

# StorageTek Client System Component for MVS Environments

---

System Programmer's Guide

Version 7.0

**ORACLE**

Part Number: E38050-02  
November 2012

Submit comments about this document to [STP\\_FEEDBACK\\_US@ORACLE.COM](mailto:STP_FEEDBACK_US@ORACLE.COM).

Oracle welcomes your comments and suggestions for improving this book. Contact us at [STP\\_FEEDBACK\\_US@ORACLE.COM](mailto:STP_FEEDBACK_US@ORACLE.COM). Please include the title, part number, issue date, and revision.

Copyright © 2009, 2012, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

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

# Contents

---

<b>Preface</b>	<b>11</b>
<b>1. Introduction</b>	<b>19</b>
Overview	19
MVS/CSC Operating Environment	20
MVS/CSC Basic Functions	20
MVS/CSC System Interfaces	21
MVS/CSC Configurations	21
Security Administration Considerations for Communication	23
StorageTek Library Control System (LCS) Software Products	23
Third-Party Software Interaction	25
Tape Management Systems	25
Multi-image Manager (MIM)	25
Data Facility Hierarchical Storage Manager (DFHSM)	25
System Authorization Facility (SAF)	26
MVS/CSC Interaction With Fault Analyzer for z/OS	26
Communication Methods	27
User Policy Definition for Mixed Media and Devices	30
Device Preferencing	30
DFSMS/MVS Storage Management Subsystem Support	30
<b>2. Tape Management System Interface</b>	<b>31</b>
Overview	31
Synchronizing your Tape Management System with the ACS	31
Cartridge Status Changes	31

	Specification of Scratch Pools	32
	Tape Management Systems Supported by MVS/CSC	33
<b>3.</b>	<b>Communications and Operator Console Interfaces</b>	<b>35</b>
	Overview	35
	Communications Interface	35
	Operator Console Interface	36
	HSC Commands	37
	CLS Commands	38
	Operator Response	38
<b>4.</b>	<b>Programmatic Interface</b>	<b>39</b>
	Overview	39
	Command Syntax	39
	PGMI Interface Processing	40
	Multiple MVS/CSCs and the PGMI	41
	Authorization Checking for the PGMI	41
	SCSXREQ Macro Format	42
	Specifying the List Form	42
	Specifying the Execute Form	42
	SCSXREQ Return Specifications	43
	PGMI Supported Requests	46
	QCSC	46
	QDRIVES	48
	QVOLUME	50
	READ	52
	Sample PGMI Requests	54
	Sample QVOLUME Request (Single MVS/CSC Environment)	54
	Sample QVOLUME Request (Multiple MVS/CSC Environment)	59
<b>5.</b>	<b>MVS/CSC Utility Functions</b>	<b>65</b>
	Overview	65
	Control Statement Syntax Conventions	66
	Command Syntax	67
	SCUADMIN Utility Program	68

SCUADMIN PARM Parameter	69
SCUADMIN Return Codes	70
Event Log Report Utility (VM-based LCS only)	71
LOGRPT Control Statement	72
Configuration Verification Utility	75
CONFIGV Control Statement	75
Scratch Update Utility	80
SCRATCH and UNSCRATCH Control Statements	80
Scratch Conversion Utility	83
Using SCUCONDB with CA-1 Release 4.9 or Below	84
Using SCUCONDB with CA-DYNAM/TLMS Release 5.3 or Below	85
Specifying Values for the PARM Parameter	85
Parameter Descriptions	86
Sample JCL	88
Sample Output	89
<b>6. Performance and Operational Considerations</b>	<b>91</b>
Overview	91
Performance Considerations	91
Redistributing Scratch Cartridges	91
Maintaining Quantities of Scratch Cartridges	92
Reducing Scheduling Contention	92
Reducing Cartridge Exchange Activity	93
Operational Constraints	93
Cartridge Naming Restrictions	93
Transport Address Restrictions	94
Association with a Library Control System (LCS)	94
Sharing MVS/CSC-controlled Transports	94
Sharing Transports Between MVS/CSC Clients on Different MVS Host Systems	95
Sharing Transports Between MVS/CSC Clients and Non-MVS Clients	95
Sharing Transports Between MVS/CSC and HSC's Residing on Different MVS Systems	95
<b>A. Programmatic Interface SCSXREQM Macro Mappings</b>	<b>97</b>

SCSXREQM Macro Mappings 98

**B. Gathering Diagnostic Materials 113**

Overview 113

MVS Diagnostic Materials 113

Tape Format 114

**C. List of Abbreviations 115**

**Glossary 119**

**Index 135**

# Figures

---

FIGURE 1-1	MVS/CSC to LCS Configuration	22
FIGURE 1-2	Communications Using TCP/IP and SNA LU6.2 (Unix-Based LCS)	28
FIGURE 1-3	Communications Using TCP/IP and VTAM “3270 BISYNC” (VM-Based LCS)	29
FIGURE 4-1	QVOLUME Request (Single MVS/CSC) (1 of 5)	54
FIGURE 4-2	QVOLUME Request (Single MVS/CSC) (2 of 5)	55
FIGURE 4-3	QVOLUME Request (Single MVS/CSC) (3 of 5)	56
FIGURE 4-4	QVOLUME Request (Single MVS/CSC) (4 of 5)	57
FIGURE 4-5	QVOLUME Request (Single MVS/CSC) (5 of 5)	58
FIGURE 4-6	QVOLUME Request (Multiple MVS/CSCs) (1 of 6)	59
FIGURE 4-7	QVOLUME Request (Multiple MVS/CSCs) (2 of 6)	60
FIGURE 4-8	QVOLUME Request (Multiple MVS/CSCs) (3 of 6)	61
FIGURE 4-9	QVOLUME Request (Multiple MVS/CSCs) (4 of 6)	62
FIGURE 4-10	QVOLUME Request (Multiple MVS/CSCs) (5 of 6)	63
FIGURE 4-11	QVOLUME Request (Multiple MVS/CSCs) (6 of 6)	64
FIGURE 5-1	Sample JCL for Invoking Multiple Utility Functions	68
FIGURE 5-2	Sample JCL for Event Log Report Utility	73
FIGURE 5-3	Event Log Report Utility Sample Output-Input Control Statement	74
FIGURE 5-4	Event Log Report Utility Sample Output with VM-Based LCS-Formatted Event Log	74
FIGURE 5-5	Sample JCL for Configuration Verification Utility	76
FIGURE 5-6	Configuration Verification Utility Sample Output-Input Control Statement	77
FIGURE 5-7	CONFIGV Sample Output - All Parameters Valid (VM-Based LCS with TCP/IP)	78
FIGURE 5-8	CONFIGV Sample Output - Some Invalid Parameters (VM-Based LCS with TCP/IP)	79
FIGURE 5-9	Sample JCL for Scratch Update utility	81
FIGURE 5-10	Scratch Update Utility Sample Output-Input Control Statement	82
FIGURE 5-11	Scratch Update Utility Sample Output-Messages	82

FIGURE 5-12	Sample JCL for Scratch Conversion Utility	88
FIGURE 5-13	Scratch Conversion Utility Sample Output - Messages	89



# Tables

---

TABLE 4-1	Return Code Specifications	44
TABLE 4-2	Reason Code Specifications	45
TABLE 5-1	SCUADMIN Return Codes	70
TABLE A-1	SCSXREQM Macro	98



# Preface

---

This guide provides configuration and administration information for Oracle's StorageTek Client System Component for MVS Environments (MVS/CSC) software. It is intended for storage administrators, system programmers and operators responsible for configuring and maintaining MVS/CSC.

---

## Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support:

<http://www.oracle.com/support/contact.html>

<http://www.oracle.com/accessibility/support.html> (for hearing impaired)

---

## Related Documentation

### Oracle's StorageTek Client System Component for MVS Environments (MVS/CSC)

- *MVS/CSC Configuration Guide*
- *MVS/CSC Messages and Codes Guide*
- *MVS/CSC Operator's Guide*
- *MVS/CSC Syntax Quick Reference*

### Oracle's StorageTek Enterprise Library Software (ELS)

- *Introducing ELS*
- *Installing ELS*
- *ELS Command, Control Statement, and Utility Reference*
- *ELS Syntax Quick Reference*
- *ELS Messages and Codes*
- *ELS Programming Reference*
- *ELS Legacy Interfaces Reference*
- *Configuring HSC and VTCS*
- *Managing HSC and VTCS*
- *Configuring and Managing SMC*
- *ELS Disaster Recovery and Offsite Data Management Guide*

### Oracle's StorageTek Automated Cartridge System Library Software (ACSL) Publications for the UNIX-Based LCS

- *ACSL Installation, Configuration and Administration Guide*
- *ACSL Messages*
- *ACSL Reference*

---

# Conventions for Reader Usability

## Typographic

Some JCL examples in this guide include *italic* type. Italic type is used to indicate a variable. You must substitute an actual value for these variables.

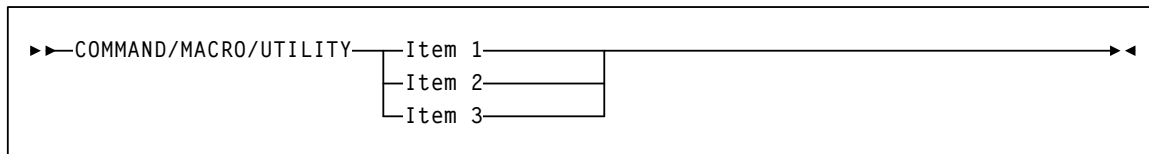
The use of mixed upper and lower case characters for commands, control statements, and parameters indicates that lower case letters may be omitted to form abbreviations. For example, you may simply enter POL when executing the POLicy command.

## Syntax Flow Diagrams

Syntax flow diagramming conventions include the following:

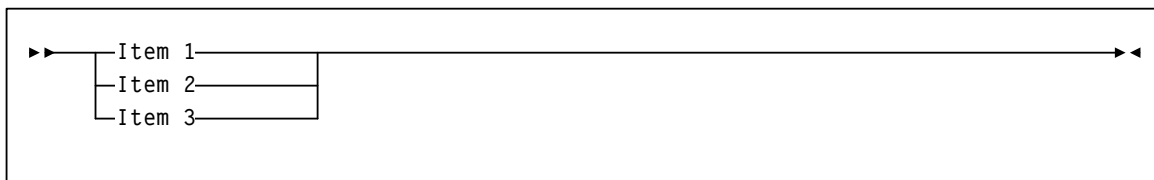
### Flow Lines

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



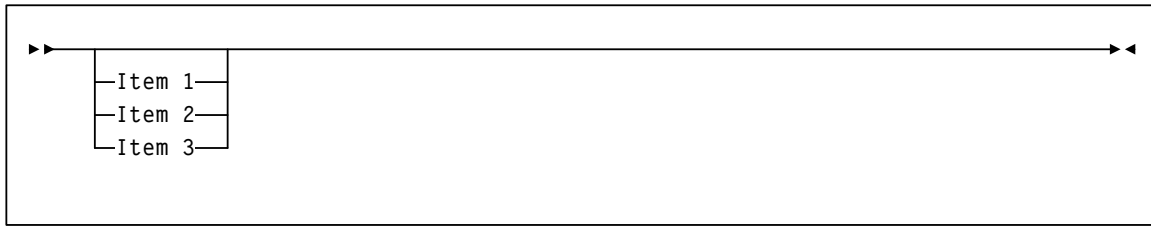
### Single Required Choice

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



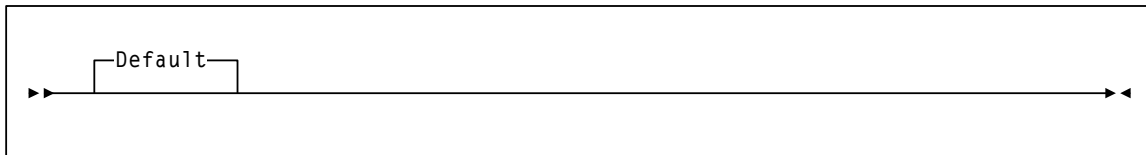
## Single Optional Choice

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

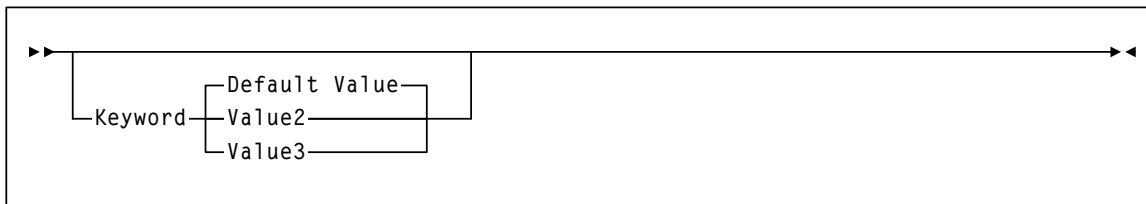


## Defaults

Default values and parameters appear above the baseline.

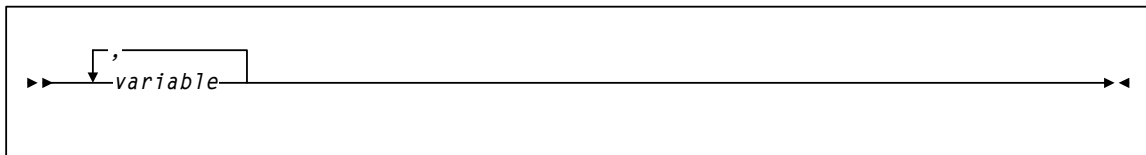


Some keyword parameters provide a choice of values in a stack. When the stack contains a default value, the keyword and the value choices are placed below the baseline to indicate that they are optional, and the default value appears above the keyword line.



## 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 following example indicates that a comma is required as the repeat delimiter.



## Keywords

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

## Variables

Italic type is used to indicate a variable.

## Alternatives

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

## Optional

Brackets [ ] are used to indicate that a command parameter is optional.

## Delimiters

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

## Ranges

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

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

A decimal range consists of a pair of decimal numbers (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, any portion is valid as the incremental range. For example:

A00B00                    the largest range that can be specified is A00B00 through A99B99.  
A0B0CC                    the largest range that can be specified is A0B0CC through A9B9CC.  
000XXX                    the largest range that can be specified is 000XXX through 999XXX.

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

A00A0-A99A0                    increments VOLSERs A00A0 through A09A0, then A10A0 through A99A0.  
9AA9A-9ZZ9A                    increments VOLSERs 9AA9A through 9AZ9A, then 9BA9A through 9ZZ9A.  
111AAA-111ZZZ                    increments VOLSERs 111AAA through 111AAZ, then 111ABA through 111ZZZ  
999AM8-999CM8                    increments VOLSERs 999AM8 through 999AZ8, then 999BA8 through 999CM8  
A3BZZ9-A3CDE9                    increments VOLSERs A3BZZ9 through A3CAA9, then A3CAB9 through A3CDE9  
AAAAAA-AAACCC                    increments VOLSERs AAAAAA through AAAAAZ, then AAAABA through AAACCC  
CCNNN-DDDNNN                    increments VOLSERs CCNNN through CCCNNZ, then CCCNOA through DDDNNN \*

\* **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. A control statement is terminated if the statement is not continued.
- /\* and \*/ can be used to enclose comments in the job stream. HSC PARMLIB members and definition data sets must specify comments in this format.
  - A comment is not required as the first control statement of any PARMLIB member.
  - Comments can be continued over multiple lines, but cannot be nested.
  - The maximum length for any control statement is 1024 characters.



# Introduction

---

---

## Overview

MVS/CSC provides client functions and communications between an MVS host and the Library Control System (LCS) or server residing on a Unix or VM host.

When combined with the LCS and SMC, the MVS/CSC provides the following benefits:

- a library shared by multiple host systems (both IBM and non-IBM)
- secondary library attachment for remote backup
- library attachment to more than sixteen MVS hosts, with MVS/CSC installed on each attached host system

---

**Note** – MVS/CSC 7.0 is not compatible with Oracle's StorageTek LibraryStation. In an MVS-only environment, you must use Oracle's StorageTek SMC and its HTTP server subtask to provide communication between MVS hosts. Refer to the publication *Configuration and Managing SMC* for more information.

---

This chapter summarizes the following MVS/CSC features and functions:

- MVS/CSC operating environment
- MVS/CSC basic functions
- MVS/CSC system interfaces
- MVS/CSC configurations
- StorageTek LCS software products
- Third-party software products that coexist with MVS/CSC
- Communications methods used to transmit commands to the LCS
- Mixed media and devices for the UNIX-based LCS

# MVS/CSC Operating Environment

MVS/CSC operates with the Storage Management Component (SMC) on any IBM or compatible processor running MVS (any IBM-supported version of z/OS). The MVS/CSC supports both JES2 and JES3 environments. Except for noted differences, the information in this publication applies to both JES2 and JES3 environments.

In addition, references in this publication to JES2 apply to both JES2 environments and JES3 environments that run without TAPE SETUP processing; references to JES3 apply only to JES3 environments that run with TAPE SETUP processing.

---

## MVS/CSC Basic Functions

The MVS/CSC's primary functions are to provide user policy information to the SMC and to transmit information requests and directives to the appropriate LCS.

The SMC manages the following functions:

- Drive allocation
- Processing of Mount, Dismount, and Swap messages on MVS systems. If a message requests an MVS/CSC drive, the SMC routes the request to the MVS/CSC.

Refer to the publication *Configuring and Managing SMC* for more information.

Once the cartridge is mounted, the data is transferred using the data path under the control of the MVS client operating system.

Depending on the configuration, the MVS/CSC communicates with the LCS using one of the following communications methods:

- Virtual Telecommunications Access Method (VTAM) "3270 BISYNC"
- Systems Network Architecture Logical Unit 6.2 (SNA LU 6.2)
- Transmission Control Protocol/Internet Protocol (TCP/IP)

The MVS/CSC translates each request to the command format appropriate for the LCS.

In addition to basic functions provided to start and stop the MVS/CSC software, the MVS/CSC provides diagnostic aids (event logging and tracing), utility functions, user exits, and recovery processing. The MVS/CSC also provides an operator interface on MVS consoles through which you can issue commands to MVS/CSC. For the VM-based LCS, commands can be forwarded to the CLS or VM/HSC using the communications link.

---

# MVS/CSC System Interfaces

The MVS/CSC consists of the following system interfaces:

- Tape management system interfaces to communicate with your tape management system
- Communications interfaces to link the MVS/CSC to the LCS for sending and receiving messages
- Operator console interfaces to allow operator commands to be issued for the MVS/CSC
- Programmatic interface to allow programs to request certain services from the MVS/CSC (UNIX-based LCS only)

---

# MVS/CSC Configurations

The MVS/CSC program runs as a subsystem on the IBM MVS operating system along with the SMC subsystem.

MVS/CSC can coexist with the MVS Host Software Component (MVS/HSC) on the same MVS host, thus providing access to multiple libraries from a single MVS host environment. This allows the MVS/HSC to control a local primary library complex<sup>1</sup> while one or more MVS/CSC subsystems access secondary, possibly remote libraries.

When multiple MVS/CSCs (or an HSC with one or more MVS/CSCs) exist on the same MVS host, the SMC on this host determines whether to use the HSC or any of the MVS/CSCs to process a particular allocation or Mount/Dismount/Swap message event. Refer to the publication *Configuring and Managing SMC* for more information.

Each MVS/CSC can communicate with only one LCS at a time. In turn, each LCS manages a single library complex. Multiple MVS/CSC subsystems can exist on a single MVS host system, and each MVS/CSC subsystem can be attached to a different LCS. MVS/CSC supports the following LCS platforms:

- UNIX-based
- VM-based

---

1. A library complex consists of one HSC Control Data Set (CDS) and a maximum of 256 Automatic Cartridge Systems (ACSs). Each ACS can contain a maximum of 24 Library Storage Modules (LSMs).

