
 "tasks default: *dd* active: *aa* limit: *ll*

Explanation: Response to an operator display status command and indicates the server name, version, start date and the total number of requests received. It also details the worker task settings, the default number of idle tasks, the maximum limit of dynamically started tasks and the current number of active tasks.

SKY018I Request *n* task completed *rc=ret*

Explanation: The worker task *t* has completed with return code *ret*.

SKY023E Unknown server command: *cmd*

Explanation: The command *cmd* is not a valid console command.

SKY025E Invalid command option: *oooo*

Explanation: The option *oooo* is not a valid option for the operator command.

SKY026I Task: *t* Requests: *n* Client: *xxx.xxx.xxx.xxx* : *pppp*

Explanation: This message is issued in response to a display client console command. Task *t* has serviced *n* requests from the client at IP address *xxx.xxx.xxx.xxx* port *pppp*.

SKY027I No active client connections

Explanation: This message is issued in response to a display client console command, when there are no clients currently connected to the server.

SKY030E Missing to/from translation string

Explanation: The data set file name translation is incorrect.

SKY031I Connections total: *nn* max: *mm/min*

Explanation: This message is issued in response to a display client console command. There have been a total of *nn* connections at a maximum rate of *mm* per minute.

SKY031W Not APF authorized, some facilities not available

Explanation: The HTTP server is not APF authorized.

SKY032I - SKY051E

SKY032I Connection rates: *nn*/min *mm*/hour

Explanation: This message is issued in response to a display client console command. The connection rates are *nn* per minute and *mm* per hour.

SKY040I Option *nnnn* *ssss*

Explanation: The parameter option *nnnn* is set to *ssss*.

SKY043E Terminating execution due to parameter error(s)

Explanation: There was an error in one or more parameters. The server shuts down.

SKY045E SAF authentication requested but not APF authorized

Explanation: SAF authentication requires caller to be APF authorized, but server is not APF authorized. The server shuts down.

SKY046E Supplied *parm* is too long

Explanation: The length of the parameter string *parm* exceeds the allowed length.

SKY047E Unable to open *dsname*

Explanation: The HTTP server was unable to open the data set *dsname* in response to a client request.

SKY048E *abend_message_line*

Explanation: Message number for abend reporting messages.

SKY049E Not APF authorized, cannot continue

Explanation: The server is not APF authorized but APF authorization is required. The server shuts down.

SKY050E Task shutdown time expired, terminating tasks

Explanation: During HTTP server shutdown, some tasks did not stop before the shutdown timeout was exceeded. These tasks are forcibly terminated.

SKY051E Cannot find server module *name*

Explanation: When the HTTP server performed its startup checks, it could not find its module: *name*.

- SKY052E** Cannot find all server modules, shutting down
Explanation: The HTTP server could not find all of its modules when it performed its startup checks. The server shuts down.
- SKY053I** Current active worker tasks: *t*
Explanation: The current number of worker tasks for servicing requests.
- SKY053E** Fatal error in main task, commencing forced shutdown
Explanation: The server suffered a fatal error in its main task and was unable to perform an orderly shutdown.
- SKY054E** Fatal error in main task, attempting orderly shutdown
Explanation: The server suffered a fatal error in its main task and is attempting to perform an orderly shutdown.
- SKY055I** Forcibly terminating task *t*
Explanation: The task *t* is forcibly terminated.
- SKY056I** Disable abend handling option specified, handling disabled
Explanation: The HTTP server abend handlers will not be enabled so any abends can cause dumps and may also shut down the server.
- SKY059E** Error initializing translation tables
Explanation: An error occurred while initializing the server file name translation table.
- SKY060E** SERVERBASE not set, shutting down
Explanation: The required parameter `SERVERBASE` was not set in the parameter file. The server shuts down because it cannot locate its data files.
- SKY061I** Authentication request received from client at *xxx.xxx.xxx.xxx* :
pppp
Explanation: A request to authenticate was received from the client at network address *xxx.xxx.xxx.xxx* and port *pppp*.