The following figure illustrates a basic client-server configuration using TCP/IP as the communications method.

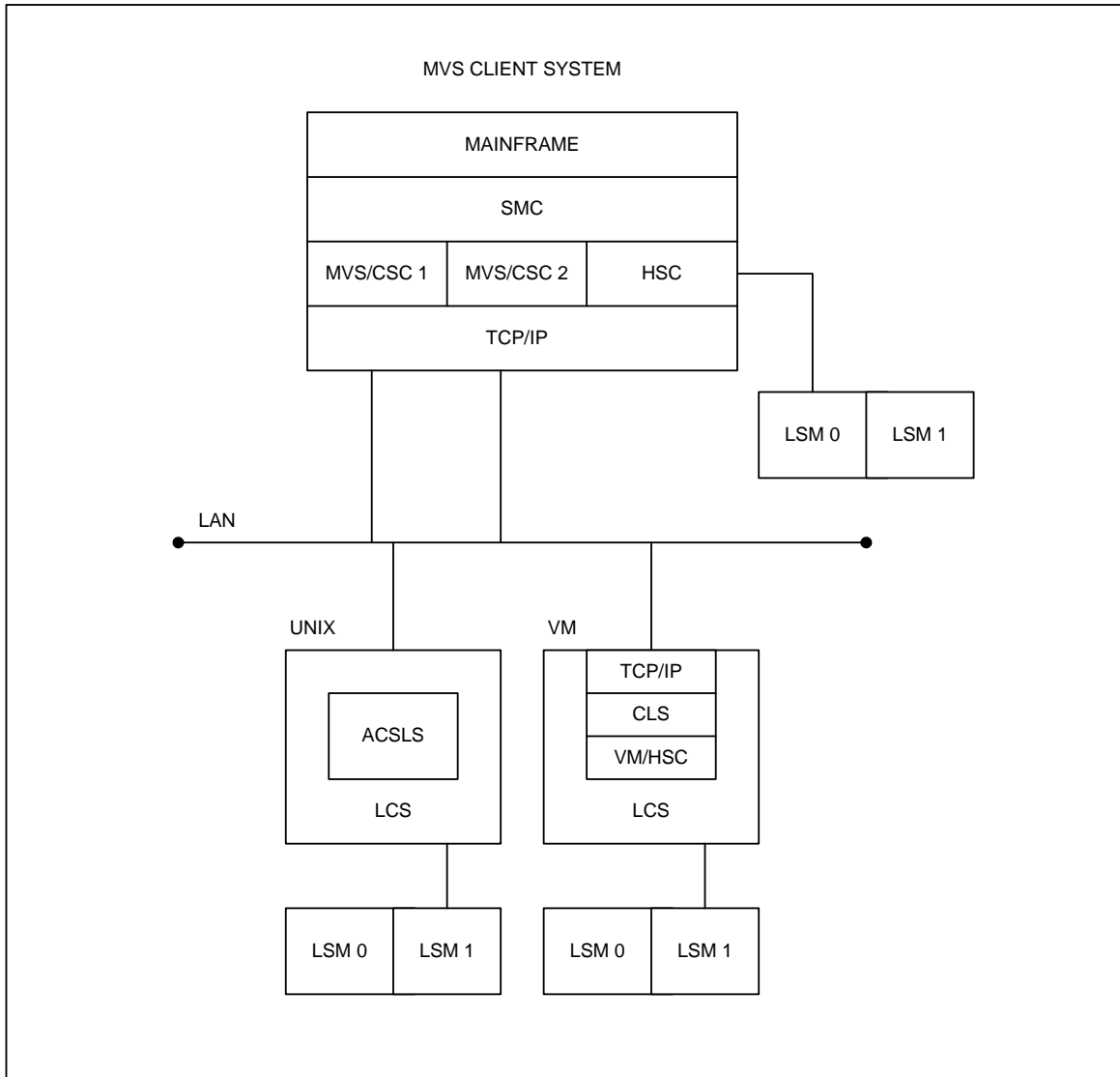


FIGURE 1-1 MVS/CSC to LCS Configuration

---

# Security Administration Considerations for Communication

z/OS users must define an OMVS segment in RACF for the userid associated with MVS/CSC. If this is not done, a UNIX process initialization failure occurs. To define the OMVS segment, refer to the *IBM Communications Server IP Migration Guide*.

If you are running a functionally equivalent security product (e.g., ACF2), refer to the documentation for that product.

---

## StorageTek Library Control System (LCS) Software Products

The StorageTek LCS is the control interface between the mainframe computer systems (client systems) and the StorageTek library products. The LCS consists of hardware and software products that are attached to the MVS/CSC through a communications link.

The MVS/CSC receives requests from the SMC or the MVS host system and translates them to messages, which it sends to the LCS. The LCS receives the requests from the MVS/CSC to perform the automated handling of library cartridges. The LCS directs and monitors a single library and manages message and request traffic from one or more connected client systems. The LCS determines where the cartridge resides.

The LCS controls the library and manages the library database, which contains volume location and volume attribute information for all cartridges within the library. The LCS also performs activities such as mounting, dismounting, and entering and ejecting cartridges. The Library Management Unit (LMU) manages the movement (or exchanges) of cartridges between the Library Storage Modules (LSMs).

The MVS/CSC can be attached to either of the following LCSs:

- UNIX-based LCS, which consists of the Automated Cartridge System Library Software (ACSL)
- VM-based LCS, which consists of the Host Software Component for VM (VM/HSC) and the Common Library Services (CLS)

Each LCS is described in more detail in the following sections.

---

**Note** – Refer to the publication *Installing ELS* for information about ELS hardware and software requirements.

---

## UNIX-Based LCS

The UNIX-based LCS consists of the StorageTek ACSLS software product. ACSLS consists of a system administration component, interfaces to client system applications, and library management facilities that support the entire family of Nearline Automated Cartridge Systems.

The UNIX-based LCS resides on a UNIX-based platform. The MVS/CSC using the UNIX-based LCS requires that the ACSLS software be installed.

## VM-based LCS

The VM-based LCS consists of the following StorageTek software products:

- Oracle's StorageTek Host Software Component for VM (VM/HSC)
- Oracle's StorageTek Common Library Services (CLS)

**Host Software Component (HSC)** controls the ACS. It runs as a VM application on the VM-based LCS. The library database records cell status, characteristics, and disposition of all cartridges stored in the library.

**Common Library Services (CLS)** provides the communications interface between the client system (in this case MVS) and the VM/HSC. The CLS receives client requests and translates them to a form that can be executed by the HSC.

The VM-based LCS resides on an IBM System 370 processor running the Virtual Machine (VM) operating system. The MVS/CSC using the VM-based LCS requires that the CLS and VM/HSC software be installed.



---

## Third-Party Software Interaction

The MVS/CSC subsystem operates in conjunction with various other third-party software, including:

- CA-1 (TMS) and CA-DYNAM/TLMS Tape Management Systems
- Data Facility Hierarchical Storage Manager (DFHSM)
- MIM
- AutoMedia (Zara) Tape Management System
- Any System Authorization Facility (SAF) compliant software product

---

**Note** – Only those third-party software products known to coexist with MVS/CSC are listed above.

---

### Tape Management Systems

The MVS/CSC provides support for the following tape management products:

- CA-1
- CA-DYNAM/TLMS (Tape Library Management System)
- AutoMedia (Zara)

Interaction with tape management systems is managed by the Storage Management Component (SMC). Refer to the publication *Configuring and Managing SMC* for more information.

### Multi-image Manager (MIM)

MIM is a third-party software product that is used in a multi-CPU environment to control the allocation of transports to a particular host. The MVS/CSC can coexist with MIM. However, you must follow certain procedures when using MIM with the MVS/CSC. Refer to the *MVS/CSC Configuration Guide* for information about MIM restrictions.

---

**Note** – With MIM Release 2.0, there are no restrictions for startup and no restrictions on MIM features.

---

### Data Facility Hierarchical Storage Manager (DFHSM)

The MVS/CSC supports the use of 3480, 3490, 3490E, 3590, and helical-type transports by DFHSM. MVS/CSC supports dynamic allocation of cartridge transports by DFHSM.

## System Authorization Facility (SAF)

The MVS/CSC operates with and does not compromise the integrity of any security facility using the SAF interface.

---

# MVS/CSC Interaction With Fault Analyzer for z/OS

The IBM program Fault Analyzer for z/OS is used to determine why an application abends. It may be installed on systems that also run StorageTek ELS software, however, **it is not useful when applied to abends that occur in ELS code**. Because of the complex subsystem environment where ELS code executes, Fault Analyzer itself may abend.

If Fault Analyzer for z/OS is installed on your ELS system, it is **strongly recommended** that you specify the following update to ensure that this product ignores ELS product abends.

When Fault Analyzer is installed, perform the following update to SYS1.PARMLIB(IDICNF00):

```
EXCLUDE (NAME (HSC) NAME (SMC) NAME (CSC) )
```

where:

- *HSC* is the name of the HSC console-started-task
- *SMC* is the name of the SMC console-started-task
- *CSC* is the name of the MVS/CSC console-started-task.

Alternatively, you can specify **EXCLUDE (TYPE(STC))** to exclude all console-started tasks from evaluation by Fault Analyzer. However, this broad exclusion may not be appropriate in your environment.

---

# Communication Methods

The MVS/CSC subsystem is connected to the LCS using a communications link. The following list describes the communications links that can be used to connect the MVS/CSC:

- Transmission Control Protocol/Internet Protocol (TCP/IP) is used by the VM-based and UNIX-based LCS. You can use the following software for TCP/IP communications:
  - IBM TCP/IP
  - CA Unicenter TCPaccess Communications Server
  - CA Unicenter TCPaccess X.25 server
- Virtual Telecommunications Access Method (VTAM) is divided into two categories:
  - VTAM for “3270 BISYNC” communications, which is used only by the VM-based LCS
  - VTAM for SNA LU 6.2 communications, which is used by the UNIX-based LCS

---

**Note** – Refer to the publication *Installing ELS* for supported communication software release levels.

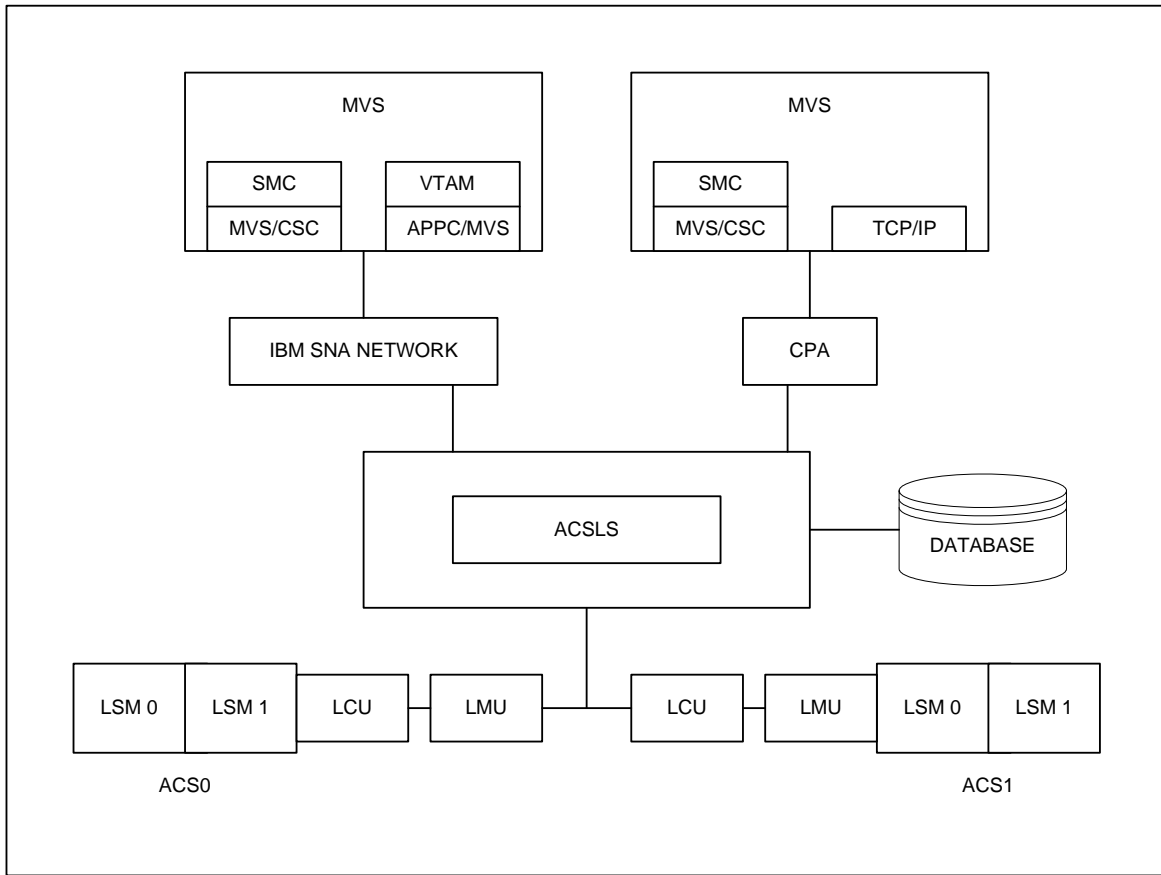
---

The following figure shows the communications connections using the TCP/IP communications protocol and the SNA LU 6.2 communications protocol for a UNIX-based LCS.

---

**Note** – The data path is not shown in this illustration.

---



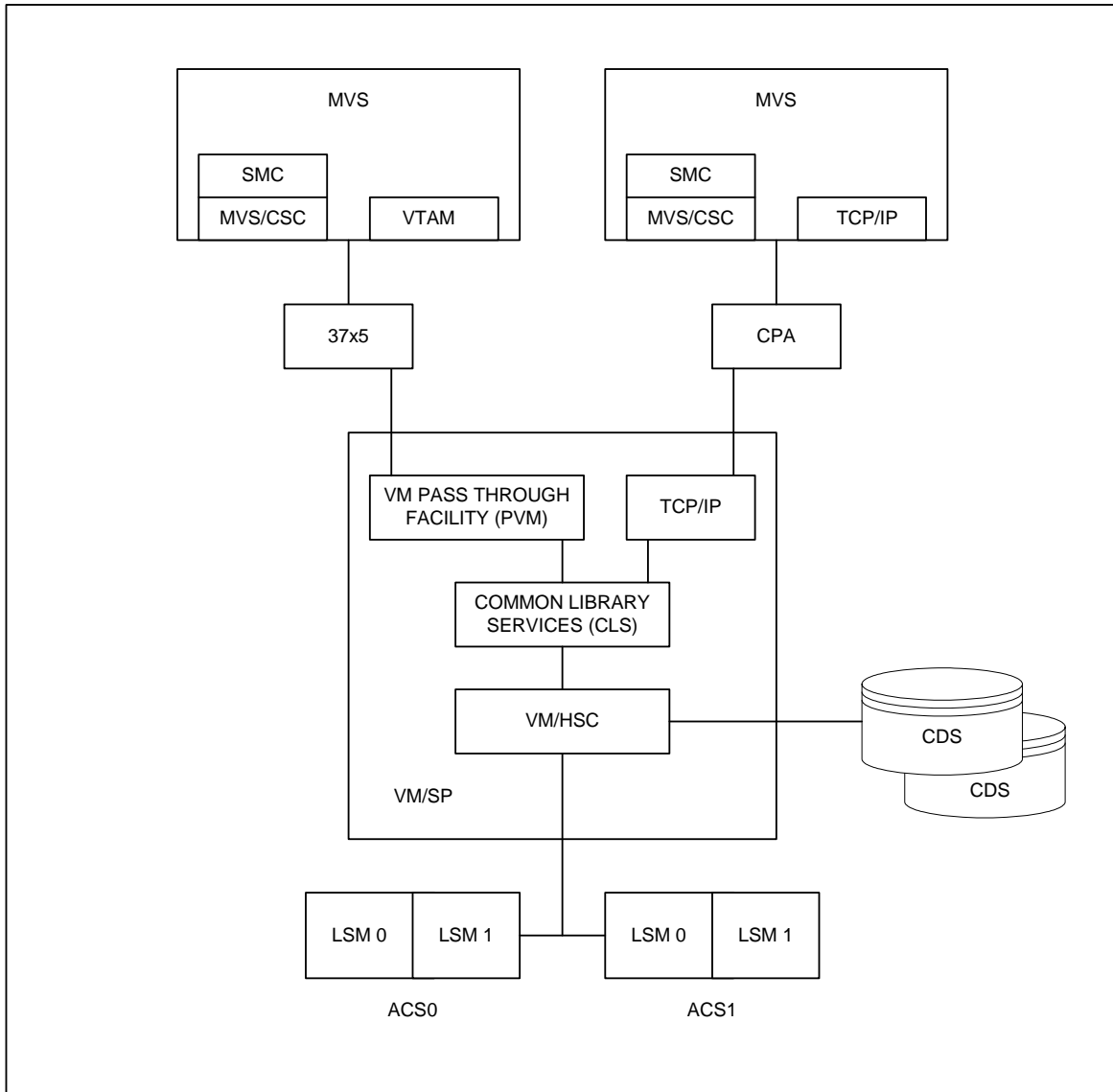
**FIGURE 1-2** Communications Using TCP/IP and SNA LU6.2 (Unix-Based LCS)

The following figure shows the TCP/IP and VTAM “3270 BISYNC” communications protocol for a VM-based LCS.

---

**Note** – The data path is not shown in this illustration.

---



**FIGURE 1-3** Communications Using TCP/IP and VTAM “3270 BISYNC” (VM-Based LCS)

## User Policy Definition for Mixed Media and Devices

The MVS/CSC supports mixed media and mixed cartridge transports in an ACS for the UNIX-based LCS. Mixed media and cartridge transport devices are **not** supported for the VM-based LCS.

The 4400 ACS supports a mixture of transports and associated media, including 4480, 4490, 9490, 9490EE, SD-3 (helical), T9840 series, and T9940 series transports.

The SL8500 ACS supports a mixture of T9840 and T9940 series transports and associated media. These are the **only** transports supported for the SL8500.

The Storage Management Component (SMC) calls on MVS/CSC policies in order to perform drive exclusion and Mount/Dismount/Swap processing in a library environment containing mixed media and cartridge transport devices. This support does not require changes to JCL or the invocation of MVS/CSC user exits. Refer to the publication *Configuring and Managing SMC* for more information.

SMC TAPEREQ control statements are used to specify tape request attributes. These statements are used to place a data set that meets the criteria specified by the TAPEREQ attributes on a specific media type, and create a data set using a specific recording technique. Refer to the *ELS Command, Control Statement, and Utility Reference* for more information about the SMC TAPEREQ control statement.

## Device Preferencing

Device preferencing is applicable only to library configurations containing a mixture of StorageTek's 36-track 4490, 9490, and 9490EE Cartridge Subsystems. It is managed by the Storage Management Component (SMC). Refer to the publication *Configuring and Managing SMC* for more information.

---

# DFSMS/MVS Storage Management Subsystem Support

User policy specification via SMS is supported by the Storage Management Component (SMC). Refer to the publication *Configuring and Managing SMC* for more information.

# Tape Management System Interface

---

---

## Overview

This chapter describes the tape management system interface provided with the MVS/CSC.

---

## Synchronizing your Tape Management System with the ACS

A tape management system is a tool for protecting and organizing tape data in a library. Cartridges with active data are designated nonscratch and are protected against accidental data loss. Cartridges available for output are designated as scratch cartridges. A tape management system maintains a list of scratch cartridges available for use and updates the list when they are used and become unavailable.

The MVS/CSC provides a tape management system interface to allow the MVS/CSC to synchronize your tape management system with the library. To do this, your tape management system must provide the cartridge status changes and specification of scratch pools to the MVS/CSC.

## Cartridge Status Changes

The MVS/CSC provides utilities to retrieve the volume serial numbers of scratch cartridges and to change the scratch status of cartridges defined to a tape management system. The Scratch Conversion utility reads the tape management system database to determine changes in cartridge status. These status changes are transferred to the LCS (using the Scratch Update utility) to maintain synchronization between your tape management system and the library. The status of a cartridge may change from nonscratch to scratch or from scratch to nonscratch.

## Specification of Scratch Pools

Scratch pools are defined through the LCS at initialization time.

- For the UNIX-based LCS, scratch pool information is defined to ACSLS using ACS System Administrator (ACSSA) console commands. Each scratch pool can contain only one label type.
- For the VM-based LCS, scratch pool and label type information is defined to the LCS using the CLS Configuration Management (CLSCM) program. The CLS allows scratch pools with different label types.

MVS/CSC user exits and SMC TAPEREQ parameters allow you to identify a scratch pool. For the VM-based LCS, the user exits allow you to identify a label type for nonspecific requests.

---

### Note –

- Refer to the *ELS Legacy Interfaces Reference* for more information about MVS/CSC user exits.
  - Refer to the *ELS Command, Control Statement, and Utility Reference* for more information about the TAPEREQ control statement.
- 

During nonspecific mount message processing and transport allocation processing, MVS/CSC user exits can be used to supply scratch pool and label type information. You supply a scratch pool identifier and if necessary a label type specification for each scratch cartridge allocation request and mount message request. If this information is not supplied, MVS/CSC uses transports associated with the default scratch subpool. Scratch pool identifiers supplied by the user exit must match the identifiers specified when the LCS was initialized.



---

# Tape Management Systems Supported by MVS/CSC

Any of the following tape management systems can be used with the library and the MVS/CSC:

- A third-party software package
- A tape management system that you have designed and implemented
- A set of manual procedures to manage tape resources

The MVS/CSC is designed to coexist with a variety of tape management systems and does not depend on the design of any one system. This coexistence is accomplished through the utilities and user exits provided by the MVS/CSC.

The CA-1 (TMS), CA-DYNAM/TLMS, DFSMSrmm, and Zara third-party tape management systems can function with the MVS/CSC without modification. Other tape management systems or user-designed systems, including any manual procedures for tape management, might require modification to the MVS/CSC Scratch Conversion utility. The Scratch Conversion utility is provided as an interface between the MVS/CSC and your tape management system. If you need multiple scratch pool and label type definitions, you will also need to modify your user exits and/or TAPEREQ statements.

---

## Note –

- See [“Scratch Conversion Utility” on page 83](#) for more information about the Scratch Conversion utility.
  - Refer to the *ELS Legacy Interfaces Reference* for more information about MVS/CSC user exits.
  - Refer to the *ELS Command, Control Statement, and Utility Reference* for more information about the TAPEREQ control statement.
-



# Communications and Operator Console Interfaces

---

---

## Overview

This chapter describes the communications and operator console interfaces provided with the MVS/CSC.

---

## Communications Interface

The communications interface provides a link between the MVS/CSC and the LCS system. The MVS/CSC uses this link to send messages to the LCS and receive messages from the LCS. The communications link is provided using one of the following communications methods:

- Transmission Control Protocol/Internet Protocol (TCP/IP)  
The MVS/CSC supports CA Unicenter TCPaccess Communications Server, CA Unicenter TCPaccess X.25 Server, and IBM TCP/IP for TCP/IP communications. You must specify the subsystem name or address space name of the TCP/IP stack in the startup parameter file, if the name changed during the installation of the TCP/IP communications software. Otherwise, the default subsystem name or address space name is used.
- Virtual Telecommunications Access Method (VTAM)  
You must specify a VTAM application name in the VTAM configuration tables. The VTAM application name specifies the VTAM application associated with the MVS/CSC.
- Systems Network Architecture Logical Unit 6.2 (SNA LU 6.2)  
You must specify a symbolic destination name in the startup parameter file in order for MVS/CSC to register with APPC/MVS. This name must match the name specified on the DESTNAME keyword in the APPC/MVS side information file defined for the MVS/CSC.

---

**Note –**

- Refer to the publication *Installing ELS* for the supported communications software release levels.
  - Refer to the *ELS Command, Control Statement, and Utility Reference* for information about setting up communications between the MVS/CSC and LCS.
- 

---

## Operator Console Interface

MVS/CSC commands are provided to display MVS/CSC information, alter startup parameter settings, start recovery processing, and enable diagnostic procedures. MVS commands are used to start, stop, cancel, or force the shutdown of MVS/CSC. These commands are issued from the MVS operator console.

For the VM-based LCS, the MVS operator console interface also supports VM/HSC commands as well as CLS commands. You can also receive and reply to HSC and CLS messages. Information in response messages is also displayed on the MVS operator console.

HSC, CLS, and SLK<sup>1</sup> commands must be prefixed by identifiers that explicitly direct the commands to those software components. If a system qualifier is not entered for these commands, the commands are passed in the following order and executed by the appropriate system:

- MVS/CSC
- CLS
- SCP (with prefix SLK)
- HSC

The MVS/CSC interprets commands as they are entered. It executes MVS/CSC commands and sends CLS and HSC command strings to the CLS. The CLS interprets each command, executes CLS commands, and forwards HSC commands to the HSC for execution.

The following sections list the HSC and CLS commands supported by the MVS operator console interface.

---

1. SLK identifies the VM/HSC Control Program

## HSC Commands

The MVS operator console interface supports the following VM/HSC commands:

- Cappref
- Clean
- Dismount
- Display
- Drain
- Eject
- Enter
- Modify
- Mount
- Senter
- Set

Commands intended for the MVS/CSC are not prefixed with a component identifier. Commands intended for the HSC must be prefixed with (1) either the MVS/CSC command prefix character or the MVS/CSC subsystem name *and* (2) the HSC component identifier (HSC). In the following example, the first HSC Display command is prefixed with the MVS/CSC command prefix character (!). The second HSC Display command is prefixed with the MVS/CSC subsystem name (CSC0).

```
!HSC DISPLAY LSM 000
CSC0 HSC DISPLAY LSM 000
```

Because the commands contain the HSC component identifier, the MVS/CSC sends them first to the CLS for interpretation. The CLS, in turn, sends them to the HSC for execution.

The MVS/CSC command shown in the following example is interpreted and executed by the MVS/CSC:

```
CSC0 DISPLAY AVAIL
```

If the command were not an MVS/CSC command, it would be passed on to the CLS. If the command is a CLS command, it is processed by the CLS, otherwise CLS passes the command to the HSC for processing.

For HSC commands that are also MVS/CSC commands, the HSC component identifier must be specified in order to execute the command for the HSC. For example, the Display command is both an MVS/CSC and an HSC command. The first command in the following example shows the Display command for the MVS/CSC; it has no component identifier. The second command shows the Display command for the HSC; it specifies the HSC component identifier.

```
CSC0 DISPLAY .....
CSC0 HSC DISPLAY .....
```

Refer to the *VM/HSC Operator's Guide* for a complete list of HSC commands and descriptions of each command.

## CLS Commands

The MVS operator console interface supports the following CLS commands:

- Query
- Release
- Reserve
- Tell

Refer to the *CLS Reference Manual* for a complete list of CLS commands and descriptions of each command.

## Operator Response

Some HSC and CLS commands require a reply to the response returned by the HSC or CLS. The MVS/CSC coordinates the local response with the MVS system console so that replies should use the MVS reply identifier. The CLS and HSC systems may also send unsolicited messages to the MVS host system. Responses from unsolicited messages are routed through the CLS. ACSLS does not send unsolicited messages to the MVS/CSC.

# Programmatic Interface

---

---

## Overview

This chapter describes the programmatic interface (PGMI) provided with the MVS/CSC for the UNIX-based LCS. The PGMI allows S/370 or S/390 assembler programs to request certain services from the MVS/CSC, such as:

- MVS/CSC subsystem status (QCSC request)
- Drive configuration (QDRIVES request)
- Tape cartridge status (QVOLUME request)

The MVS/CSC provides the SCSXREQ macro to allow programs to request these services. A program uses the SCSXREQ macro to query the MVS/CSC to determine if a volume is resident in the ACS, or to retrieve drive-related configuration data. A program can also request that the MVS/CSC return the response(s) that were associated with an asynchronous request that was previously issued (READ request). See [“PGMI Supported Requests” on page 46](#) for more information about the requests supported by the PGMI.

---

## Command Syntax

See [“Syntax Flow Diagrams” on page xiii](#) for syntax flow diagramming conventions.

---

# PGMI Interface Processing

The SCSXREQ macro is used to request services from the MVS/CSC. The request parameter list and response areas are mapped by the SCSXREQM macro. An assembler program that issues the SCSXREQ macro must also include the SCSXREQM macro in order to generate the DSECTs used by the SCSXREQ macro.

The SCSXREQ macro calls the SCSXCAL module to transfer the request to the MVS/CSC. The SCSXCAL module locates the MVS/CSC program call (PC) number for cross address space communication. SCSXCAL's address is passed to the SCSXREQ macro.

---

**Note** – The SCSXCAL module must already be loaded. SCSXCAL is reentrant, so you need only to load it once.

---

A return code of zero in register 15 (R15) from the SCSXREQ macro indicates that the request was successfully initiated. A return code of 20 in R15 indicates that the MVS/CSC was inactive to initiate the request. (See [“Return Codes” on page 43](#) for a description of return codes). A program can issue the query MVS/CSC (QCSC) request to periodically poll the MVS/CSC subsystem status to determine when it becomes active.

When request processing completes, the MVS/CSC returns the address of the response area along with its length. Register 1 (R1) contains the address of the response area and register 0 (R0) contains the length of the response. The response area is in nonfetch protected MVS/CSC key storage. A program should not attempt to alter or free the virtual storage (FREEMAIN) for the response area. This response area is valid until the next SCSXREQ call, after which it no longer contains the response.

Both synchronous and asynchronous interfaces are provided. Synchronous requests cause the requesting program to wait until a request is complete before continuing processing. Asynchronous requests allow the requesting program to have multiple requests processed at once, and returns control to the requesting program immediately.

---

**Note** – Asynchronous requests are not supported for the QCSC and READ requests.

---

After issuing an asynchronous request, R0 contains a token and R1 contains the address of an Event Control Block (ECB). The ECB is posted after the request is complete. After the ECB is posted, the program must issue a SCSXREQ READ request specifying the returned token. This causes a single response to be made available to the requesting program. The ECB and token are no longer valid after the response has been read.

If the PGMI application terminates before all asynchronous requests have completed, any pending requests are canceled. To verify that all requests have completed, the PGMI application must wait for all ECBs to be posted.

The SCSXREQM macro maps the responses associated with a request. The response block always contains a reply header that is mapped by the SCX DSECT. The contents of the directory sections that follow the SCX header depend on the type of request that was made. Elements of a similar type are always contiguous (i.e., all drive elements are together).



The directory sections contain:

- The number of each type of element
- The offset from the start of reply header to the start of the elements of a specified type
- The length of each element type.

In order to ensure compatibility with future enhancements, reference the field names that are in the reply area, and not the calculated offsets.

In the event that the MVS/CSC becomes inactive during request processing, the ECB is posted and the response area indicates that the MVS/CSC is inactive.

A request can only be issued in task mode with no locks held. A task can have multiple requests active at once.

---

**Note** – See [“SCSXREQ Macro Format” on page 42](#) for the syntax of the SCSXREQ macro. See [Appendix A, “Programmatic Interface SCSXREQM Macro Mappings”](#) for the syntax and mappings of the SCSXREQM macro.

---

## Multiple MVS/CSCs and the PGMI

Multiple MVS/CSCs can run on a single MVS system. Each MVS/CSC is assigned a unique four-character subsystem name. The PGMI is able to route requests to a particular MVS/CSC subsystem if the subsystem name is included on a request. A program can issue the QCSC request to determine which MVS/CSC subsystems are active, and in turn select the appropriate MVS/CSC to process a request. If an MVS/CSC subsystem name is not included on a request, the first active MVS/CSC found in the MVS subsystem name table is used to process the request.

## Authorization Checking for the PGMI

The MVS/CSC provides user exit SCSUX05 to perform authorization checking on programs that request PGMI services. If the user exit is disabled (setting return code UX05NOPR) or ABENDs, the MVS/CSC allows all programs to issue the query and read requests. You can also code user exit SCSUX05 to interface with existing security products (i.e., RACF).

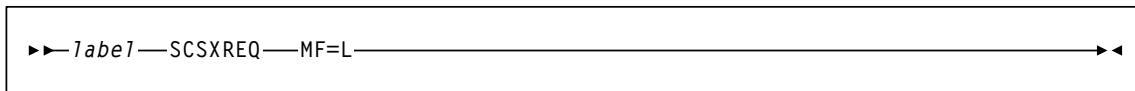
The SCSUX05 user exit is invoked for all requests received from the PGMI interface. This user exit should evaluate the request and set a return code to indicate whether the MVS/CSC should allow the request to be executed. Refer to the *ELS Legacy Interfaces Reference* for more information about the SCSUX05 user exit.

## SCSXREQ Macro Format

The SCSXREQ macro is used to request MVS/CSC services. There are two forms of the SCSXREQ macro: list and execute. You use the list form of the SCSXREQ macro to define a parameter list, and the execute form to process PGMI requests. The following sections describe both forms of the SCSXREQ macro.

### Specifying the List Form

Use the list form of the SCSXREQ macro to define a parameter list that is referenced when the SCSXREQ macro is executed for request processing. The following diagram shows the syntax for the list form.



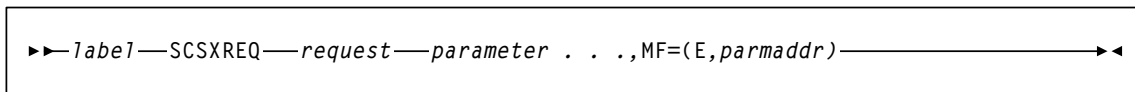
#### Parameters

##### MF=L

Specifies that a parameter list is generated. All other options are ignored.

### Specifying the Execute Form

Use the execute form of the SCSXREQ macro for request processing. The following diagram shows the syntax for the execute form.



When SCSXREQ is executed for request processing, it stores parameters in the parameter list defined by the SCSXREQ macro, and calls the SCSXCAL module.

#### Parameters

The *request* variable corresponds to one of the following requests:

##### QCSC

Determine MVS/CSC status

##### QDRIVES

Retrieve drive information

##### QVOLUME

Retrieve volume status

## READ

Read a response

The *parameter* variable corresponds to one or more of the following parameters, depending on the type of request being processed:

```

,ACSID= acsidaddr
,COUNT= countaddr
,CSCNAME= subsysaddr
,LIST= itemlistaddr
,MF=(E, parmaddr)
,OPTION={SYNCH | ASYNCH}
,TOKEN= tokenaddr
,VOLSER= voladdr or volumeaddr
,XCALADR= rtnaddr or subrtnaddr

```

See “PGMI Supported Requests” on page 46 for more information about PGMI requests and their associated parameters.

## SCSXREQ Return Specifications

The following sections describe the return codes and reason codes associated with PGMI request processing.

### Return Codes

After executing the SCSXREQ macro, R15 contains one of the following return codes:

Return Field Name	Decimal Value and Description
SCXROK	0 - Wait can be issued (OPTION=ASYNCH) or check response area for status (OPTION=SYNCH).
SCXRNCSC	20 - MVS/CSC not active; no SCX response is returned.

After the ECB associated with the request for asynchronous processing is posted, or upon return for synchronous requests, the SCXCMDRC return code field indicates the status of the request.

TABLE 4-1 on page 44 lists possible return codes.

TABLE 4-1 Return Code Specifications

Return Field Name	Decimal Value and Description
SCXROK	0 - Request completed successfully.
SCXRWARN	4 - The request completed successfully. However, an anomalous condition occurred (e.g., a QVOLUME request specified a volume not in the library).
SCXRBADP	8 - Request unsuccessful. Bad parameter list (e.g., incompatible options). SCXPEOFF provides an offset to the bad parameter.
SCXRIERR	12 - Request unsuccessful. Unrecoverable error occurred while processing the request.
SCXRFAIL	16 - Request unsuccessful. SCXSRC contains the request specific reason code or message. Note: PGMI reason codes are described in "Reason Codes". MVS/CSC messages and non-PGMI reason codes are described in the <i>MVS/CSC Messages and Codes Guide</i> .
SCXRNCSC	20 - Request unsuccessful. MVS/CSC not active.
SCXRNAUT	24 - Request unsuccessful. User not authorized. The request was failed by SCSUX05, or the MVS/CSC default authorization.
SCXNTCB	28 - Request unsuccessful. The caller was not in task mode.
SCXLOCKD	32 - Request unsuccessful. The caller held at least one lock.
SCXTPROT	36 - Request unsuccessful. The caller did not have fetch/store authority to the SCSXREQM parameter list.
SCXNORSP	40 - Request unsuccessful. No response element with the specified token existed.

## Reason Codes

If the request is unsuccessful (return code 16), the SCXSRC reason code field contains the reason code or message indicating the problem. The following table lists possible reason codes and messages.

**TABLE 4-2** Reason Code Specifications

Reason Code Field Name	Hex Value and Description
SCXVNF	X'700C' - One or more volumes where not found in the library.
SCXNOALS	X'8004' - MVS/CSC is not active.
SCXMCSC	X'8008' - Multiple MVS/CSCs active.
SCXQNOA	X'8014' - ASCOMM is not active.
SCXINVOP	X'8018' - Invalid QUAB option. Internal error; contact StorageTek Software Support.
SCXINVFC	X'800C' - Invalid function code. Internal error; contact StorageTek Software Support.
SCXNOLVT	X'8010' - No LVT. Internal error; contact StorageTek Software Support.
SCXNODRV	X'801C' - No drives found.
SCXTABND	X'8024' - ASCOMM server task abended.
SCXTINTR	X'8032' - Current task interrupted.
SCXSANF	X'8036' - Search argument not found.
SCXMSTT	X'803A' - Mismatched token types.
SCXTRNF	X'8040' - Token request area not found.
SCXSFUL	X'8050' - Reply area space full.
SCXDVMM	X'8051' - Drive/media mismatch.
SCXCMAx	X'8052' - Maximum number of active MVS/CSCs found while processing the QCSC request.
SCXSRVU	X'8053' - LCS became unavailable either before or during request processing.
SCXUNSP	X'8054' - Request type not supported.
SCXLMAX	X'8055' - Number of list items specified by requesting program exceeds maximum amount allowed.
SCXLCSF	X'8056' - LCS failed the request that was sent by the MVS/CSC. The SCXLCSRC field contains the server reason code.
SCXVINV	X'8057' - An invalid volume serial number was found during QVOLUME request processing.

# PGMI Supported Requests

The following sections describe the requests supported by the PGMI.

## QCSC

Use the QCSC request to determine if the MVS/CSC subsystem(s) are active. This allows programs to poll periodically to determine when the MVS/CSC becomes active. All MVS/CSC subsystems must run on the same system as the PGMI.

---

**Note** – The QCSC request is synchronous only.

---

## Syntax

▶▶ *label* — SCSXREQ QCSC — ,MF=(E,*parmaddr*) — ,CSCNAME=*subsysaddr* — ,XCALADR=*rtnaddr* —▶▶

## Parameters

### MF=(E, *parmaddr*)

Specifies that the SCSXREQ macro be executed for request processing using a parameter list.

Specify an RX-type address, or the number of the register (1 - 12) that contains the parameter list address.

This parameter is required.

### CSCNAME

Identifies the MVS/CSC subsystem for status checking, where *subsysaddr* specifies the address of a 4-byte field containing either the MVS/CSC subsystem name, or the literal \*ALL, which indicates that the status of all MVS/CSC subsystems is checked. *subsysaddr* can either be an RX-type address of the 4-byte field, or the number of the register (2 - 12) that contains the address of the 4-byte field.