SKY100W - SKY109I

- SKY100W** Unable to set socket *option* to *value*
Explanation: The named socket *option* could not be set. Processing continues.
- SKY101W** Unable to set socket option
Explanation: A socket option could not be set.
- SKY102E** Socket *option* error: *explanation*
Explanation: Indicates why a socket option could not be set.
- SKY103E** Cannot open parameter file *dsname*, terminating
Explanation: The HTTP server was unable to open the parameter file *dsname*. The server shuts down.
- SKY104E** Parameter *ppp* invalid
Explanation: The parameter *ppp* is not known.
- SKY105E** Unexpected character *c* encountered, skipping line
Explanation: When parsing the parameter file, an unexpected character was encountered. The parameter is skipped by moving to the next line.
- SKY106E** Expected number but non numeric: *ssss*
Explanation: When parsing the parameter file, a non numeric item was found when a number was expected.
- SKY107I** *parm* set to *vvv*
Explanation: The parameter *parm* is set to value *vvv*.
- SKY108E** Expected string: *ssss*, skipping
Explanation: When parsing the parameter file, the string *ssss* was expected but not found. Parsing skips to the next parameter.
- SKY109I** *pppp* set to *vvv*
Explanation: When reading the parameter file, the parameter *pppp* was set to value *vvv*.

- SKY110E** Unknown token type: *tttt*
Explanation: The parameter token is unknown.
- SKY112E** Cannot create *tbl*, terminating
Explanation: During initialization, the memory table *tbl* could not be allocated. The server shuts down.
- SKY113E** Cannot start task *tttt*, terminating
Explanation: During initialization, the task *tttt* could not be started. The server shuts down.
- SKY114I** Task *tttt* completed, *rc=nn*
Explanation: Task *tttt* completed with return code *nn*.
- SKY115I** Commencing server shutdown
Explanation: The server has started to shut down.
- SKY116E** Task *tttt* unable to get client id
Explanation: An error occurred while passing a socket connection to the work task *tttt*.
- SKY118I** Task *t* waiting for work
Explanation: The worker task *t* is now idle after completing a request.
- SKY119I** Task *t* shutting down
Explanation: Task *t* has commenced shutting down.
- SKY120E** Task *t* unable to allocate dir list buffer
Explanation: The worker task *t* was unable to allocate a memory buffer needed for a directory listing.
- SKY121E** Task *t* unable to open directory readme
Explanation: The worker task *t* encountered an error when attempting to open a directory readme file while listing a directory.

SKY122E - SKY130I

- SKY122E** Task *tt* connection error with *xxx.xxx.xxx.xxx* errno *error_text*
- Explanation:** The worker task *t* encountered a connection error (error number *errno*, text *error_text*) with the client at IP address *xxx.xxx.xxx.xxx*.
- SKY123E** Task *t*, connection eof from *xxx.xxx.xxx.xxx*
- Explanation:** For worker task *t*, the connection to client at IP address *xxx.xxx.xxx.xxx* ended.
- SKY124I** Task *t* connection terminated by peer *adr*
- Explanation:** The network connection to worker task *t* was terminated by the client with IP address *adr*.
- SKY125E** Task *t cmp*, abend *aaa* caught
- Explanation:** The abend handler for task *t* in component *cmp*, intercepted a type *aaa* abend. Task *t* is shut down and a replacement task is created.
- SKY126E** Task *t cmp*, illegal instruction abend caught. Code *aaa*
- Explanation:** The abend handler for task *t* in component *cmp*, intercepted an illegal instruction abend of type *aaa*. Task *t* will be shut down and a replacement task created.
- SKY127E** Task *t cmp* memory access abend caught. Code *aaa*
- Explanation:** The abend handler for task *t* in component *cmp*, intercepted a memory abend of type *aaa*. Task *t* is shut down and a replacement task is created.
- SKY128E** Task *t* request error *nnn* msg: *description*
- Explanation:** For worker task *t*, an HTTP protocol error was encountered for a client request.
- SKY129E** *ppp* out of valid range min - max
- Explanation:** The numeric parameter *ppp* is outside the allowed range.
- SKY130I** *hhh* handler ready
- Explanation:** The handler *hhh* task has finished initializing and is ready for work.

SKY131I *hhh* handler shutting down

Explanation: The handler *hhh* task has started shutting down.

SKY132E *hhh* handler startup timeout, terminating

Explanation: The startup timeout for the handler task *hhh* has been exceeded. The handler will be terminated.

SKY133E Unable to start *hhh* handler, terminating

Explanation: The HTTP server was unable to start its handler task *hhh*. The server shuts down.

SKY134E Unable to allocate *mmm* memory, terminating

Explanation: During initialization, memory could not be allocated for *mmm* memory. The server shuts down.

SKY135E Unable to open log, reason: *rrr*

Explanation: The log handler task was unable to open the log file for reason *rrr*.

SKY136W Out of memory for stack space, requested *nnnn* bytes

Explanation: An HTTP server task was unable to allocate stack memory.