This parameter is optional. If CSCNAME is not specified, the status of all MVS/CSC subsystems is checked.

### XCALADR

Specifies a location containing the address of the SCSXCAL module, where *rtnaddr* is the address. Specify either an RX-type address of the module, or the number of the register (2 - 12, or 15) that contains the address of the SCSXCAL module.

This parameter is required.

## Request Responses

If *subsysaddr* points to an MVS/CSC subsystem name, a return code in R15 indicates whether the MVS/CSC subsystem is active (0) or not (20). The response length (R0) and response address (R1) are zero.

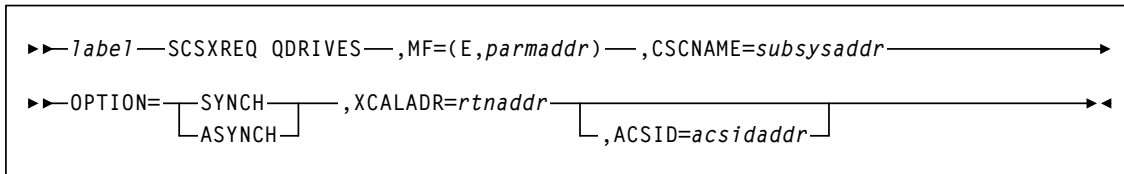
If *subsysaddr* points to the literal \*ALL, and only one MVS/CSC subsystem is active, both R15 and the SCXCMDRC field contain return code 0. If *subsysaddr* points to the literal \*ALL, and more than one MVS/CSC subsystem is active, R15 contains return code 0; the SCXCMDRC field contains return code 4, indicating that the request was successful but a warning was issued. The SCXSRC field in the response area contains a reason code (X'8008') indicating that multiple MVS/CSCs are active. R1 contains the address of the response; R0 contains the length of the response. See [“Return Codes” on page 43](#) for a description of the return codes. See [“Reason Codes” on page 45](#) for a description of the reason codes.

In addition to the reply header, the MVS/CSC Subsystem Name Information Element that is mapped by the SCXSNAM DSECT is returned when *subsysaddr* points to the literal \*ALL. If no MVS/CSC subsystem is active, R15 contains return code 20; R0 and R1 both contain zero.

## QDRIVES

Use the QDRIVES request to retrieve drive-related configuration data. The response includes the ACS and LSM address of the transports, as well as the control unit address of each transport. The response also includes the recording technique used by each transport.

### Syntax



### Parameters

#### MF=(E, parmaddr)

Specifies that the SCSXREQ macro be executed for request processing using a parameter list.

Specify an RX-type address, or the number of the register (1 - 12) that contains the parameter list address.

This parameter is required.

#### CSCNAME

Identifies the MVS/CSC subsystem for request processing, where *subsysaddr* specifies the address of a 4-byte field containing the MVS/CSC subsystem name. *subsysaddr* can either be an RX-type address of the 4-byte field, or the number of the register (2 - 12) that contains the address of the 4-byte field.

This parameter is optional. If it is not specified, the first active MVS/CSC in the MVS subsystem name table processes the request.

#### OPTION

Specifies whether the requesting program is to wait until the request is complete or continue processing while the request is active.

This parameter is required.

#### SYNCH

Specifies that the requesting program wait until the request is complete.

#### ASYNCH

Specifies that the requesting program continue processing while the request is active.

After control is returned to the requesting program, R1 contains the address of an ECB that is posted when the response completed. R0 contains the token used by SCSXREQ READ to read the response.



**XCALADR**

Specifies a location containing the address of the SCSXCAL module, where *rtnaddr* is the address. Specify either an RX-type address of the module, or the number of the register (2 - 12, or 15) that contains the address of the SCSXCAL module.

This parameter is required.

**ACSID**

Specifies the address of the ACS used to satisfy the request, where *acsidaddr* is the ACS address.

The format of *acsidaddr* is *AA*, where *AA* is the ACS number. *AA* is in hexadecimal format. For example, 01, designates ACS 01. Specify either an RX-type address of the data or the number of the register containing the address of the data.

This parameter is optional. If you do not specify an ACS address, drive information for all of the ACSs that are managed by the processing MVS/CSC subsystem is returned.

## Request Responses

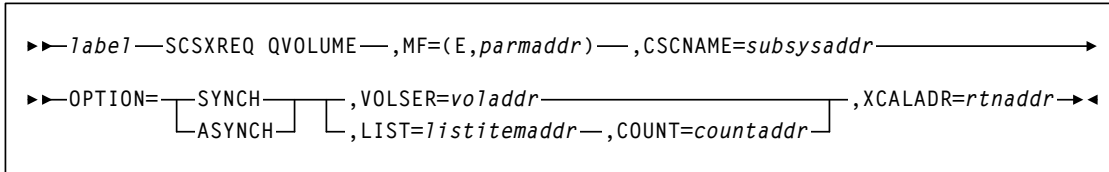
A return code in the SCXCMDRC field indicates whether or not the request was successful. If the request failed, a reason code is placed in the SCXSRC field. See [“Return Codes” on page 43](#) for a description of the return codes. See [“Reason Codes” on page 45](#) for a description of the reason codes.

In addition to the reply header, the Drive Information Element that is mapped by the SCXSDRV DSECT is returned.

# QVOLUME

Use the QVOLUME request to obtain the library status of a cartridge.

## Syntax



## Parameters

### MF=(E, *parmaddr*)

Specifies that the SCSXREQ macro be executed for request processing using a parameter list.

Specify an RX-type address, or the number of the register (1 - 12) that contains the parameter list address.

This parameter is required.

### CSCNAME

Identifies the MVS/CSC subsystem for request processing, where *subsysaddr* specifies the address of a 4-byte field containing the MVS/CSC subsystem name. *subsysaddr* can either be an RX-type address of the 4-byte field, or the number of the register (2 - 12) that contains the address of the 4-byte field.

This parameter is optional. If it is not specified, the first active MVS/CSC in the MVS subsystem name table processes the request.

### OPTION

Specifies whether the requesting program is to wait until the request is complete or continue processing while the request is active.

This parameter is required.

### SYNCH

Specifies that the requesting program wait until the request is complete.

### ASYNCH

Specifies that the requesting program continue processing while the request is active.

After control is returned to the requesting program, R1 contains the address of an ECB posted when the response completed. R0 contains the token used by SCSXREQ READ to read the response.

### VOLSER

Specifies the address of a six-character volume serial number, where *voladdr* is the VOLSER address.

Specify either an RX-type address, or the number of the register (2 -12) that contains the volume serial number address. Valid characters for the volume serial number are A-Z, 0-9, #, \$, @, and trailing blanks. This parameter is required.

---

**Note –**

- The at sign (@) character is only valid for the VM-based LCS.
  - The yen (¥) and pound sterling (£) characters correspond to the same EBCDIC values as the dollar sign (\$) and pound sign (#) characters (X '5B' and X '7B'), and can also be used in volume serial numbers if the LMU microcode level is at Release 3.5 or later.
- 

**LIST**

Specifies the address of a list of elements, where *listitemaddr* is either an RX-type address of the data, or the number of a register containing the address of the data. LIST designates a list of 6-byte volume serial numbers.

This parameter is required if VOLSER is not specified.

**COUNT**

Specifies the address of a two-byte count value, where *countaddr* is either an RX-type address of the data, or the number of the register containing the address of the data.

This parameter is required when LIST is specified.

**XCALADR**

Specifies a location containing the address of the SCSXCAL module, where *rtnaddr* is the address. Specify either an RX-type address of the module, or the number of the register (2 - 12, or 15) that contains the address of the SCSXCAL module.

This parameter is required.

## Request Responses

A return code in the SCXCMDRC field indicates whether or not the request was successful. If the request failed, a reason code is placed in the SCXSRC field. If the volume is not in the library, SCXRWARN is placed in the SCXCMDRC field and a reason code is placed in the SCXSRC field. See [“Return Codes” on page 43](#) for a description of the return codes. See [“Reason Codes” on page 45](#) for a description of the reason codes.

In addition to the reply header, the Volume Information Section that is mapped by the SCXSVOL DSECT is returned.

## READ

Use the READ request to direct the MVS/CSC to return the response(s) associated with a request that has completed as indicated by a posted ECB. The READ request is used to read a response associated with a particular token. The requesting program specifies the token that was returned on the original request.

The READ request is synchronous only.

### Syntax

```
▶▶ label — SCSXREQ READ — , MF=(E, parmaddr) — , CSCNAME=subsysaddr —————▶
▶▶ , TOKEN=tokenaddr — , XCALADR=rtnaddr —————▶▶
```

### Parameters

#### MF=(E, *parmaddr*)

Specifies that the SCSXREQ macro be executed for request processing using a parameter list.

Specify an RX-type address, or the number of the register (1 - 12) that contains the parameter list address.

This parameter is required.

#### CSCNAME

Identifies the MVS/CSC subsystem for request processing, where *subsysaddr* specifies the address of a 4-byte field containing the MVS/CSC subsystem name. *subsysaddr* can either be an RX-type address of the 4-byte field, or the number of the register (2 - 12) that contains the address of the 4-byte field.

This parameter is optional. If it is not specified, the first active MVS/CSC in the MVS subsystem name table processes the request.

#### TOKEN

Specifies the address of the token originally returned in R0 when the request was initiated, where *tokenaddr* is the token address.

Specify either an RX-type address of the token, or the number of the register (2 - 12) that contains the address of the token.

This parameter is required.

#### XCALADR

Specifies a location containing the address of the SCSXCAL module, where *rtnaddr* is the address. Specify either an RX-type address of the module, or the number of the register (2 - 12, or 15) that contains the address of the SCSXCAL module.

This parameter is required.

## Request Responses

A return code in the SCXCMDRC field indicates whether or not the request was successful. If the request failed, a reason code is placed in the SCXSRC field. See [“Return Codes” on page 43](#) for a description of the return codes. See [“Reason Codes” on page 45](#) on page 29 for a description of the reason codes.

In addition to the reply header, the appropriate Information Elements are returned based on the original request.

# Sample PGMI Requests

The following sections show sample PGMI requests for querying a volume in a single and multiple MVS/CSC environment.

## Sample QVOLUME Request (Single MVS/CSC Environment)

The following figure shows a sample QVOLUME request issued in a single MVS/CSC environment. This sample resides in member PGMISAM1 of the SAMPLIB library.

```

PSAM1      TITLE 'PGMISAM1 - PGMI Sample Program 1'
**PROLOGUE*****
*
* PGMISAM1 - PGMI Sample Program 1
*
* FUNCTION:  This sample routine demonstrates the use of the
*            QVOLUME request.
*            WTOs are used in this module as a diagnostic aid to
*            trace the execution flow.
*
* ATTRIBUTES: REENTRANT, RMODE(ANY), AMODE(31).
*
* SERIALIZATION REQUIREMENTS: None
*
* REGISTER CONVENTIONS:
*
*     R13 - PGMISAM1 workarea base register.
*     R12 - PGMISAM1 module base register.
*     R10 - SCX Response Area base register.
*     R8  - Volume Information Element base register.
*
* ENTRY POINTS (EXECUTABLE):
*
*     PGMISAM1
*
*     PARAMETER DESCRIPTION:
*     None
*

```

FIGURE 4-1 QVOLUME Request (Single MVS/CSC) (1 of 5)

```

* EXIT (NORMAL): *
* *
* RETURN CODE: N/A *
* *
* REGISTER CONDITION: All registers are the same as upon entry *
* except R15 = zero. *
* *
* DATA PASSED: *
* *
* SPECIAL CONDITIONS: None *
* *
* EXIT (ABNORMAL): None *
* *
* EXTERNAL REFERENCES: *
* *
* WTO - MVS Write to Operator. *
* SCSXCAL - MVS/CSC PGMI Interface Module *
* *
* MESSAGES GENERATED: None *
* *
* DATA AREAS: *
* *
* SCSXREQM - PGMI Request/Response Mapping Macro *
* *
* SPECIAL CONSIDERATIONS: None *
* *
* CHANGE HISTORY: *
* 01/29/97-CRTD-MJD Created initial version. *
* *
**ENDPROLOGUE*****
    EJECT
    COPY SCSGLBL
PGMISAM1 CSECT
PGMISAM1 AMODE 31
PGMISAM1 RMODE ANY
**
    EJECT
**
** Store callers registers
** Establish CSECT base address
**
    SPACE
    STM R14,R12,12(R13) Store callers regs
    LR R12,R15 Establish primary
    USING PGMISAM1,R12 address.
    SPACE
**
** Getmain a workarea
**
    GETMAIN RU,LV=PSAM1WKL,SP=0,LOC=ANY

```

FIGURE 4-2 QVOLUME Request (Single MVS/CSC) (2 of 5)

```

        ST    R13,4(,R1)      Store backward pointer
        ST    R1,8(,R13)     Store forward pointer
        LR    R13,R1         Point to new savearea
        USING PSAM1WK,R13    Map it
**
**
**              MAINLINE
**
*
        WTO   'PGMISAM1 - Entered'
*
**
**      Load the MVS/CSC Interface Module, SCSXCAL
**
        LOAD  EP=SCSXCAL      Load PGMI interface routine
        ST    R0,WKXCAL@     Save the EPA
*
**
**      In this sample module, we hard-code a volume serial number
**      to be used on the subsequent QVOLUME request.
**
*
        MVC   WKVOLSR,=CL6'MYVOL1' Set volser we're looking for.
*
**
**      Issue a QVOLUME request to retrieve information about the
**      specified volume. Because CSCNAME is not specified, the
**      request will be directed to the first active MVS/CSC subsystem
**      in the MVS Subsystem Vector Table.
**
*
QVOLUME EQU   *
        WTO   'PGMISAM1 - Issuing QVOLUME request.'
                SCSXREQ QVOLUME,
                VOLSER=WKVOLSR,
                XCALADR=WKXCAL@,
                OPTION=SYNCH,
                MF=(E,WKXPLST)
*
        LTR   R15,R15        Request successfully initiated?
        BNZ   NOCSC          No, CSC not active.
        LR    R10,R1         Get addr of SCX into R10.
        USING SCX,R10        Map the SCX.
        LR    R8,R10         Copy into R8.
        A     R8,SCXXVOLO    Get offset to vol info entry.
        USING SCXSVOL,R8     Map the vol info entry.
        TM    SCXVSTA,SCXVILB Volume in library?
        BO    LIBVOL         Yes, go do stuff for vol in library.
*

```

FIGURE 4-3 QVOLUME Request (Single MVS/CSC) (3 of 5)



```

NOLIBVOL EQU *
          WTO 'PGMISAM1 - Volume not in library.'
*
*      Insert code here to take the appropriate action for a
*      nonlibrary volume.
*
          B      RETURN          Go return to caller.
*
LIBVOL   EQU *
          WTO 'PGMISAM1 - Volume in library.'
*
*      Insert code here to take the appropriate action for a
*      library volume.
*
          B      RETURN          Go return to caller.
*
NOCS    EQU *
          WTO 'PGMISAM1 - No CSC subsystem active.'
*
*      Insert code here to take the appropriate action for the
*      condition where no CSC subsystems are active.
*
          B      RETURN          Go return to caller.
*
**
**      RETURN: **      Return to Caller.
**
RETURN  EQU *
          WTO 'PGMISAM1 - Exiting'
          L      R13,4(,R13)      Get addr of callers savearea
          LM     R14,R12,12(R13)   Load callers regs
          XR     R15,R15          Give zero return code.
          BR     R14              And return.
*
*
*
*
          TITLE 'PGMISAM1 Constants'
**
** Constants.
**
*
**
** Literals.
**
          LTORG
          EJECT ,
          TITLE 'PGMISAM1 Module Workarea'

```

FIGURE 4-4 QVOLUME Request (Single MVS/CSC) (4 of 5)

## Sample PGMI Requests

```
PSAM1WK DSECT
        DS      18F                SCSCALL Register Save Area.
WKXPLST SCSXREQ MF=L
WKVOLSR DS      CL6                Volser
WKXCAL@ DS      F                  Pointer to SCSXCAL EPA
        DS      0D
PSAM1WKL EQU *-PSAM1WK
*
*
        SPACE 2
        SCSXREQM                    PGMI Request/Response Mapping Macro
***     SYMBOLIC REGISTER EQUATES
        SPACE
R0      EQU    0
R1      EQU    1
R2      EQU    2
R3      EQU    3
R4      EQU    4
R5      EQU    5
R6      EQU    6
R7      EQU    7
R8      EQU    8
R9      EQU    9
R10     EQU    10
R11     EQU    11
R12     EQU    12
R13     EQU    13
R14     EQU    14
R15     EQU    15
*
        END
```

**FIGURE 4-5** QVOLUME Request (Single MVS/CSC) (5 of 5)

## Sample QVOLUME Request (Multiple MVS/CSC Environment)

The following figure shows a sample QVOLUME request issued in a multiple MVS/CSC environment. This sample resides in member PGMISAM2 of the SAMPLIB library.

```

PSAM2      TITLE 'PGMISAM2 - PGMI Sample Program 2'
**PROLOGUE*****
*
* PGMISAM2 - PGMI Sample Program 2
*
* FUNCTION:  This sample routine illustrates several features of
*            the MVS/CSC PGMI. This module assumes that it is
*            possible to have more than one MVS/CSC subsystem active
*            on a single MVS image. The QCSC request is used to
*            determine all active CSC subsystems. We then want to
*            retrieve volume information for a particular volser.
*            We loop through all active subsystems issuing QVOLUME
*            requests until a CSC tells us that it has the volume in
*            its library, or, until we exhaust the list of active
*            subsystems.
*            WTOs are used in this module as a diagnostic aid to
*            trace the execution flow.
*
* ATTRIBUTES: REENTRANT, RMODE(ANY), AMODE(31).
*
*
* SERIALIZATION REQUIREMENTS: None
*
* REGISTER CONVENTIONS:
*
*     R13 - PGMISAM2 workarea base register.
*     R12 - PGMISAM2 module base register.
*     R10 - SCX Response Area base register.
*     R9  - CSC Name Information Element base register.
*     R8  - Volume Information Element base register.

```

**FIGURE 4-6** QVOLUME Request (Multiple MVS/CSCs) (1 of 6)

```

*
* ENTRY POINTS (EXECUTABLE):
*
*     PGMISAM2
*
*             PARAMETER DESCRIPTION:
*             None
*
* EXIT (NORMAL):
*
*     RETURN CODE: N/A
*
*     REGISTER CONDITION: All registers are the same as upon entry
*     except R15 = zero.
*
*     DATA PASSED:
*
*     SPECIAL CONDITIONS: None
*
* EXIT (ABNORMAL): None
*
* EXTERNAL REFERENCES:
*
*     WTO - MVS Write to Operator.
*     SCSXCAL - MVS/CSC PGMI Interface Module
*
* MESSAGES GENERATED: None
*
* DATA AREAS:
*
*     SCSXREQM - PGMI Request/Response Mapping Macro
*
* SPECIAL CONSIDERATIONS: None
*
* CHANGE HISTORY:
*     01/29/97-CRTD-MJD   Created initial version.
*
**ENDPROLOGUE*****
*
*     EJECT
*     COPY  SCGLBL
PGMISAM2 CSECT
PGMISAM2 AMODE 31
PGMISAM2 RMODE ANY
**
*
*     EJECT
**
** Store callers registers
** Establish CSECT base address
**

```

FIGURE 4-7 QVOLUME Request (Multiple MVS/CSCs) (2 of 6)

```

SPACE
STM  R14,R12,12(R13)   Store callers regs
LR   R12,R15           Establish primary
USING PGMISAM2,R12     address.
SPACE
**
** Getmain a workarea
**
GETMAIN RU,LV=PSAM2WKL,SP=0,LOC=ANY
ST   R13,4(,R1)        Store backward pointer
ST   R1,8(,R13)        Store forward pointer
LR   R13,R1            Point to new savearea
USING PSAM2WK,R13      Map it
**
**
**           MAINLINE
**
*
WTO   'PGMISAM2 - Entered'
*
**
** Load the MVS/CSC Interface Module, SCSXCAL
**
LOAD  EP=SCSXCAL        Load PGMI interface routine
ST    R0,WKXCAL@        Save the EPA
*
**
** Find the names of all active MVS/CSC subsystems. Preserve
** these names in a table in the workarea.
**
*
SCSXREQ QCSC,           Locate all active CSC subsystems      +
      XCALADR=WKXCAL@,                                     +
      MF=(E,WKXPLST)
*
LTR   R15,R15           Is any CSC active?
BZ    CSCOK             Yes, proceed.
B     NOCSC             Anything else -> can't talk to CSC.
CSCOK EQU *
LR    R10,R1            Get address of SCX response area.
USING SCX,R10          Map the SCX.
L     R3,SCXXNAMN       Get number of active CSCs.
ST    R3,WKCSCNUM       Save for future use.
LA    R4,WKKNAMES       Get addr of CSC name save table.
LR    R9,R10            Get address of SCX into R9.
A     R9,SCXXNAMO       Add offset to first CSC name element.
USING SCXSNAM,R9       Map the CSC name element.
*

```

FIGURE 4-8 QVOLUME Request (Multiple MVS/CSCs) (3 of 6)

```

NAMELOOP EQU      *
              WTO   'PGMISAM2 - Active CSC subsystem found.'
              MVC   0(4,R4),SCXNAME      Save the CSC name.
              LA    R4,4(,R4)           Bump pointer to next table entry.
              A     R9,SCXXNAML         Move to next CSC name element.
              BCT   R3,NAMELOOP         Go process next CSC name.

*
**
**      In this sample module, we hard-code a volume serial number
**      to be used on the subsequent QVOLUME request.
**
*
              MVC   WKVOLSR,=CL6'MYVOL1' Set volser we're looking for.
*
**
**      Loop thru the active MVS/CSC subsystems that we found by
**      doing the QCSC request and issue a QVOLUME request for our
**      volser. As soon as it is found, exit the loop and go perform
**      whatever logic is necessary to process a library volume. If
**      we loop through all the active MVS/CSC subsystems and don't
**      find the volume, go perform whatever logic is necessary to
**      to process a nonlibrary volume.
**
*
              LA    R4,WKNAMES          Get addr of CSC name save table.
              L     R3,WKCSCNUM         Get number of active CSCs.
QVOLOOP EQU      *
              WTO   'PGMISAM2 - Checking active CSC for volume.'
              SCSXREQ QVOLUME,
              VOLSER=WKVOLSR,
              CSCNAME=(R4),
              XCALADR=WKXCAL@,
              OPTION=SYNCH,
              MF=(E,WKXPLST)
*
              LTR   R15,R15             Request successfully initiated?
              BNZ   NEXTCSC            No, go try next CSC subsystem.
              LR    R10,R1             Get addr of SCX into R10.
              LR    R8,R10             Copy into R8.
              A     R8,SCXXVOLO        Get offset to vol info entry.
              USING SCXSVOL,R8        Map the vol info entry.
              TM    SCXVSTA,SCXVILB    Volume in library?
              BO    LIBVOL             Yes, go do stuff for vol in library.
NEXTCSC EQU      *
              LA    R4,4(R4)          Position to next CSC in table.
              BCT   R3,QVOLOOP        Go see if next CSC has the volume.
*

```

FIGURE 4-9 QVOLUME Request (Multiple MVS/CSCs) (4 of 6)

```

NOLIBVOL EQU *
          WTO 'PGMISAM2 - Volume not in library.'
*
*      Insert code here to take the appropriate action for a
*      nonlibrary volume.
*
          B      RETURN          Go return to caller.
*
LIBVOL   EQU *
          WTO 'PGMISAM2 - Volume in library.'
*
*      Insert code here to take the appropriate action for a
*      library volume.
*
          B      RETURN          Go return to caller.
*
NOCS    EQU *
          WTO 'PGMISAM2 - No CSC subsystem active.'
*
*      Insert code here to take the appropriate action for the
*      condition where no CSC subsystems are active.
*
          B      RETURN          Go return to caller.
*
**
**      RETURN:
**      Return to Caller.
**
RETURN  EQU *
          WTO 'PGMISAM2 - Exiting'
          L      R13,4(,R13)      Get addr of callers savearea
          LM     R14,R12,12(R13)  Load callers regs
          XR     R15,R15          Give zero return code.
          BR     R14              And return.
*
*
*
*
          TITLE 'PGMISAM2 Constants'
**
** Constants.
**
*
**
** Literals.
**
          LTORG
          EJECT ,
          TITLE 'PGMISAM2 Module Workarea'

```

FIGURE 4-10 QVOLUME Request (Multiple MVS/CSCs) (5 of 6)

```

PSAM2WK  DSECT
          DS      18F                SCSCALL Register Save Area.
WKXPLST  SCSXREQ MF=L
WKNNAMES DS      128CL4            CSC Subsystem Name Table area.
WKVOLSR  DS      CL6              Volser
WKXCAL@  DS      F                Pointer to SCSXCAL EPA
WKCSCNUM DS      F                Number of active CSC subsystems
          DS      0D
PSAM2WKL EQU *-PSAM2WK
*
*
          SPACE 2
SCSXREQM                PGMI Request/Response Mapping Macro
***  SYMBOLIC REGISTER EQUATES
          SPACE
R0      EQU  0
R1      EQU  1
R2      EQU  2
R3      EQU  3
R4      EQU  4
R5      EQU  5
R6      EQU  6
R7      EQU  7
R8      EQU  8
R9      EQU  9
R10     EQU  10
R11     EQU  11
R12     EQU  12
R13     EQU  13
R14     EQU  14
R15     EQU  15
*
          END

```

FIGURE 4-11 QVOLUME Request (Multiple MVS/CSCs) (6 of 6)



# MVS/CSC Utility Functions

---

---

## Overview

This chapter describes the following utilities provided with MVS/CSC:

- SCUADMIN utility program
- Event Log Report utility function
- Configuration Verification utility function
- Scratch Update utility function
- Scratch Conversion utility function

It also describes syntax conventions for utility control statements.

Utilities are provided to assist in managing library resources and reporting information. Some utility functions may require that the Library Control System (LCS) and MVS/CSC are active.

For convenience, most utility functions are packaged to appear as a single user-invoked program that is invoked as a normal batch job using the SCUADMIN utility program and user-specified control statements. The exception is the Scratch Conversion utility, which is invoked by the SCUCONDB program.

---

# Control Statement Syntax Conventions

A control statement consists of a command (indicating the utility function) followed by zero or more parameters in 80-character card-image records. Rules for specifying control statements are as follows:

- Control statement information must be contained between columns 2 and 72.
- Parameters must be separated by one or more blanks or commas.
- A value is associated with a parameter by enclosing the value in parentheses and concatenating it immediately after the parameter.
  - Case (upper or lower) is ignored in actual control statements.
  - Lists identify several specific elements (each one separated by a comma), an inclusive range of elements, or a single element. Lists may be a combination of both ranges and specific identifiers if commas are placed between the elements.
  - Inclusive ranges specify a beginning element identifier and an ending identifier separated by a dash.
- Control statement card images may be interspersed with comment card images designated by an asterisk in column 1.
- A control statement is terminated by a noncontinued card image.
- The 80-character card-image records use conventional continuation rules.
  - A space and a dash (-) anywhere before column 73 specify that the continued control information is on the next card image. The continuation starts after column 1 in the first nonblank column.
  - A plus sign (+) anywhere before column 73 specifies that the continued control information is to be concatenated directly after the character preceding the +. This allows continuation of the parameter or parameter value.
- A vol-list can consist of a volume serial number (volser), a list of volume serial numbers, ranges of volume serial numbers, or combinations of lists and ranges separated by commas. The entire list must be enclosed in parentheses. Rules for specifying a range of volume serial numbers (volsers) follows:
  - Both volsers specified must contain the same number of characters. For example, a range from volser 1 through volser 1800 must be specified as 0001-1800.
  - The specified beginning and ending elements must contain an incremental decimal numeric portion.
  - If the volser contains more than one incremental range, the portion that is lowest in order of magnitude is the portion that is incremented. Examples are shown as follows:

A00B00	The largest range is A00B00-A00B99
A0B0CC	The largest range is A0B0CC-A0B9CC
000XXX	The largest range is 000XXX-999XXX

- Any single utility control statement cannot contain more than 32767 characters. In the sample control statements described in the remainder of this chapter, the following rules apply:
  - Acceptable abbreviations of parameters are identified by capitalization of the required string.
  - Optional parameters are enclosed in brackets [ ].
  - Choices are indicated by separating items with a vertical bar |.
  - Defaults are indicated by underscoring \_.

---

**Note** – Unexpected results may occur if the SCSIN DD statement locates a data set that contains anything other than 80-byte logical records.

---

---

## Command Syntax

See [“Syntax Flow Diagrams” on page xiii](#) for syntax flow diagramming conventions.

---

## SCUADMIN Utility Program

The SCUADMIN utility program performs initialization processing for each utility function, except for SCUCONDB. It operates in a user address space. The SCUADMIN program accepts multiple control statements, which are handled as serial requests for multiple utility functions. If all requirements needed to execute the requested utility function are met, the SCUADMIN utility program passes control to the utility that handles the actual processing.

Utility functions specified in additional control statements are not started until the previous utility completes and control is returned to the SCUADMIN program.

Figure 8 shows a sample JCL stream that invokes three utilities: the Scratch Update utility, the Configuration Verification utility, and the Event Log Report utility.

---

**Note** – The DD statements required by the particular utility function being invoked must be specified.

---

```
//JOB1      JOB  ACCT,PGMR
//STEP1     EXEC PGM=SCUADMIN, PARM=' MIXED, DATE=4YR'
//STEPLIB   DD  DISP=SHR,DSN=your.SCSLINK
//          DD  DISP=SHR,DSN=your.TCPLINK /* optional for CA TCP */
//          DD  DISP=SHR,DSN=your.SACLINK
//          DD  DISP=SHR,DSN=your.CSLLINK
//SCSLOG    DD  DSN=EVENTLOG,DISP=SHR
//SCSPARM   DD  DSN=your.parmlib(csc-member)
//SCSPRINT  DD  SYSOUT=*
//SCSIN     DD  *
           LOGR BEGIN(19990130) HEART(NO)
           CONFIGV
           SCRATCH VOLSER(A1B1C1,A1B1C2,A1B1C3,A1B1C4)
/*
//
```

**FIGURE 5-1** Sample JCL for Invoking Multiple Utility Functions

---

**Note** – If you are running MVS/CSC and MVS/HSC on the same MVS system, the STEPLIB DD statement that identifies the MVS/CSC load libraries is required (see [TABLE 5-1](#)). If you omit the STEPLIB DD statement, unpredictable MVS/CSC operations may occur.

---

## SCUADMIN PARM Parameter

The PARM parameter specifies the case for printing output and the format for the date that appears in the body of printed reports. The following section describes the values you can specify on the PARM parameter.

### MIXED

Specifies that the SCSPRINT output be printed in mixed case. MIXED is the only value that you can specify on the PARM parameter for printed output.

If you do not specify MIXED on the PARM parameter, the SCSPRINT output is printed in uppercase.

### DATE

Specifies the format for the date that appears in the body of printed reports.

---

**Note** – This parameter applies only to dates that are printed in the body of reports. Dates printed in report headings always appear in four-year format.

---

### 2YR

Specifies that the date appears in two-year format (mm/dd/yy).

### 4YR

Specifies that the date appears in four-year format (yyyymmdd).

4YR is the default value.

## SCUADMIN Return Codes

The SCUADMIN utility program sets a return code for the execution of the job step as defined in Table 3. The return code for each utility function is reported in the SCS0155I message. For return codes other than 0, additional messages are reported providing more information and description of the error condition. Refer to the *MVS/CSC Messages and Codes Guide* for message descriptions.

Examples of output from utility functions, including messages and return codes, are provided throughout this chapter.

**TABLE 5-1** SCUADMIN Return Codes

<b>Return Code</b>	<b>Description</b>
0	All utility functions requested by control statements completed normally.
4	At least one utility function encountered an abnormal condition, but the abnormal condition did not terminate that utility function or any subsequent utility functions.
8	At least one utility function encountered an error condition that prevented that utility function from continuing. Any subsequent utility functions specified in control statements were attempted.
12	A fatal error occurred and no further processing will occur. The utility function cannot continue.

---

## Event Log Report Utility (VM-based LCS only)

The primary purpose of event logging is to record all message traffic between the MVS/CSC and the associated server systems. Some of the items written to the event-log data set include:

- Starts, stops, and cancellations of the MVS/CSC system
- Messages and responses sent to and received from the server
- MVS/CSC trace recordings
- Missed heartbeat intervals (for VM-based LCS)
- Miscellaneous communications events (for UNIX-based LCS)

For VM-based configurations, you can print the event log using the Event Log Report Utility. You can run the Event Log Report utility regardless of whether or not the MVS/CSC is running.

---

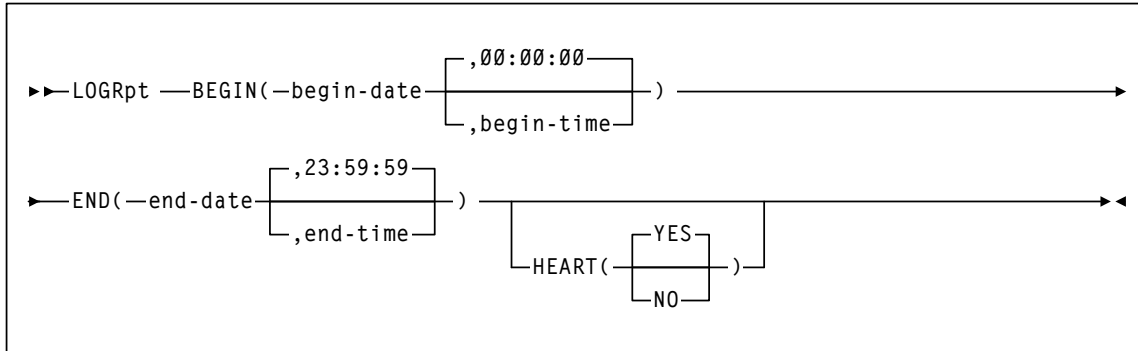
**Note –**

- The Event Log Report utility is not used for UNIX-based LCS configurations; log records are formatted when written to the event-log data set. Refer to the *MVS/CSC Operator's Guide* for information about using the Event Log Facility to log events for UNIX-based configurations.
  - If the Event Log Report utility is run against an active event-log data set, unpredictable results may occur. Therefore, the utility should be run against an inactive event-log data set. This can be done either with the MVS/CSC not running or by issuing the LOG NO command to close the current event-log data set.
-

## LOGRPT Control Statement

Use the LOGRPT control statement to invoke the Event Log Report utility.

### Syntax



### Parameter Descriptions

This section describes the values you can specify with the Event Log Report utility LOGRPT control statement.

#### BEGIN

Specifies the beginning of the period to start extracting data for the activities.

*begin-date* is the starting date of the reporting, expressed in either two-year or four-year format (mm/dd/yy or yyyyymmdd).

*begin-time* is the beginning time of day (twenty-four-hour value), expressed in hh:mm:ss format. The default value is 00:00:00.

#### END

Specifies the end of the period for the reporting.

*end-date* is the ending date of the reporting, expressed in either two-year or four-year format (mm/dd/yy or yyyyymmdd).

---

**Note** – The END date format must match the BEGIN date format.

---

*end-time* is the ending time of day (twenty-four-hour value), expressed in hh:mm:ss format. The default value is 23:59:59.

---

**Note** – The END parameter is valid only if the BEGIN parameter is specified.

---

#### HEART

Specifies whether CLS heartbeat messages are included in the Event Log Report.

YES specifies that heartbeat messages are included in the Event Log Report. This is the default value.

NO specifies that heartbeat messages are not included in the Event Log Report.



## Example of LOGRPT Control Statement

The following example shows a control statement used to invoke the Event Log Report utility to begin at 10:00 A.M. and end at 10:00 P.M. on January 7, 2010 and to include CLS heartbeat message. Notice that the date appears in four-year format (yyyymmdd).

```
LOGR BEGIN(20100107,10:00:00) END(20100107,22:00:00) HEART(YES)
```

## Sample JCL

Figure 9 shows sample JCL used to invoke the Event Log Report utility.

```
//CSC0LOGR JOB ACCT,PGMR
//STEP0 EXEC PGM=SCUADMIN, PARM='MIXED,DATE=4YR'
//STEPLIB DD DISP=SHR,DSN=your.SCSLINK
// DD DISP=SHR,DSN=your.TCPLINK /* Optional TCP load modules */
// DD DISP=SHR,DSN=your.SACLINK
// DD DISP=SHR,DSN=your.CSLLINK
//SCSLOG DD DISP=SHR,DSN=your.log.file
//SCSPRINT DD SYSOUT=*
//SCSIN DD *
LOGR [BEGIN(begin-date[,begin-time]] -
      [END(end-date[,end-time]] -
      [HEART(YES|NO)]
/*
//
```

**FIGURE 5-2** Sample JCL for Event Log Report Utility

The JCL to invoke the Event Log Report utility is contained in sample member JCLLOGR in the SAMPLIB library.

---

**Note –** If you are running MVS/CSC and MVS/HSC on the same MVS system, the STEPLIB DD statement that identifies the MVS/CSC load libraries is required (see [FIGURE 5-2](#)). If you omit the STEPLIB DD statement, unpredictable MVS/CSC operations may occur.