SKY138W No free worker tasks, at maximum limit

Explanation: A request has been received but there are no idle worker tasks and the number of tasks is at the maximum limit. A server busy request is returned to the client. If this condition occurs frequently consider increasing the maximum limit of worker tasks to a value where this is a rare occurrence.

Appendix A. Intercepted Messages

IBM Operating System Messages

The messages listed in Table 8 are received by the SMC. IBM message manuals describe the exact format (spacing, etc.) and definition for each message. Ellipses are used to indicate that the message contains more text than is shown.

Correct operation of the SMC depends on these messages. Do not suppress or alter them using products designed to handle messages through the subsystem interface (SSI). Many automated operations systems make use of the subsystem interface to intercept and alter or suppress messages.



Note: If the “suppressed by subsystem” and “hardcopy only” bits are turned on in the WQE (the MVS write queue element) before the SMC has received the message, the SMC ignores the WTO, and the message is not displayed at the console.

If you plan to use an automated operations system and are unsure about how it intercepts messages, contact the product vendor.

Although these messages may be suppressed (that is, prevented from displaying in the console) by using MPFLSTxx parameters or an MPF exit, the text of these messages should not be changed. The use of other WTO exits to change the display characteristics or the text of these messages is not supported by the SMC.

Volume serial numbers (‘ser’) as specified in messages from the operating system, are defined as follows:

SCRATCH: a public scratch volume is to be mounted (nonspecific request).

PRIVATE: a private scratch volume is to be mounted (nonspecific request).

xxxxxx: volume serial number to be processed (specific request)

Messages that contain VOLSERS with more than six characters or any character except A-Z, 0-9, # (crosshatch), \$, ¥ (yen character), and optional trailing blanks are ignored by the SMC.

Table 8. Intercepted Operating System Messages

Message ID	Description
IEC068A	U dddd,ser
IEC101A	M dddd,ser,...
IEC111E	D dddd,ser
IEC114E	D dddd...
IEC135A	U dddd,ser...
IEC400A	M dddd, ser...
IEC401A	F dddd,ser...
IEC501A	M dddd,ser{,labtyp}
IEC501E	M dddd,ser{,labtyp}
IEC502E	n,dddd,ser...
IEC509A	F dddd,ser...
IEC512I	I/O ERR LBL ERR SEC VOL...
IEC701D	M dddd, VOLUME TO BE LABELED ser
IEC702I	dddd, VOLUME LABELS CANNOT BE VERIFIED
IEC703I	dddd, VOLUME IS FILE PROTECTED
IEF233A	M dddd,ser{,labtyp}
IEF233D	M dddd,ser{,labtyp}
IEF234E	{K D R} dddd{,ser...}
IGF500I	SWAP dddd to eeee - OPERATOR I/O ERROR
IGF502E	PROCEED WITH SWAP OF dddd TO eeee
IGF503I	ERROR ON dddd, SELECT NEW DEVICE
IGF509I	SWAP ddd - OPERATOR I/O ERROR
IGF511A	WRONG VOLUME MOUNTED ON dddd, MOUNT ser,...
_TA0233D	Message for ASM2

JES3 Messages

The following JES3 messages are processed by the SMC:

- IAT5210
- IAT5310
- IAT5410

IBM message manuals describe the exact format (spacing, etc.) and definition of each message.

Tape Management System Messages

CA-1 Messages

The following CA-1 (TMS) messages are received by the SMC. Refer to the appropriate Computer Associates publication for the exact format and meaning of each message.

Table 9. Tape Management System Messages - CA-1

Message ID	Description
CTS001	See CA-1 User Manual, Volume 1
CTS002	See CA-1 User Manual, Volume 1
CTS004	See CA-1 User Manual, Volume 1
CTS005	See CA-1 User Manual, Volume 1
CTS007	See CA-1 User Manual, Volume 1
CTS008	See CA-1 User Manual, Volume 1
CTS009	See CA-1 User Manual, Volume 1
CTS010	See CA-1 User Manual, Volume 1
CTS011	See CA-1 User Manual, Volume 1
CTS014	See CA-1 User Manual, Volume 1
CTS015	See CA-1 User Manual, Volume 1
CTT100A	See CA-1 User Manual, Volume 1
CTT101A	See CA-1 User Manual, Volume 1
CTT102A	See CA-1 User Manual, Volume 1
CTT103A	See CA-1 User Manual, Volume 1
CTT104A	See CA-1 User Manual, Volume 1
CTT105A	See CA-1 User Manual, Volume 1
TMS001	See CA-1 User Manual, Volume 1
TMS002	See CA-1 User Manual, Volume 1