---

## Sample Output

Output resulting from the execution of the Event Log Report utility includes:

- A listing of input control statements with appropriate messages when syntax errors occur. Sample output is shown in [FIGURE 5-3](#).
- A formatted printout of all the events recorded in the event-log data set for the specified time period. Sample output for the VM-based LCS is shown in [FIGURE 5-4](#).

```

SCUADMIN (7.0.0) StorageTek MVS Client/Server System Utility          PAGE 001
TIME 10:51:02          Control Card Image Listing                  DATE 2010-06-07

LOGR  BEGIN(20100107,15:00:00) -
      END(20100107,23:30:00)
    
```

**FIGURE 5-3** Event Log Report Utility Sample Output-Input Control Statement

```

SCUADMIN (7.0.0)          StorageTek MVS Client/Server System Utility          PAGE 0002
TIME 16:34:10          MVS/CSC Event Log Report                  DATE 2010-06-07

|----- Event or -----|          1  1  2  2  3 (Detail)  4  5  5  6  6  7  7
Type Msg From/To Req ID Date -GMT- Time 0----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5
--5----0----5

Event Started MVS/CSC      20100107 15:00:00 Starting MVS/CSC Subsystem CSC0
Logon      CLS->CSC          LOGON ACCEPTED
Avail Req  CLS->CSC 000000 20100607 22:01:07 03.0301SMYTHE1
Avail Rsp  CLS<-CSC 000000 20100607 22:02:01 10001
Hrtbt Req  CLS->CSC 000000 20100607 22:26:00
Hrtbt Rsp  CLS<-CSC 000000 20100607 22:26:29
Comnd Req  CLS<-CSC 000013 20100607 22:26:23 01123HSC D V 000200
Comnd Rsp  CLS<-CSC 000045 20100607 22:26:38 10001
Comnd Req  CLS<-CSC 000013 20100607 22:26:15 02021.SLS0602I SCRATCH: NO
Comnd Rsp  CLS<-CSC 000045 20100607 22:26:43 10001
Comnd Req  CLS->CSC 000013 20100607 22:26:22 02022.SLS0602I SELECTED: NO
Comnd Rsp  CLS<-CSC 000045 20100607 22:26:47 10001
Comnd Req  CLS->CSC 000013 20100607 22:26:35 02031.SLS0602I LAST KNOWN LOCATIONS:
Comnd Rsp  CLS<-CSC 000045 20100607 22:26:51 10001
Comnd Req  CLS->CSC 000013 20100607 22:26:54 02032.SLS0602I HOME: CELL 001:18:02:15
Comnd Rsp  CLS<-CSC 000045 20100607 22:26:55 10001
Comnd Req  CLS->CSC 000013 20100607 22:26:57 02035.SLS0602I SOURCE: CELL 001:18:02:15
Comnd Rsp  CLS<-CSC 000045 20100607 22:26:58 10001
Comnd Req  CLS->CSC 000013 20100607 22:27:01 02033.SLS0602I DESTINATION: DRIVE 0A23
Comnd Rsp  CLS<-CSC 000045 20100607 22:27:02 10001
Comnd Req  CLS->CSC 000000 20100607 23:16:01 02073.CLS3700 CLS system Shutting down due to
          STOP ALL command issued by SMYTHE1
SCS0155I Condition Code for Utility Function is 0
    
```

**FIGURE 5-4** Event Log Report Utility Sample Output with VM-Based LCS-Formatted Event Log

# Configuration Verification Utility

The Configuration Verification utility allows you to verify and report on the following resources:

- User-selected startup parameters
- Device esoteric definitions
- LCS configuration compatibility

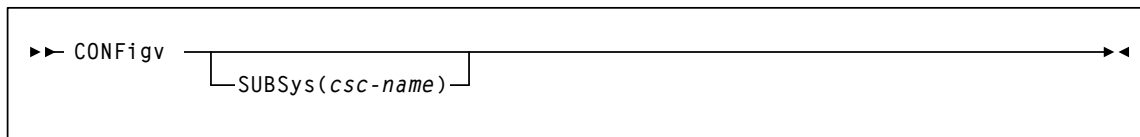
The Configuration Verification utility is primarily used to verify the values specified for the startup parameters at initialization time. If the optional SUBSYS parameter is specified, the Configuration Verification utility is used to verify that the LCS configuration is compatible with what is defined in the startup parameters.

The MVS/CSC modules must be installed before you can use the Configuration Verification utility.

## CONFIGV Control Statement

### Syntax

The syntax of the CONFIGV control statement is:



### Parameter Descriptions

This section describes the value you can specify with the CONFIGV control statement.

#### SUBSYS

Specifies the MVS/CSC subsystem name to test the LCS.

*csc-name*

specifies the name of the MVS/CSC subsystem as defined in the IEFSSNyy member of SYS1.PARMLIB.

---

**Note** – If you specify the SUBSYS parameter, both the MVS/CSC and LCS must be running.

---

## Example of CONFIGV Control Statement

The following example shows a control statement used to invoke the Configuration Verification utility for subsystem CSC1:

```
CONF SUBS (CSC1)
```

In this example, the Configuration Verification utility generates a report on the status of the MVS/CSC installation.

## Sample JCL

The following figure shows sample JCL used to invoke the Configuration Verification utility.

```
//CSC0CFGV JOB ACCT,PGMR
//STEP0    EXEC PGM=SCUADMIN, PARM='MIXED,DATE=4YR'
//STEPLIB  DD DISP=SHR,DSN=your.SCSLINK
//          DD DISP=SHR,DSN=your.TCPLINK /* optional TCP load modules */
//          DD DISP=SHR,DSN=your.SACLINK
//          DD DISP=SHR,DSN=your.CSLLINK
//SCSPARM  DD DISP=SHR,DSN=your.parmlib(csc-param-member)
//SCSPRINT DD SYSOUT=*
//SCSIN    DD *
           CONFIGV [SUBS (CSC1) ]
/*
//
```

**FIGURE 5-5** Sample JCL for Configuration Verification Utility

The JCL to invoke the Configuration Verification utility is contained in sample members JCLCFGV1 and JCLCFGV2 in the SAMPLIB library.

---

**Note** – If you are running MVS/CSC and MVS/HSC on the same MVS system, the STEPLIB DD statement that identifies the MVS/CSC load libraries is required (see [FIGURE 5-5](#)). If you omit the STEPLIB DD statement, unpredictable MVS/CSC operations may occur.

---

## Sample Output

Output resulting from the execution of the Configuration Verification utility includes:

- A listing of input control statements with appropriate messages when syntax errors occur. Sample output is shown in [FIGURE 5-6](#).
- A formatted printout of MVS/CSC startup parameters verified (when all parameters are valid). Sample output is shown in [FIGURE 5-7 on page 78](#).
- A formatted printout of MVS/CSC startup parameters verified including error conditions for invalid parameters. Sample output is shown in [FIGURE 5-8 on page 79](#).

```
SCUADMIN (7.0.0)   StorageTek MVS Client/Server System Utility           PAGE 001
TIME 10:51:02      Control Card Image Listing                                     DATE 2010-06-07

CONF SUBS(CSC1)
```

**FIGURE 5-6** Configuration Verification Utility Sample Output-Input Control Statement

```

SCUADMIN (7.0.0) StorageTek MVS Client/Server System Utility          PAGE 002
TIME 10:51:02 Configuration Verify                                DATE 2010-06-07

*****
*****
SERVER(CLS) * SERVER TYPE
LIBDEV(LIB0,LIB1) * LIBRARY ESOTERICS
LIBUNIT(10A0,10A1,10A2,10A3, - * LIBRARY DEVICES
          10A4,10A5,10A6,10A7, - * LIBRARY DEVICES
          10B0,10B1,10B2,10B3, - * LIBRARY DEVICES
          10B4,10B5,10B6,10B7) * LIBRARY DEVICES
UNITMAP(10A0,00:00:8:6,10A1,00:00:8:5, - * UNIT MAPPINGS
         10A2,00:00:8:8,10A3,00:00:8:7, - * UNIT MAPPINGS
         10A4,00:00:9:6,10A5,00:00:9:5, - * UNIT MAPPINGS
         10A6,00:00:9:8,10A7,00:00:9:7, - * UNIT MAPPINGS
         10B0,01:00:8:6,10B1,01:00:8:5, - * UNIT MAPPINGS
         10B2,01:00:8:8,10B3,01:00:8:7, - * UNIT MAPPINGS
         10B4,01:00:9:6,10B5,01:00:9:5, - * UNIT MAPPINGS
         10B6,01:00:9:8,10B7,01:00:9:7) * UNIT MAPPINGS
*
COM(TCPIP) * TCPIP COMMUNICATIONS
COMPRFX('/ /') * MVS/CSC COMMAND PREFIX
MSGCASE(MIXED) * OUTPUT CASE
SCRLABL(SL) * SCRATCH LABEL TYPE
TCPNAME(TCPIP) * IBM's TCP/IP ADDRESS SPACE NAME
INTERNET(129.80.41.126) * INTERNET ADDRESS
TRACDEST(LOG) * TRACE DESTINATION
Trace(YES) * TRACE ACTIVITY
LOG(RESET) * LOGGING OPTION

SCS0159I MVS/CSC Startup parameters verified
SCS0155I Condition code for utility function is 0
    
```

FIGURE 5-7 CONFIGV Sample Output - All Parameters Valid (VM-Based LCS with TCP/IP)

```

SCUADMIN (7.0.0) StorageTek MVS Client/Server System Utility          PAGE 002
TIME 10:51:02 Configuration Verify                                DATE 2010-06-07

*****
*****
SERVER(CLS) * SERVER TYPE
LIBDEV(LIB0,LIB1) * LIBRARY ESOTERICS
LIBUNIT(1A00,10A1,10A2,10A3, - * LIBRARY DEVICES
          10A4,10A5,10A6,10A7, - * LIBRARY DEVICES
          10B0,10B1,10B2,10B3, - * LIBRARY DEVICES
          10B4,10B5,10B6,10B7) * LIBRARY DEVICES
UNITMAP(10A0,00:00:8:6,10A1,00:00:8:5, - * UNIT MAPPINGS
         10A2,00:00:8:8,10A3,00:00:8:7, - * UNIT MAPPINGS
         10A4,00:00:9:6,10A5,00:00:9:5, - * UNIT MAPPINGS
         10A6,00:00:9:8,10A7,00:00:9:7, - * UNIT MAPPINGS
         10B0,01:00:8:6,10B1,01:00:8:5, - * UNIT MAPPINGS
         10B2,01:00:8:8,10B3,01:00:8:7, - * UNIT MAPPINGS
         10B4,01:00:9:6,10B5,01:00:9:5, - * UNIT MAPPINGS
         10B6,01:00:9:8,10B7,01:00:9:7) * UNIT MAPPINGS
*
COM(TCPIP) * TCPIP COMMUNICATIONS
COMPRFX('/ ') * MVS/CSC COMMAND PREFIX
MSGCASE(MIXED) * OUTPUT CASE
SCRLABL(SL) * SCRATCH LABEL TYPE
TCPNAME(TCPIP) * IBM's TCP/IP ADDRESS SPACE NAME
INTERNET(129.80.41.126) * INTERNET ADDRESS
TRACDEST(LOG) * TRACE DESTINATION
Trace(YES) * TRACE ACTIVITY
LOG(RESET) * LOGGING OPTION

SCS0836E SCSPARM parameter DEFER value invalid; must be YES, NO, or JES3
SCS0723E Library device 1A00 supplied in the SCSPARM LIBUNIT parameter not
         contained in any ACS esoteric
SCS0155I Condition code for utility function is 8

```

FIGURE 5-8 CONFIGV Sample Output - Some Invalid Parameters (VM-Based LCS with TCP/IP)

# Scratch Update Utility

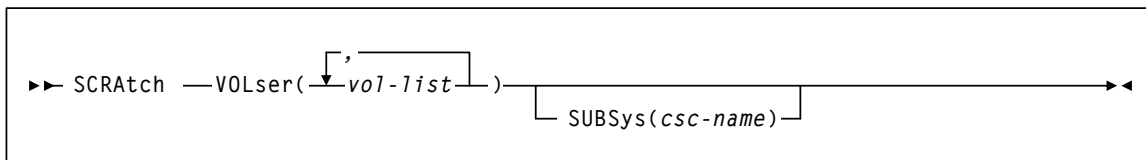
The Scratch Update utility accepts a list of volume serial numbers and generates a message to the LCS. The message directs the LCS to either add or delete the volume serial numbers from the existing scratch list contained in the LCS database.

The Scratch Update utility must be run with both the MVS/CSC and the LCS systems running.

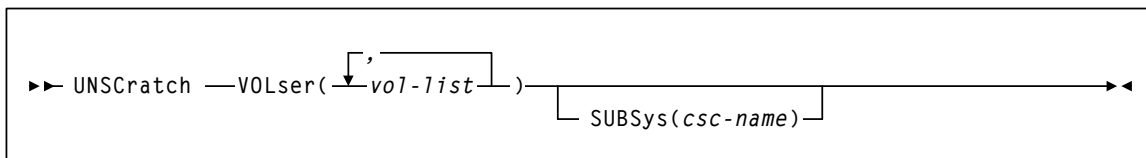
## SCRATCH and UNSCRATCH Control Statements

### Syntax

The syntax of the SCRATCH control statement is:



The syntax of the UNSCRATCH control statement is:



### Parameter Descriptions

This section describes the values you can specify with the SCRATCH and UNSCRATCH control statements.

#### **VOLSER**

Specifies the volume serial numbers to be added to or deleted from the scratch list.

*vol-list* is a list of volume serial numbers to be added to or deleted from the scratch list. The values specified can consist of a list of volume serial numbers, ranges of volume serial numbers, or combinations of lists and ranges delimited by commas.

#### **SUBSYS**

Specifies the name of the MVS/CSC subsystem to receive the scratch transaction. The default is the first active MVS/CSC subsystem in the SSCVT chain.

*csc-name* specifies the name of the MVS/CSC subsystem as defined in the IEFSSNyy member of SYS1.PARMLIB.



## Example of UNSCRATCH Control Statement

The following example shows a control statement used to invoke the Scratch Update utility:

```
UNSC VOL(U01180,U01183-U01185) SUBS(CSC0)
```

## Sample JCL

The following figure shows sample JCL used to invoke the Scratch Update utility.

```
//CSC0SCUP JOB ACCT,PGMR
//STEP0 EXEC PGM=SCUADMIN,PARM='MIXED,DATE=4YR'
//STEPLIB DD DISP=SHR,DSN=your.SCSLINK
// DD DISP=SHR,DSN=your.TCPLINK /* optional TCP load modules */
// DD DISP=SHR,DSN=your.SACLINK
// DD DISP=SHR,DSN=your.CSLLINK
//SCSPRINT DD SYSOUT=*
//SCSIN DD *
 SCRA VOLSER(vol-list) [SUBSYS(csc-name)]
/* or
 UNSC VOLSER(vol-list) [SUBSYS(csc-name)]
/*
//
```

**FIGURE 5-9** Sample JCL for Scratch Update utility

The JCL to invoke the Scratch Update utility is contained in sample member JCLSCRUP in the SAMPLIB library.

---

**Note** – If you are running MVS/CSC and MVS/HSC on the same MVS system, the STEPLIB DD statement that identifies the MVS/CSC load libraries is required (see [FIGURE 5-9](#)). If you omit the STEPLIB DD statement, unpredictable MVS/CSC operations may occur.

---

## Sample Output

Output resulting from execution of the Scratch Update utility includes the following:

- A listing of input control statements with appropriate messages when syntax errors occur. Sample output is shown in [FIGURE 5-10](#).
- Messages indicating any errors encountered during execution of the utility. Sample output is shown in [FIGURE 5-11](#).
- Messages indicating successful scratching or unscratching of specific volumes. Sample output is shown in [FIGURE 5-11](#).
- An update to the LCS database reflecting cartridges added to or deleted from the scratch lists.

```

SCUADMIN (7.0.0)  StorageTek MVS Client/Server System Utility          PAGE 001
TIME 10:51:02    Control Card Image Listing                                     DATE 2010-01-07

SCRA VOL(104212,104214,VACANT)

```

**FIGURE 5-10** Scratch Update Utility Sample Output-Input Control Statement

```

SCUADMIN (7.0.0)  StorageTek MVS Client/Server System Utility          PAGE 002
TIME 10:51:02    Scratch Update                                               DATE 2010-01-07

SCS0167I Volume 104212 successfully added to library as scratch
SCS0168I Volume 104214 successfully deleted from library scratch
pool
SCS0163E Volume VACANT not in library or not in LCS Scratch Subpool definition
SCS0155I Condition code for utility function is 4

```

**FIGURE 5-11** Scratch Update Utility Sample Output-Messages

---

# Scratch Conversion Utility

The Scratch Conversion utility allows a tape management system to indicate to the LCS which of its volumes are in scratch status. The Scratch Conversion utility is invoked by the SCUCONDB program. It is the only MVS/CSC utility not invoked by SCUADMIN. You can run the Scratch Conversion utility regardless of whether or not the MVS/CSC is running.

SCUCONDB 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, SCUCONDB calls the appropriate tape management system database READ routine to produce a list of volume serial numbers for scratch cartridges. For Zara, SCUCONDB invokes the SLUDRZAR module, which in turn invokes an application programming interface (API) to read Zara's database and produce a list of volume serial numbers for scratch cartridges. The format of these lists is acceptable as input to the Scratch Update utility.

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.

## SLUDRTLM

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 SMC SMCLINK distribution library.

---

If your installation uses a different tape management system, you can develop a program (using the source code provided in the HSC sample library) to process the tape management system database into a format acceptable for the Scratch Update utility. Alternatively, you can manually generate the control statements and run the Scratch Update utility stand-alone. For another alternative, refer to the *ELS Command, Control Statement, and Utility Reference* for information about the SMCUDBX scratch synchronization utility.

SCUCONDB, 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.

---

**Note –**

- 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 SCUCONDB with CA-DYNAM/TLMS Release 5.3 or Below” on page 85](#) 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 SCUCONDB with CA-DYNAM/TLMS Release 5.3 or Below” on page 85](#) for information about reassembling this routine.
  - If your tape management system is Zara, the SCUCONDB 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 SCUCONDB 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 SCUCONDB with CA-1 Release 4.9 or Below

To use SCUCONDB 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 SCUCONDB with CA-DYNAM/TLMS Release 5.3 or Below

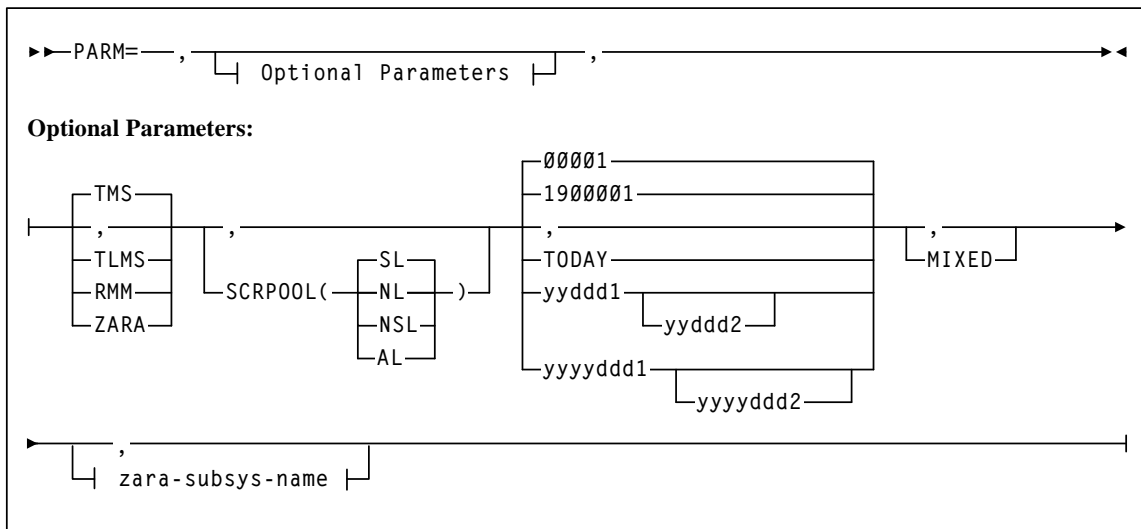
Perform the following steps to use SCUCONDB with CA-DYNAM/TLMS Release 5.3 or below:

1. Verify that the CA-DYNAM/TLMS Release 5.3 MACLIB containing the VMFBASE MACRO is accessible for the re-assembly of the SLUDRTL routine.
2. Reassemble the SLUDRTL 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 Conversion 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 pool used by the MVS/CSC. The value specified for this keyword must match the value specified for the SCRLABL startup parameter. Acceptable values are:

**SL** specifies standard label. This is the default value.

**NL** specifies nonlabeled.

**AL** specifies ANSI label.

**NSL** specifies nonstandard label.

### 00001 | 1900001

Specifies January 1, 1900 as the date. This is the default date and can be specified in either two-year (00001) or four-year (1900001) format.

---

**Note** – You must specify a four-year format for year 2000 dates. For example, specify 2010001 for January 1, 2010.

---

### TODAY

Specifies the current system date.

### *yyddd1 | yyyyddd1*

Specifies a single date in either two-year or 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).

### *yyddd1 - yyddd2 | yyyyddd1-yyyddd2*

Specifies a range of dates in which the volumes became scratch. You can specify either a two-year or four-year format for the range of dates. 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 (00001 | 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.

This parameter is required for the Zara tape management system since SCUCONDB does not read the Zara database directly. This parameter is ignored if specified for any other tape management system.

## Sample JCL

The following definitions apply to the JCL used for the Scratch Conversion utility to process scratch report files.