Table 9. Tape Management System Messages - CA-1 (Continued)

Message ID	Description
TMS004	See CA-1 User Manual, Volume 1
TMS005	See CA-1 User Manual, Volume 1
TMS007	See CA-1 User Manual, Volume 1
TMS008	See CA-1 User Manual, Volume 1
TMS009	See CA-1 User Manual, Volume 1
TMS010	See CA-1 User Manual, Volume 1
TMS011	See CA-1 User Manual, Volume 1
TMS014	See CA-1 User Manual, Volume 1
TMS015	See CA-1 User Manual, Volume 1
IECTMS7	See CA-1 User Manual, Volume 1
CA\$F810A	See CA-1 Message Guide
CA\$F813A	See CA-1 Message Guide

CONTROL-M/TAPE (formerly CONTROL-T) Messages

The following CONTROL-M/TAPE messages are received by the SMC. Refer to the appropriate BMC publication for the exact format and meaning of each message.

Message ID	Description
CTT100A	See BMC's INCONTROL for OS/390 and z/OS Message Manual
CTT101A	See BMC's INCONTROL for OS/390 and z/OS Message Manual
CTT102A	See BMC's INCONTROL for OS/390 and z/OS Message Manual
CTT103A	See BMC's INCONTROL for OS/390 and z/OS Message Manual
CTT104A	See BMC's INCONTROL for OS/390 and z/OS Message Manual
CTT105A	See BMC's INCONTROL for OS/390 and z/OS Message Manual

DFSMSrmm Messages

DFSMSrmm mount message (EDG6627A) must be acted on by the SMC when the volume or drive specified in the message is under the control of the SMC. The action of the SMC is similar to the SMC actions for the normal MVS mount messages (e.g., IEC233A, etc).

The DFSMSrmm Tape Initialization program (EDGINERS) issues a series of messages describing the success or failure of tape initialization, erasure, and/or verification. This series of messages is used to drive the dismount of the tapes mounted from the EDG6627A message. Refer to Table 10 for the messages that must be acted on by the SMC to dismount a tape.

Table 10. Tape Management System Messages - DFSMSrmm

Message ID	Description
EDG6620I	VOLUME volser INITIALIZATION AND VERIFICATION SUCCESSFUL
EDG6621E	VOLUME volser INITIALIZATION FAILED
EDG6623I	VOLUME volser ERASE, INITIALIZATION AND VERIFICATION SUCCESSFUL
EDG6624I	VOLUME volser ERASE FAILED
EDG6627E	M dev VOLUME (volser) RACK (rack-number) TO BE action, lbtype
EDG6642E	VOLUME volser LABELLED SUCCESSFULLY
EDG6643E	VOLUME volser ERASED AND LABELLED SUCCESSFULLY

Appendix B. Differences Between SMC and NCS 4.1 and Below

The SMC product represents a complete re-architecture of the MVS interfaces for StorageTek tape library management systems. Customers converting to SMC from HSC or MVS/CSC releases 4.1 or below should be aware that minor differences to allocation and message processing may be found with SMC. In a few cases Sun Microsystems recommends changes to existing HSC policies and parameters to preserve behavior consistent with older releases. Significant differences are noted in this appendix.

Allocation Differences

1. SMC allocation is based on exclusion levels (see Chapters 4 and 5 for a detailed description) and patented allocatability testing algorithms. In all cases, the SMC attempts to ensure that the most important policies and criteria are applied first. For example, the SMC ensures that a specific volume is never allocated to an incompatible drive type. Other policies (those above "minimum level") are considered optional. The SMC will ignore optional policies in order to be able to allocate the job.
2. In JES2, SMC allocation uses documented operating system interfaces (IEFSSI, IEFSSSTA, IEFJFRQ) to perform allocation influencing. For the most part, if no parameter overrides such as MIACOMPAT or CAIRTS are specified, the SMC performs its modifications at SSI 78 (MVS tape allocation) rather than updating the Eligible Device List (EDL) at SSI 24. In some cases other vendors' products rely on the StorageTek EDL modifications at SSI 24 to work correctly. You can specify MIACOMPAT(ON) to force the SMC to update the EDL as earlier NCS releases did.
3. SMC policies use logical defaults, including:
 - Always separate affinity chains
 - Always honor user exit and TAPEREQ policies for head-of-chain when an affinity chain is not separated
 - Allow allocation to fail when a volume can not be mounted
 - Do not attempt to influence allocation if the original list of eligible drives does not contain any drive known by Sun Microsystems.