### SCSTMS

Input to the Scratch Conversion utility in the form of the CA-1 TMC, the CA-DYNAM/TLMS VMF, or the DFSMSrmm report extract file.

For Zara, you must specify a DUMMY DD for SCSTMS, because SCUCONDDB does not read the Zara database directly. For example:

```
//SCSTMS DD DUMMY
```

### SCSSOUT

Output from the Scratch Conversion utility specifying the Scratch Update utility control statement.

### SCSPRINT

Output messages from the Scratch Conversion utility program.

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 19 shows sample JCL used to invoke the Scratch Conversion utility to process the CA-DYNAM/TLMS VMF for all tapes scratched in 1998.

```
//SCUCNVT JOB ACCT,PGMR
//STEP0 EXEC PGM=SCUCONDDB,PARM='TLMS,,1998001-1998365'
//STEPLIB DD DISP=SHR,DSN=your.SCSLINK
//SCSTMS DD DISP=SHR,DSN=your.tlms.database
//SCSSOUT DD DISP=(NEW,KEEP),DSN=scratch.update.trans,
//          SPACE=(TRK,(5,1)),UNIT=SYSDA
//SCSPRINT DD SYSOUT=*
/*
```

**FIGURE 5-12** Sample JCL for Scratch Conversion Utility

The JCL to invoke the Scratch Conversion utility is contained in sample member JCLCONDDB in the SAMPLIB library.



## Sample Output

Output resulting from the execution of the Scratch Conversion utility includes:

- Messages associated with error conditions resulting from an unsuccessful attempt to execute scratch conversion processing.
- Messages indicating errors or actions occurring during processing.
- An output data set containing Scratch Update utility control statements.

For example, Figure 20 shows the output that would be produced if an output data set contained the following Scratch Update utility control statement.

```
SCRATCH VOL(S(U01180,U01181,U01187,U01188))
```

```
SCUCONDB      StorageTek MVS Client/Server Conversion Utility      Page 002
Time 10:51:02                                     Date 2010-01-07

SCS0286I 00004 volumes have been selected
SCS0155I Condition code for utility function is 0
```

**FIGURE 5-13** Scratch Conversion Utility Sample Output - Messages



# Performance and Operational Considerations

---

---

## Overview

This chapter describes the following performance and operational considerations:

- Performance considerations
- Operational constraints
- Sharing MVS/CSC-controlled transports
- Third-party software restrictions
- Write protection and virtual thumbwheel processing

## Performance Considerations

This section describes the following factors that can affect MVS/CSC performance:

- Redistributing scratch cartridges
- Maintaining quantities of scratch cartridges
- Reducing scheduling contention
- Reducing pass-through activity

## Redistributing Scratch Cartridges

In the VM-based LCS environment, the HSC Scratch Redistribution utility can be used to redistribute scratch cartridges across the LSMs within a particular ACS. You can choose to have the scratch cartridges go to only specified LSMs or all LSMs within an ACS. If necessary, the Scratch Redistribution utility swaps scratch cartridges with nonscratch cartridges to equalize the count of scratch cartridges in an ACS.

In the UNIX-based LCS environment, you can manually scratch and redistribute cartridges in order to equalize the distribution of scratch cartridges. You can then run volume reports to verify the distribution.

## Maintaining Quantities of Scratch Cartridges

Scratch cartridges are maintained in scratch pools that are defined across the library LSMs. The number of cartridges maintained in the scratch pools should be sufficient for normal library operation at your data center.

In CLS server environments, the HSC Warn command can be used to control when the HSC warns of low quantities of scratch cartridges. The Warn command establishes thresholds that control when the HSC notifies the operator of an insufficient quantity of scratch cartridges in an ACS. The scratch pool depletion warning notifies the operator when the number of scratch volumes in an ACS falls below a specified minimum value. By specifying warning thresholds, you can know when the number of scratch cartridges is too low for normal library operation.

## Reducing Scheduling Contention

Effective scheduling can increase library performance. Scheduling activities that require library robotic functions and server functions away from peak production times can help improve performance time. Controlling the following scheduling related areas can help significantly in further increasing library performance:

- Balancing the workload
- Scheduling library activity

### Balancing the Workload

If you experience mount times that are higher than average, but at the same time experience an improvement in production through-put, it could be an indication that your system is periodically flooded with work rather than having a balanced workload.

For example, all of your production jobs may be getting submitted at the beginning of a shift so that the library robots are overworked for the first few hours of the shift. Then the robots may remain idle for the remainder of the shift. If you are using a scheduling software package and it releases jobs every hour on the hour, there may be contention for transports for the first few minutes of each hour while the transports are unused for the remainder of each hour.

These situations tend to increase average mount times; however, as long as the work is performed on time, there is probably no need to change conditions. However, if the work is not getting performed on schedule, you can improve performance by balancing your production workload.

### Scheduling Library Activity

Library activities that require many pass-thrus (for example, redistributing scratch cartridges, moving large quantities of cartridges, or entering and ejecting many cartridges) can impact mount and dismount time. These activities should be scheduled during low periods so they do not contend with mounts and dismounts for high-priority production jobs. Running these activities during low periods also ensures that the tasks complete faster. In the case of large quantities of enters or ejects, operator time can also be optimized.

## Reducing Cartridge Exchange Activity

Scratch mounts, dismounts, enters, and ejects that require cartridge exchanges are unnecessary and should be avoided. These types of activities divert the robot from productive work, especially during periods of peak activity.

If scratch subpools are correctly defined and managed, scratch cartridges normally are not involved in cartridge exchanges. Scratch mounts are the same as specific mounts in that exchanges often cannot be avoided. In general, scratch cartridges should not require a cartridge exchange.

---

## Operational Constraints

This section describes the following restrictions that can affect MVS/CSC operations:

- Cartridge naming restrictions
- Transport address restrictions
- Association with a Library Control System (LCS)
- Message processing (WTO responses)

### Cartridge Naming Restrictions

Each cartridge in any library must have a unique volume serial number; no duplicate volume serial numbers are allowed. Volume serial numbers are contained on the external labels of each cartridge. If the cartridge is not labeled or the label cannot be read, a unique volume serial number must be assigned to the cartridge when it is entered into the LSM.

- For the VM server, you can enter the HSC Enter command from the MVS client console; the HSC responds with a message on the MVS client console prompting for a *volser*.
- For the ACSLS server, you can enter the ACSLS Venter command from the ACS System Administrator (ACSSA) console. The Venter command requires that you enter a *capid* and *volser*. The specified *volser* is permanently assigned to the cartridge in the library database.

Cartridges in (1) libraries controlled by multiple MVS/CSC subsystems in an MVS host system, or (2) libraries controlled by an MVS/CSC coexisting with an HSC on an MVS host system, must have unique volume serial numbers across all libraries.

Cartridge volume serial numbers must follow the ACS hardware guidelines. Valid volume serial numbers are limited to the following characters:

- Letters A to Z
- Numbers 0 to 9
- Trailing blanks
- National characters: #, @, and \$

---

**Note –**

- The at sign (@) character is only valid for the VM-based LCS.
  - The yen (¥) and pound sterling (£) characters correspond to the same EBCDIC values as the dollar sign (\$) and pound sign (#) characters (X'5B' and X'7B'), and can also be used in volume serial numbers.
- 

The ACS does not recognize volume serial numbers that contain characters other than those listed above, including leading blanks.

## Transport Address Restrictions

Each MVS/CSC must have unique library transports available for processing. No two library transports can respond to the same address from the same host. No single library transport can be controlled by an MVS/CSC and an HSC on the same MVS host system.

## Association with a Library Control System (LCS)

Each MVS/CSC subsystem must be associated with a separate LCS operating on a different server system. An alternate LCS can be defined to the MVS/CSC if two Internet addresses and port numbers have been specified (using the INTERNET and PORT startup parameters for CLS). Alternate ACSLS servers are not supported.

---

# Sharing MVS/CSC-controlled Transports

MVS/CSC-controlled transports can be shared with other MVS and non-MVS host systems. These transports can be shared by manually varying devices online/offline between systems, or in some cases shared dynamically using MIM or automatic tape switching.

The following sections describe how transports can be shared between MVS and non-MVS host systems.

---

**Note –** There are certain restrictions for defining devices used in automatic tape switching. Refer to *MVS/ESA Planning: Installation and Migration* for these restrictions.

---

## Sharing Transports Between MVS/CSC Clients on Different MVS Host Systems

In order to share transports between MVS/CSC clients residing on different MVS host systems, each MVS/CSC must be attached to the same LCS. MVS/CSC-controlled transports can be shared by manually varying transports online/offline between MVS systems, or shared dynamically using MIM or automatic tape switching.

A transport can be online to only one MVS system at a time when manually varying MVS/CSC-controlled transports online/offline between MVS/CSC subsystems. After manually varying an MVS/CSC-controlled transport online, you must issue the MVS/CSC RESYNCH operator command to synchronize the current state of the device with the current state of the MVS/CSC. If using MIM, or automatic tape switching for device sharing, MVS/CSC-controlled transports are online to both systems simultaneously.

## Sharing Transports Between MVS/CSC Clients and Non-MVS Clients

In order to share transports between MVS/CSC clients and non-MVS clients, each client sharing transports must either be attached to the same LCS, or attached to multiple LCSs sharing the same library. MVS/CSC-controlled transports can only be shared by manually varying transports online/offline between clients. MVS/CSC-controlled transports cannot be shared dynamically in this environment.

A transport can be online to only one client at a time when manually varying MVS/CSC-controlled transports online/offline between clients. After manually varying an MVS/CSC-controlled transport online, you must issue the MVS/CSC RESYNCH operator command to synchronize the current state of the device with the current state of the MVS/CSC.

## Sharing Transports Between MVS/CSC and HSC's Residing on Different MVS Systems

Library transports can be shared between MVS/CSC and HSC's residing on different MVS systems. However, MVS systems running the MVS/CSC can only be attached to one LCS running HSC. MVS/CSC-controlled transports and HSC-controlled transports can be shared by manually varying transports online/offline between the MVS/CSC and HSC subsystems, or shared dynamically using MIM or automatic tape switching.

A transport can be online to only one MVS system at a time when manually varying transports online/offline between MVS/CSC and HSC subsystems. After manually varying a transport online, you must issue the MVS/CSC RESYNCh operator command to synchronize the current state of the device with the current state of the MVS/CSC. If using MIM or automatic tape switching for device sharing, MVS/CSC-controlled transports and HSC-controlled transports are online to both systems simultaneously.

---

**Note** – MVS/CSC-controlled transports cannot be shared by both the MVS/CSC and HSC if they reside on the same MVS system.

---

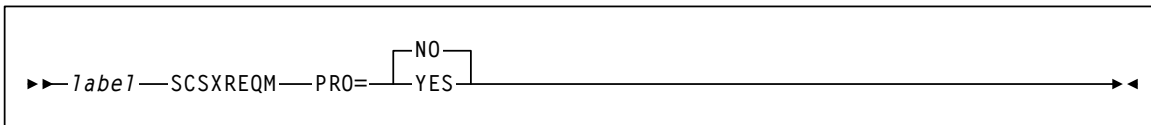




## Programmatic Interface SCSXREQM Macro Mappings

The SCSXREQM macro must be specified in any program assembly that issues the SCSXREQ macro. SCSXREQM generates symbols used by the SCSXREQ macro, and generates the DSECTs of the response areas. The SCX DSECT maps the reply header. The remaining DSECTs map the information elements returned from Programmatic Interface (PGMI) requests. The prologue (if requested) in each information element lists the PGMI request that it is returned for.

The syntax for the SCSXREQM macro is:



**PRO**

Specifies whether or not to generate a macro prologue.

**YES**

Generates a macro prologue.

**NO**

Does not generate a macro prologue.

# SCSXREQM Macro Mappings

TABLE A-1 SCSXREQM Macro

Dec	Hex	Type	Len	Label	Description
0	(0)	STRUCTURE		SCSXREQM	USER INTERFACE
0	(0)	CHARACTER	4	SCSXHDR	EYECATCHER
'SCSX'	(E2D3E2E7 )	CHAR CONST		SCSXID	IS AN SCSXREQ
4	(4)	SIGNED-FWORD	4	SCSXLEN	LENGTH OF CONTROL BLOCK
8	(8)	A-ADDR	1	SCSXVER	VERSION OF SCSXREQ
6	(06)	CONST		SCSXVN	CURRENT VERSION
230	(E6)	BINARY CONST		SCSXSP	SUB-POOL NUMBER
THE FOLLOWING FLAGS DEFINE THE TYPE OF PROCESSING REQUESTED					
<b>NOTE:</b> ONLY A LIMITED SUBSET OF THESE FLAGS ARE CURRENTLY USED BY MVS/CSC. THEY ARE THE FLAGS REQUIRED TO SUPPORT THE QCSC, QVOLUME, QDRIVES, AND READ FUNCTIONS. ALL OTHER FLAGS WILL REMAIN DEFINED FOR POSSIBLE FUTURE USE.					
9	(9)	A-ADDR	1	SCSXRT	REQUEST TYPE
0	(00)	CONST		SCSXNOOP	NOOP - NO OPERATION
1	(01)	CONST		SCSXQCSC	QCSC - IS CSC UP?
2	(02)	CONST		SCSXREST	RESET - DISMOUNT ALL VOL
3	(03)	CONST		SCSXREAD	READ - READ A RESPONSE
4	(04)	CONST		SCSXSTOP	STOP - STOP A PROCESS
5	(05)	CONST		SCSXRS05	Reserved - VM Only
20	(14)	CONST		SCSXQCNF	QCONFIG - QUERY CONFIG
21	(15)	CONST		SCSXQDRV	QDRIVES - QUERY DRIVES
22	(16)	CONST		SCSXQDRL	QDRLIST - QUERY DRIVES
23	(17)	CONST		SCSXQSCR	QSCRATCH - QUERY SCRATCH
24	(18)	CONST		SCSXQVOL	QVOLUME - QUERY VOLUME
25	(19)	CONST		SCSXQEJT	QEJECT - QUERY EJECT TASK
26	(1A)	CONST		SCSXQCAP	QCAP - QUERY CAP CONFIG
27	(1B)	CONST		SCSXRS17	Reserved - VM Only
28	(1C)	CONST		SCSXQRQS	QREQUEST - QUERY REQUESTS
29	(1D)	CONST		SCSXQDSN	QDSN - QUERY DATASET

TABLE A-1 SCSXREQM Macro (Continued)

Dec	Hex	Type	Len	Label	Description
40	(28)	CONST		SCSXMMNT	MOUNT - MOUNT A VOLUME
41	(29)	CONST		SCSXMOVE	MOVE - MOVE A CARTRIDGE
42	(2A)	CONST		SCSXDSM	DISMOUNT - DISMOUNT A VOLUME
43	(2B)	CONST		SCSXJECT	EJECT - EJECT A VOLUME
60	(3C)	CONST		SCSXSSCR	SELSCR - SELECT SCRATCH
61	(3D)	CONST		SCSXSCRA	SCRATCH - SCRATCH A VOLUME
62	(3E)	CONST		SCSXUNSC	UNSCRATCH - UNSCRATCH A VOLUME
63	(3F)	CONST		SCSXRS63	Reserved - VM Only
64	(40)	CONST		SCSXRS64	Reserved - VM Only
10	(A)	BITSTRING	1	SCSXFLG1	FLAG BYTE 1 (PGMI Control 1)
		1... .... X'80'		SCSXSYNC	OPTION=SYNC SPECIFIED
		.1. .... X'40'		SCSXF140	Reserved VM User=option
		..1. .... X'20'		SCSXF120	Reserved VM Dialog OFF
		...1 .... X'10'		SCSXF110	Reserved VM Dialog ON
		.... 1... X'08'		SCSXF108	Reserved VM Notify=INSEDL
		.... .1. X'04'		SCSXF104	Reserved VM Notify=NOINSEDL
		.... ..1. X'02'		SCSXF102	Reserved VM ACCT2
		.... ...1 X'01'		SCSXF101	Reserved VM ACCT1
11	(B)	HEXSTRING	1	SCSXFLG2	FLAG BYTE 2 (PGMI Control 2)
12	(C)	BITSTRING	1	SCSXFLG3	FLAG BYTE 3 (Movement Control 1)
		1... .... X'80'		SCSXVHST	SCSXHOST CONTAINS A HOSTID
		.1. .... X'40'		SCSXVACS	
		..1. .... X'20'		SCSXVLSM	SCSXLSMI CONTAINS THE LSMID
		...1 .... X'10'		SCSXVCAP	SCSXCAP CONTAINS CAP
		.... 1... X'08'		SCSXF308	Reserved Future Use
		.... .1. X'04'		SCSXVVOL	SCSXVOLS CONTAINS A VOLSER
		.... ..1. X'02'		SCSXVLST	SCSXLISTA CONTAINS A LIST ADR
		.... ...1 X'01'		SCSXVCNT	SCSXCNT CONTAINS LIST COUNT

TABLE A-1 SCSXREQM Macro (Continued)

Dec	Hex	Type	Len	Label	Description
13	(D)	BITSTRING	1	SCSXFLG4	FLAG BYTE 4 (Movement Control 2)
		1... .... X'80'		SCSXVTLM	SCSXTLISM CONTAINS TOLISM
		.1.. .... X'40'		SCSXVTPN	SCSXTPAN CONTAINS TOPANEL
		..1. .... X'20'		SCSXF420	Reserved Future Use
		...1 .... X'10'		SCSXF410	Reserved Future Use
		.... 1... X'08'		SCSXVPAN	SCSXPAN CONTAINS PANEL #
		.... .1.. X'04'		SCSXVROW	SCSXROW CONTAINS ROW #
		.... ..1. X'02'		SCSXVCOL	SCSXCOL CONTAINS COLUMN #
		.... ...1 X'01'		SCSXVDRV	SCSXDRIV CONTAINS DEVICE NUMBER
14	(E)	HEXSTRING	1	SCSXFLG5	FLAG BYTE 5 (Movement Control 3)
15	(F)	BITSTRING	1	SCSXFLG6	FLAG BYTE 6 (Miscellaneous Control 1)
		1... .... X'80'		SCSXPROT	PROTECT=YES SPECIFIED
		.1.. .... X'40'		SCSXSCR	SCRATCH=YES SPECIFIED
		..1. .... X'20'		SCSXVSCP	SCSXPOOL HAS A SCRATCH POOL NUM
		...1 .... X'10'		SCXSUBN	SCXSUBP CONTAINS TEXT
		.... 1... X'08'		SCSXVTKN	SCSXTKNO CONTAINS TOKEN NUM
		.... .1.. X'04'		SCSXVTXT	SCSXTEXT CONTAINS TEXT
		.... ..1. X'02'		SCSXF602	Reserved Future Use
		.... ...1 X'01'		SCSXF601	Reserved Future Use
16	(10)	HEXSTRING	1	SCSXFLG7	FLAG BYTE 7 (Miscellaneous Control 2)
		1... .... X'80'		SCSXSSN	CSC SUBSYSTEM NAME SPECIFIED
17	(11)	BITSTRING	1	SCSXLABT	LABEL TYPE
0	(00)	CONST		SCSXLLDT	LDT (LIBRARY DEFAULT TYPE)
1	(01)	CONST		SCSXLSL	SL (STANDARD LABEL)
2	(02)	CONST		SCSXLAL	AL (ANSI LABEL)
3	(03)	CONST		SCSXNLN	NL (NONLABELED)
4	(04)	CONST		SCSXLNS	NSL (NONSTANDARD LABEL)
18	(12)	CHARACTER	8	-RESERVED-	Reserved VM User
26	(1A)	CHARACTER	8	-RESERVED-	Reserved VM Acct1

TABLE A-1 SCSXREQM Macro (Continued)

Dec	Hex	Type	Len	Label	Description
34	(22)	CHARACTER	8	-RESERVED-	Reserved VM Acct2
44	(2C)	SIGNED-FWORD	4	SCSXRLN	RESPONSE LENGTH
48	(30)	SIGNED-FWORD	4	SCSXTKNO	PGMI PASS THRU TOKEN NUMBER
52	(34)	A-ADDR	4	SCSXLSTA	ADDRESS OF LIST
56	(38)	SIGNED-HWORD	2	SCSXCNT	COUNT OF ELEMENTS IN LIST
58	(3A)	SIGNED-HWORD	2	SCSXLOFF	OFFSET, FROM START OF PARAMETER LIST TO START OF THE ELEMENT LIST AREA
60	(3C)	CHARACTER	6	SCSXVOLS	VOLSER FROM VOLSER= PARAMETER
66	(42)	CHARACTER	8	SCSXHOST	ASSOCIATED HOSTID
74	(4A)	AREA	4	SCSXALC	ACS / LSM / CAP IDENTIFICATION
74	(4A)	HEXSTRING	1	SCSXACS	ACS ID NUMBER
74	(4A)	HEXSTRING	2	SCSXLSMI	LSM ID NUMBER
74	(4A)	HEXSTRING	4	SCSXCAP	CAP-ID
77	(4D)	HEXSTRING	1	-RESERVED-	NOT IMPLEMENTED (ALWAYS X'00')
78	(4E)	HEXSTRING	1	SCSXSPAN	CELL- PANEL
79	(4F)	HEXSTRING	1	SCSXROW	CELL- ROW
80	(50)	HEXSTRING	1	SCSXCOL	CELL- COLUMN
81	(51)	HEXSTRING	2	SCSXTLSM	TO LSM ID FOR MOVE FUNCTION
83	(53)	HEXSTRING	1	SCSXTPAN	"TO" PANEL FOR MOVE FUNCTION
84	(54)	A-ADDR	2	SCSXDRIV	DRIVE DEVICE NUMBER
86	(56)	A-ADDR	1	SCSXPOOL	SCRATCH POOL NUMBER
87	(57)	CHARACTER	32	SCSXTEXT	TEXT ASSOCIATED WITH REQUEST
119	(77)	CHARACTER	13	SCSXSUBP	SCRATCH SUBPOOL NAME
<b>NOTE:</b> THE FOLLOWING FIELDS ARE CURRENTLY UNUSED BY MVS/CSC, BUT WILL REMAIN DEFINED FOR POSSIBLE FUTURE USE.					
TAPEREQ INPUT KEY VALUES.					
THE FOLLOWING VALUES ARE USED AS THE KEY IN SEARCHING THE CURRENT TAPEREQ PARAMETER FILE.					