For most customers, these changes eliminate the need to specify previously required policies such as UNITAFF(SEP) and UX02/08(HONOR).

4. SMC supports the ability to specify any esoteric for substitution in a user exit or TAPEREQ provided that:

- Devices cannot be added to the original list of eligible drives
- The specified esoteric must be a valid esoteric, and must contain one or more drives in common with the original list of eligible drives.

Customers with TAPEREQs or user exits with specified esoterics that were invalid in previous releases may experience unexpected results when these esoterics are honored by SMC.

5. In SMC virtual tape is treated more like a unique device type and may be used when policies include both real and virtual drives. For example, if a subpool including real and virtual volumes is specified, SMC allocation may select virtual drives.
6. SMC ignores allocations for tape DDs when all devices in the EDL are undefined to any library subsystem. This behavior allows better coexistence between StorageTek software and other vendors' products that influence tape allocation. In previous releases, customers were sometimes instructed to include non-StorageTek drives in the HSC nonlibrary esoteric. With SMC, Sun Microsystems recommends that the HSC nonlibrary esoteric include only StorageTek devices outside the library, and if necessary, include HSC UNITATTRs to specify MODEL(IGNORE) for non-existent drives outside the library.
7. The SMC does not use the HSC-defined nonlibrary esoteric to influence allocations of volumes outside the library. Instead, the SMC uses all HSC UNITATTRs to determine characteristics of drives outside the library. The HSC-defined nonlibrary esoteric is used in SMC 6.0 only to trigger implementation of the HSC SPECVOL policy. Note that virtual drives should not be included in the nonlibrary esoteric for either JES2 or JES3 installations.
8. A single version of an allocation user exit is executed, regardless of the number of libraries defined. Therefore, user exit return codes such as “USE LIBRARY DRIVES”, “USE VIRTUAL DRIVES”, etc. are **not** used to determine a request owner, nor does the user exit return code or “USE NONLIBRARY DRIVES” result in the selection of a “different” library.

The preferred way to select a library owner from among multiple libraries is through esoteric substitution on TAPEREQ. Alternatively, user exits can be used to specify a library name or esoteric.



Warning: If your pre-NCS 6.0 user exit uses the “USE NONLIBRARY DRIVES” return code to select an MVS/CSC subsystem in a mixed HSC and MVS/CSC environment, you must now use one of the methods described above to properly select a library.

SMS Differences

Beginning in SMC 5.1 the ability to use DFSMS to influence allocation was implemented in SMC and removed from HSC and MVS/CSC. The SMC invocation of DFSMS routines is supported using standard IBM interfaces, that is, SSI 55. This change requires that the StorageTek ACS routines (&ACSENVIR=STKTAP1) receive control after IBM ACS routines (&ACSENVIR=ALLOC), rather than before, as was the case in HSC and MVS/CSC 4.1 and below. If your IBM DFSMS routines apply a Storage Class to a data set, the data set is now SMS-managed, and StorageTek DFSMS routines will not be in effect for this data set.

Message Handling Differences

- The SMC message handling component does not call user exit 01 for all console messages. The USERMsg operator command allows specification of additional messages to be intercepted and passed to the library subsystem's user exit 01.
- For swap processing, only the enhanced swap was implemented in SMC. Customers previously doing custom swap processing by defaulting to HSC non-enhanced swap and intercepting IBM swap messages may need to examine their processing and possibly intercept SMC messages.

Client/Server Differences

In an MVS-only environment SMC provides the ability to communicate directly with HSC systems on other hosts. Since the format of communications between hosts is identical to that between SMC and HSC on the same host, customers who implement the SMC client/server feature should see no differences between HSC running on the same host as SMC or running on a different host, with a few minor exceptions:

- Any HSC acting as a server for SMC clients running on other hosts must define all drives used by any host in its host CDS drive definition, and all drive addresses on client and server hosts must be identical, that is, the same drive may not have different drive addresses between a client host and a server host. It is not necessary for all drives defined on the server host to have channel connections to the server host.
- Policies and drive definitions are not automatically updated when the SMC is communicating with a remote HSC. If policies or drive definitions on the HSC host are changed, an SMC RESYNC command must be issued on the client host to refresh the policy and drive definitions.

Appendix C. Message Change Summary

The following messages have been added, changed, or deleted for SMC Release 6.0.

New Messages

SMC Messages

- SMC0051
- SMC0067
- SMC0085
- SMC0086
- SMC0087
- SMC0088
- SMC0093
- SMC0094
- SMC0113
- SMC0114
- SMC0115
- SMC0116
- SMC0117
- SMC0118
- SMC0119
- SMC0120
- SMC0121
- SMC0122
- SMC0123
- SMC0124
- SMC0125
- SMC0126
- SMC0127
- SMC0128
- SMC0129
- SMC0130
- SMC0131
- SMC0132
- SMC0133
- SMC0134
- SMC0135
- SMC0136
- SMC0137
- SMC0140
- SMC0141
- SMC0142
- SMC0143
- SMC0144
- SMC0145
- SMC0146
- SMC0170
- SMC0171
- SMC0174
- SMC0175
- SMCU0008
- SMCU0009
- SMCU0010
- SMCU0011
- SMCU0012
- SMCU0013

StorageTek HTTP Server Messages

- SKY001I
- SKY002E
- SKY003I
- SKY004E
- SKY005E
- SKY006I
- SKY007E
- SKY008E
- SKY013I
- SKY016I
- SKY018I
- SKY023E
- SKY025E
- SKY030E
- SKY031W
- SKY043E
- SKY045E
- SKY046E
- SKY047E
- SKY049E
- SKY050E
- SKY051E
- SKY52E
- SKY053E
- SKY054E
- SKY055I
- SKY056I
- SKY059E
- SKY060E
- SKY061I
- SKY100W
- SKY101W
- SKY102E
- SKY103E
- SKY104E
- SKY105E
- SKY106E
- SKY108E
- SKY109I
- SKY110E
- SKY115I
- SKY116E
- SKY117I
- SKY118I
- SKY119I
- SKY120E
- SKY121E
- SKY124I
- SKY125E
- SKY126E
- SKY127E
- SKY129E
- SKY130I
- SKY131I
- SKY132E
- SKY133E
- SKY135E
- SKY136W
- SKY138W

Changed Messages

- SMC0006
- SMC0008
- SMC0014
- SMC0022
- SMC0025
- SMC0029
- SMC0030
- SMC0041
- SMC0044
- SMC0049
- SMC0063
- SMC0066
- SMC0068
- SMC0077
- SMC0089
- SMC0090
- SMC0113
- SMC0122
- SMC0123
- SMC0128
- SMC0133
- SMCU003

Deleted Messages

- SMC0051
- SMC0105
- SMC0138
- SMC0139

Index

A

- affinity separation
 - JES2 40
 - JES3 75
- allocation
 - JES2
 - affinity separation 40
 - deferring mounts 45
 - drive exclusion 26
 - drive prioritization 44
 - exclusion levels 29, 34
 - intentionally failing a job 28
 - minimum exclusion level, JES2 27
 - scratch requests 34
 - SMC interaction with other software 55
 - specific requests 29
 - JES3
 - affinity separation 75
 - constraints 95
 - deferring mounts 84
 - device preferencing consideration 93
 - drive exclusion 61
 - drive prioritization 83
 - esoteric unit name replacement 79
 - exclusion levels 64, 69
 - initialization parameter considerations 85
 - intentionally failing a job 63
 - JES3 managing drives 58
 - JES3 not managing drives 58
 - minimum exclusion level 62
 - scratch requests 69
 - SMC interaction with other software 109
 - specific requests 64
 - suppressing fetch messages 82
 - virtual drive considerations 93
 - ZEROSCR considerations 93
- ALLOCDef command 120
- ALLOCJob command 123
- audience, intended xvii

C

- CA-1 (TMS) messages 243

- CMDDef command 127
- command prefix, SMC 118
- commands
 - issuing 118
 - ALLOCDef 120
 - ALLOCJob 123
 - CMDDef 127
 - LIBrary 129
 - LISt 133
 - MSGDef 135
 - MSGJob 137
 - READ 140
 - RESYNChronize 141
 - SERVer 142
 - SMSDef 147
 - TCPip 151
 - TRace 152
 - UEXit 157
 - USERMsg 160
- comments, reader xxvi
- configuration scenarios
 - SMC and a single HSC library configuration with multiple remote server paths 11
 - SMC and a single HSC library configuration with one local and one remote path 9
 - SMC and a single local HSC library 5
 - SMC and a single remote HSC library configuration 7
 - SMC with two libraries 13, 15
- conventions
 - control statements xxiv
 - product names xviii
 - symbols xviii
 - syntax flow diagrams xix
 - typographic xviii

D

- deferring mounts
 - JES2 45
 - JES3 84
- description, SMC 1
- device preferencing consideration, JES3 93
- DFSMSrmm messages 245

differences, SMC and NCS 4.1 and below 247
drive exclusion 1, 26, 61
drive prioritization 2
 JES2 44
 JES3 83

E

EDTOX report, JES3 Configuration 180
EETOX report, JES3 Configuration 178
esoteric unit name replacement, JES3 79
examples
 ALLOCDef command 122
 ALLOCJob command 126
 CMDDef command 128
 EDTOX report 180
 EETOX report 178
 EXTOD report 179
 EXTOE report 177
 EXTON report 177
 JES3 DEVICE statements 87
 JES3 high watermark setup 96
 JES3 HWSNAME statements 92
 JES3 SETNAME statements 89
 Llst command 131, 134
 MSGDef command 136
 MSGJob command 139
 READ command 140
 SMC START Procedure 20
 SMSDef command 150
 TRace command 155
 UEXit command 159
 USERMsg command 160
exclusion levels, scratch request
 JES2 34
 JES3 69
exclusion levels, specific request
 JES2 29
 JES3 64

EXEC Statement, SMC 21
EXTOD report, JES3 Configuration 179
EXTOE report, JES3 Configuration 177
EXTON report, JES3 Configuration 177

F

fetch messages, suppressing 82
flow diagrams, conventions xix

H

help, technical support xxvi
HSC mount-related messages 114

HTTP Server for OS/390 and z/OS 3
 messages 231

I

IBM (International Business Machines, Inc.)
 operating system messages 241
initialization parameter considerations, JES3 85
initialization, SMC 19
intentionally failing a job 28
intercepted messages
 IBM operating system messages 241
 JES3 messages 243
 tape management system messages 243
introduction, SMC 1

J

JES2
 allocation
 affinity separation 40
 deferring mounts 45
 drive exclusion 26
 drive prioritization 44
 exclusion levels 29, 34
 intentionally failing a job 28
 minimum exclusion level 27
 scratch requests 34
 SMC interaction with other software 55
 specific requests 29
JES3
 allocation
 affinity separation 75
 constraints 95
 deferring mounts 84
 device preferencing consideration 93
 drive exclusion 61
 drive prioritization 83
 esoteric unit name replacement 79
 exclusion levels 64, 69
 initialization parameter considerations 85
 intentionally failing a job 63
 JES3 managing drives 58
 JES3 not managing drives 58
 minimum exclusion level 62
 scratch requests 69
 SMC interaction with other software 109
 specific requests 64
 suppressing fetch messages 82
 virtual drive considerations 93
 ZEROSCR considerations 93
JES3 Configuration Report Utility 176
JES3 Converter/Interpreter (C/I) 59
JES3 Main Device Scheduler (MDS) 59

job, intentionally failing
 JES2 28
 JES3 63

L

LIBrary command 129
library subsystem selection 45
library, defined 1
LlSt command 133

M

message handling, SMC 111
messages 114
 descriptions 182
 DFSMSrmm messages 245
 format 181, 231
 HTTP server 231
 intercepted 241
 messages intercepted by HSC 241
 SMC listing 183
 SMC system 181
 StorageTek HTTP Server 232
 tape management system messages 243
 utility 226
 variable definitions 182
minimum exclusion level
 JES2 27
 JES3 62
Modify command, MVS 118
mounts, managing from the SMC client 115
MSGDef command 135
MSGJob command 137
MVS START Command 22

O

one local and one remote 13
one local and one remote with multiple paths 15
open type J macro precautions 109
operator commands
 issuing 118
 ALLOCDf 120
 ALLOCJob 123
 CMDDef 127
 LIBrary 129
 LlSt 133
 MSGDef 135
 MSGJob 137
 READ 140
 RESYNChronize 141
 SERVer 142

SMSDef 147
TCPip 151
TRace 152
UEXit 157
USERMsg 160

organization of this Guide xvii

P

parameters
 ALLOCDf command 120
 ALLOCJob command 123
 CMDDef command 127
 LIBrary command 130
 LlSt command 133
 MSGDef command 135
 MSGJob command 137
 MVS START Command 22
 READ command 140
 SMC EXEC Statement 21
 SMSDef command 148
 TCPip command 151
 TRace command 153
 UEXit command 158
 USERMsg command 160
publications, related xxv

R

READ command 140
reader's comments xxvi
recovery procedures 161
related publications xxv
reports, JES3 Configuration 177
RESYNChronize command 141

S

scratch requests
 JES2 34
 JES3 69
SERVer command 142
SMC
 configuration scenarios 5
 differences from NCS 4.1 and below 247
 functions 1
 initializing 19
 interaction with other software, JES2 55
 interaction with other software, JES3 109
 JES2 allocation 23
 JES3 allocation 57
 library subsystem selection 16
 message handling 111
 messages 181

- operator commands 117
- policy and drive type information synchronization 16
- recovery procedures 161
- starting 19
- utilities 167
- SMCCMDS data set 119
- SMCPARMS data set 119
- SMSDef command 147
- specific requests
 - JES2 29
 - JES3 64
- SSI23 JES3 Dynamic Allocation 59
- SSI24 common allocation 24, 59
- SSI55 Interpreter/Dynamic Allocation Exit (IDAX) 24, 58
- SSI78 tape allocation 24
- START Command, MVS 22
- START procedure, creating 20
- starting SMC 19
- subsystem selection 16
- support, technical xxvi
- swap processing, SMC 113
- syntax
 - ALLOCDf command 120
 - ALLOCDJob command 123
 - CMDDef command 127
 - conventions xix
 - LIBrary command 129
 - List command 133
 - MSGDef command 135
 - MSGJob command 137
 - MVS START Command 22
 - READ command 140
 - RESYNchronize command 141
 - SMC EXEC Statement 21
 - SMSDef command 147
 - TRace command 152
 - UExit command 157
 - USERMsg command 160

T

- Tape Management Extract Utility 168
- Tape Management System (TMS)
 - messages 243
 - supported 113
- TCPip command 151
- technical support xxvi
- TRace command 152
- troubleshooting xxvi

U

- UExit command 157
- USERMsg command 160
- utilities
 - JES3 Configuration Report Utility 176
 - Tape Management Extract Utility 168

V

- virtual drive definition considerations 93
- volume serial numbers, defined 241

Z

- ZEROSCR considerations, JES3 93



Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA Phone 1-650-960-1300 or 1-800-555-9SUN Web sun.com



ARGENTINA: 5411-4317-5636 • AUSTRALIA: 1-800-550-786 • AUSTRIA: 43-1-601-26-0 • BALKANS: 301-6188-111 • BELGIUM: 32 2-704 89 83 • BRAZIL: 55-11-51872100 • BRUNEI: 65-216-8333 • CANADA: 1-800-422-8020 (GENERAL); 416-964-2001 (LEARNING MANAGEMENT SYSTEM SALES, TORONTO) • CHILE: 562-372-4500 • COLOMBIA: 571-629-2323
CZECH REPUBLIC: 420 2 33009311 • DENMARK: 45 4556 5040 • EGYPT: 00 202 570 9442 • FINLAND: 358-9-525-551 • FRANCE: 33-1-41-33-17-17 • GERMANY: 49-89-460-08-2788 • GREECE: 30-01-6188101 • HONG KONG: 852-2877-7077 • HUNGARY: 361-202-4415 • INDIA: 91-80-229-8989 • INDONESIA: 65-216-8333 • IRELAND: 353-1-668-4377
ISRAEL: 972-9-9710500 • ITALY: 39-02-9259511 • JAPAN: 81-3-5779-1820 • KOREA: 82-2-3453-6602 • MALAYSIA: 603-2116-1887 • MIDDLE EAST: 00 9714 3366333 • MEXICO: 525-261-0344 • NETHERLANDS: 31-33-4515200 • NEW ZEALAND: 0800-786-338 • NORTH WEST AFRICA: 00 9714 3366333 • NORWAY: FROM NORWAY: 47-22023950, TO NORWAY: 47-23369650 • PAKISTAN: 00-9714-3366333 • PEOPLE'S REPUBLIC OF CHINA: 8610-6803-5588 • PHILIPPINES: 632-885-7867 • POLAND: 48-22-8747848 • PORTUGAL: 351-21-413-4000 • RUSSIA: 7-095-935-8411 • SAUDI ARABIA: 00 9714 3366333 • SINGAPORE: 65-216-8300 • SOUTH AFRICA: 27-11-256-6300 • SPAIN: 34-902-210-412 • SRI LANKA: 65-2168333 • SWEDEN: 46 8 631 22 00 • SWITZERLAND: 41-1-908-90-50 (GERMAN) 41-22-999-0444 (FRENCH) • TAIWAN: 886-2-25185735 • THAILAND: 662-344-6855 • TURKEY: 90 212 335 22 00 • UNITED KINGDOM: 44-1276-416-520 • UNITED STATES: 1-800-422-8020 • VENEZUELA: 582-905-3800 • VIETNAM: 65-216-8333 • WORLDWIDE HEADQUARTERS: 1-650-960-1300

SUN™ THE NETWORK IS THE COMPUTER ©2006 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, and the Sun logo are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.