TABLE A-1 SCSXREQM Macro (Continued)

Dec	Hex	Type	Len	Label	Description
132	(84)	BITSTRING	1	SCSXFLG8	FLAG BYTE 8 (TAPEREQ CONTROL 1)
		1... .... X'80'		SCSXFJOB	SCSXJOBN PRESENT
		.1.. .... X'40'		SCSXFSTP	SCSXSTEP PRESENT
		..1. .... X'20'		SCSXFPGM	SCSXPGMN PRESENT
		...1 .... X'10'		SCSXFDSN	SCSXDSN PRESENT
		.... 1... X'08'		SCSXFEXP	SCSXEXPD PRESENT
		.... .1.. X'04'		SCSXFRET	SCSXRETP PRESENT
		.... ..1. X'02'		SCSXFVOL	SCSXVOLT PRESENT
133	(85)	CHARACTER	8	SCSXJOBN	STRING TO MATCH TAPEREQ JOBNAME VALUE
141	(8D)	CHARACTER	8	SCSXSTEP	STRING TO MATCH TAPEREQ STEPNAME VALUE
149	(95)	CHARACTER	8	SCSXPGMN	STRING TO MATCH TAPEREQ PROGNAME VALUE
157	(9D)	CHARACTER	44	SCSXDSN	STRING TO MATCH TAPEREQ DSN VALUE
201	(C9)	HEXSTRING	3	SCSXEXPD	VALUE TO MATCH TAPEREQ EXPDT VALUE
204	(CC)	HEXSTRING	2	SCSXRETP	VALUE TO MATCH TAPEREQ RETPD VALUE
206	(CE)	CHARACTER	1	SCSXVOLT	VALUE TO MATCH TAPEREQ VOLTYPE VALUE
<b>NOTE:</b> THE FOLLOWING FIELDS ARE CURRENTLY UNUSED BY MVS/CSC, BUT WILL REMAIN DEFINED FOR POSSIBLE FUTURE USE.					
TAPEREQ OUTPUT VALUES.					
THE FOLLOWING VALUES ARE USED AS OVERRIDE (OR SPECIFIC) VALUES TO THE VALUES FOUND IN THE CURRENT TAPEREQ PARAMETER FILE.					
207	(CF)	BITSTRING	1	SCSXFLG9	FLAG BYTE 9 (TAPEREQ CONTROL 2)
		1... .... X'80'		SCSXFREC	SCSXRECT PRESENT
		.1.. .... X'40'		SCSXFMED	SCSXMED PRESENT
208	(D0)	CHARACTER	8	SCSXRECT	RECORDING TECHNIQUE
216	(D8)	CHARACTER	8	SCSXMED	MEDIA
224	(E0)	CHARACTER	4	SCSXNAME	CSC SUBSYSTEM NAME
228	(E4)	AREA	252	-RESERVED-	FUTURE PARM EXPANSION

TABLE A-1 SCSXREQM Macro (Continued)

Dec	Hex	Type	Len	Label	Description
480	(1E0)	AREA	1	SCSXLIST	ELEMENT LIST DESIGNATED BY LIST= PARAMETER BEGINS HERE (FIELD SCSXCNT CONTAINS NUMBER OF ELEMENTS IN THIS LIST)
6	(06)	CONST		SCSXLVSL	LENGTH OF EACH ELEMENT (VOLSER) IN THE LIST USED BY THE QVOLUME FUNCTION
500	(1F4)	CONST		SCSXMVSL	MAXIMUM NUMBER OF ELEMENTS ALLOWED IN THE LIST USED BY THE QVOLUME FUNCTION
4	(04)	CONST		SCSXLCAP	LENGTH OF EACH ELEMENT (CAP IDENTIFIER) IN THE LIST USED BY THE QCAP FUNCTION
500	(1F4)	CONST		SCSXMCAP	MAXIMUM NUMBER OF ELEMENTS ALLOWED IN THE LIST USED BY THE QCAP FUNCTION
4	(04)	CONST		SCSXLTK	LENGTH OF EACH ELEMENT (UNIQUE TOKEN) IN THE LIST USED BY THE QCAP FUNCTION
500	(1F4)	CONST		SCSXMTOK	MAXIMUM NUMBER OF ELEMENTS ALLOWED IN THE LIST USED BY THE QEJECT/STOP FUNCTIONS
256	(100)	CONST		SCSXMCSC	MAXIMUM NUMBER OF CSC NAME INFORMATION ELEMENTS THAT CAN BE RETURNED ON A QCSC REQUEST
REPLY HEADER					
0	(0)	STRUCTURE		SCX	
0	(0)	AREA	1	SCXRPLY	
0	(0)	CHARACTER	3	SCXHID	HEADER IDENTIFIER
230	(E6)	BINARY CONST		SCXSP	SUB-POOL NUMBER
3	(3)	A-ADDR	1	SCXCMDRC	RETURN CODE
0	(00)	CONST		SCXROK	REQUEST SUCCESSFUL
4	(04)	CONST		SCXRWARN	REQUEST SUCCESSFUL WITH WARNING - SCXSRC WILL PROVIDE THE SPECIFIC REASON FOR THE WARNING

TABLE A-1 SCSXREQM Macro (Continued)

Dec	Hex	Type	Len	Label	Description
8	(08)	CONST		SCXRBADP	REQUEST FAILED - THE PARAMETER LIST (SCSXREQM) CONTAINED INVALID DATA (E.G., INCOMPATIBLE OPTIONS) SCXPEOFF (PARM ERROR OFFSET) WILL PROVIDE AN OFFSET OF THE SCSXREQM FIELD FOUND IN ERROR
12	(0C)	CONST		SCXRIERR	REQUEST FAILED - AN UNRECOVERABLE INTERNAL ERROR OCCURRED WHILE PROCESSING THE REQUEST
16	(10)	CONST		SCXRFAIL	REQUEST FAILED - SCXSRC WILL PROVIDE THE SPECIFIC REASON FOR THE FAILURE
20	(14)	CONST		SCXRNCSC	REQUEST FAILED - CSC NOT AVAILABLE (MVS ONLY)
24	(18)	CONST		SCXRNAUT	REQUEST FAILED - USER NOT AUTHORIZED. THE REQUEST WAS FAILED BY SCSUX05 OR THE CSC DEFAULT AUTHORIZATION
28	(1C)	CONST		SCXNTCB	REQUEST FAILED - THE CALLER WAS NOT IN TASK MODE (MVS ONLY)
32	(20)	CONST		SCXLOCKD	REQUEST FAILED - THE CALLER HELD AT LEAST ONE LOCK (MVS ONLY)
36	(24)	CONST		SCXTPROT	
REQUEST FAILED - THE CALLER DID NOT HAVE FETCH/STORE AUTHORITY TO THE SCSXREQM PARAMETER LIST (MVS ONLY)					
40	(28)	CONST		SCXNORSP	REQUEST FAILED - NO RESPONSE ELEMENT WITH THE SPECIFIED TOKEN EXISTED (MVS ONLY)
4	(4)	A-ADDR	1	SCXVERS	REPLY VERSION CODE
7	(07)	CONST		SCXVCODE	VERSION 7 OF THE REPLY AREA
5	(5)	BITSTRING	1	SCXRFLG1	RESPONSE FLAGS
		1... .... X'80'		SCXR0KM	MORE RESPONSES CAN BE READ
6	(6)	HEXSTRING	2	-RESERVED-	RESERVED
8	(8)	SIGNED-FWORD	4	SCXSRC	FAILURE/WARNING REASON CODE



TABLE A-1 SCSXREQM Macro (Continued)

Dec	Hex	Type	Len	Label	Description
28684	(700C)	CONST	4	SCXVNF	ONE OR MORE VOLUMES WERE NOT FOUND IN THE LIBRARY ON A QVOLUME REQUEST
32772	(8004)	CONST		SCXNOALS	CSC IS NOT ACTIVE
32776	(8008)	CONST		SCXMCSC	MULTIPLE CSCs ARE ACTIVE
32788	(8014)	CONST		SCXQNOA	ASCOMM IS NOT ACTIVE
32792	(8018)	CONST		SCXINVOP	INVALID QUAB OPTION INTERNAL ERROR - CONTACT STK.
32780	(800C)	CONST		SCXINVFC	INVALID FUNCTION CODE INTERNAL ERROR - CONTACT STK.
32784	(8010)	CONST		SCXNOLVT	NO LVT INTERNAL ERROR - CONTACT STK.
32796	(801C)	CONST		SCXNODRV	NO DRIVES FOUND ON QDRIVE CALL THAT ARE OWNED BY THE PROCESSING CSC
32804	(8024)	CONST		SCXTABND	ASCOMM SERVER TASK ABENDED
32818	(8032)	CONST		SCXTINTR	CURRENT TASK INTERRUPTED
32822	(8036)	CONST		SCXSANF	SEARCH ARGUMENT NOT FOUND
32826	(803A)	CONST		SCXMSTT	MIS-MATCHED TOKEN TYPES
32832	(8040)	CONST		SCXTRNF	TOKEN REQUEST AREA NOT FOUND
32848	(8050)	CONST		SCXSFUL	REPLY AREA SPACE FULL
32849	(8051)	CONST		SCXDVMM	DRIVE/MEDIA MISMATCH - THE REQUESTED MEDIA IS NOT COMPATIBLE WITH THE DRIVE SPECIFIED IN THE PGMI CALL
32850	(8052)	CONST		SCXCMAX	MAX NUMBER OF ACTIVE CSC'S FOUND WHILE PROCESSING A QCSC REQUEST
32851	(8053)	CONST		SCXSRVU	THE LCS BECAME UNAVAILABLE BEFORE OR DURING A REQUEST. THE RESPONSE MAY NOT CONTAIN ALL EXPECTED INFORMATION ELEMENTS
32852	(8054)	CONST		SCXUNSP	THE REQUEST TYPE IS NOT SUPPORTED IN THIS CSC'S SERVER ENVIRONMENT

TABLE A-1 SCSXREQM Macro (Continued)

Dec	Hex	Type	Len	Label	Description
32853	(8055)	CONST		SCXLMAX	THE NUMBER OF LIST ITEMS SPECIFIED BY THE REQUESTOR EXCEEDS THE MAXIMUM (SEE SCSXMVSL)
32854	(8056)	CONST		SCXLCSF	THE LCS FAILED THE REQUEST SENT BY MVS/CSC; SCXLCSRC CONTAINS THE SERVER REASON CODE
12	(0C)	A-ADDR	4	SCXLCSRC	FAILURE REASON CODE FROM THE LCS
16	(10)	A-ADDR	4	SCXLEN	TOTAL RESPONSE LENGTH
20	(14)	SIGNED-FWORD	4	SCXPEOFF	PARAMETER ERROR OFFSET
24	(18)	CHARACTER	4	SCXCSCNM	CSC SUBSYSTEM THAT PROCESSED THE REQUEST
28	(1C)	LENGTH		SCXHLL	SCX HEADER LENGTH
<b>NOTE:</b> THE FIELDS IN THIS SECTION ARE CURRENTLY UNUSED BY MVS/CSC, BUT WILL REMAIN DEFINED FOR POSSIBLE FUTURE USE					
CONFIGURATION SUMMARY SECTION DIRECTORY					
68	(44)	SIGNED-FWORD	4	SCXXCFGN	NUMBER OF CONFIGURATION ELEMENTS PRESENT IN THIS REPLY
72	(48)	SIGNED-FWORD	4	SCXXCFGO	OFFSET TO CONFIGURATION SECTION FROM START OF REPLY, OR 0 IF REPLY DOESN'T CONTAIN ANY CONFIGURATION ELEMENTS
76	(4C)	SIGNED-FWORD	4	SCXXCFGL	LENGTH OF A CONFIGURATION ELEMENT, OR 0 IF REPLY DOESN'T CONTAIN ANY CONFIGURATION ELEMENTS
VOLUME INFORMATION SECTION DIRECTORY					
80	(50)	SIGNED-FWORD	4	SCXXVOLN	NUMBER OF VOLUME ELEMENTS PRESENT THIS REPLY
84	(54)	SIGNED-FWORD	4	SCXXVOLO	OFFSET TO VOLUME SECTION FROM START OF REPLY
88	(58)	SIGNED-FWORD	4	SCXXVOLL	LENGTH OF A VOLUME ELEMENT

TABLE A-1 SCSXREQM Macro (Continued)

Dec	Hex	Type	Len	Label	Description
DRIVE INFORMATION SECTION DIRECTORY					
92	(5C)	SIGNED-FWORD	4	SCXXDRVN	NUMBER OF DRIVE ELEMENTS PRESENT IN THIS REPLY
96	(60)	SIGNED-FWORD	4	SCXXDRVO	OFFSET TO DRIVE SECTION FROM START OF REPLY
100	(64)	SIGNED-FWORD	4	SCXXDRVL	LENGTH OF A DRIVE ELEMENTS
NOTE: THE FIELDS IN THIS SECTION ARE CURRENTLY UNUSED BY MVS/CSC, BUT WILL REMAIN DEFINED FOR POSSIBLE FUTURE USE					
MESSAGE TEXT SECTION DIRECTORY					
104	(68)	SIGNED-FWORD	4	SCXXMSGN	NUMBER OF MESSAGE ELEMENTS PRESENT IN THIS REPLY
108	(6C)	SIGNED-FWORD	4	SCXXMSGO	OFFSET TO MESSAGE SECTION, FROM START OF REPLY, OR 0 IF REPLY DOESN'T CONTAIN ANY MESSAGE ELEMENTS
112	(70)	SIGNED-FWORD	4	SCXXMSGL	LENGTH OF A MESSAGE ELEMENT
NOTE: THE FIELDS IN THIS SECTION ARE CURRENTLY UNUSED BY MVS/CSC, BUT WILL REMAIN DEFINED FOR POSSIBLE FUTURE USE					
QCAP INFORMATION SECTION DIRECTORY					
116	(74)	SIGNED-FWORD	4	SCXXCAPN	NUMBER OF CAP ELEMENTS PRESENT IN THIS REPLY
120	(78)	SIGNED-FWORD	4	SCXXCAPO	OFFSET TO CAP SECTION FROM START OF REPLY
124	(7C)	SIGNED-FWORD	4	SCXXCAPL	LENGTH OF A CAP ELEMENT
NOTE: THE FIELDS IN THIS SECTION ARE CURRENTLY UNUSED BY MVS/CSC, BUT WILL REMAIN DEFINED FOR POSSIBLE FUTURE USE					
QEJECT INFORMATION SECTION DIRECTORY					
128	(80)	SIGNED-FWORD	4	SCXXQJTN	NUMBER OF QEJECT ELEMENTS PRESENT THIS REPLY

TABLE A-1 SCSXREQM Macro (Continued)

Dec	Hex	Type	Len	Label	Description
132	(84)	SIGNED-FWORD	4	SCXXQJTO	OFFSET TO QEJECT SECTION FROM START OF REPLY
136	(88)	SIGNED-FWORD	4	SCXXQJTL	LENGTH OF A QEJECT ELEMENT
NOTE: THE FIELDS IN THIS SECTION ARE CURRENTLY UNUSED BY MVS/CSC, BUT WILL REMAIN DEFINED FOR POSSIBLE FUTURE USE					
STOP INFORMATION SECTION DIRECTORY					
140	(8C)	SIGNED-FWORD	4	SCXXSTPN	NUMBER OF STOP ELEMENTS PRESENT IN THIS REPLY
144	(90)	SIGNED-FWORD	4	SCXXSTPO	OFFSET TO STOP SECTION FROM START OF REPLY
148	(94)	SIGNED-FWORD	4	SCXXSTPL	LENGTH OF A STOP ELEMENT
NOTE: THE FIELDS IN THIS SECTION ARE CURRENTLY UNUSED BY MVS/CSC, BUT WILL REMAIN DEFINED FOR POSSIBLE FUTURE USE					
QDSN INFORMATION SECTION DIRECTORY					
152	(98)	SIGNED-FWORD	4	SCXXQDSN	NUMBER OF QDSN ELEMENTS PRESENT IN THIS REPLY
156	(9C)	SIGNED-FWORD	4	SCXXQDSO	OFFSET TO QDSN SECTION FROM START OF REPLY
160	(A0)	SIGNED-FWORD	4	SCXXQDSL	LENGTH OF A QDSN ELEMENT
CSC SUBSYSTEM NAME INFORMATION SECTION DIRECTORY					
152	(98)	SIGNED-FWORD	4	SCXXNAMN	NUMBER OF CSC SUBSYSTEM NAME ELEMENTS PRESENT IN THIS REPLY
156	(9C)	SIGNED-FWORD	4	SCXXNAMO	OFFSET TO CSC SUBSYSTEM NAME SECTION FROM START OF REPLY
160	(A0)	SIGNED-FWORD	4	SCXXNAML	LENGTH OF A CSC SUBSYSTEM NAME ELEMENT
11	(B)	AREA	8	SCXEND	LABEL FOR ADDRESSING NEXT AREA.
15	(F)	LENGTH		SCXL	SCX LENGTH

TABLE A-1 SCSXREQM Macro (Continued)

Dec	Hex	Type	Len	Label	Description
VOLUME INFORMATION ELEMENT					
THIS ELEMENT SUPPLIES INFORMATION ABOUT A SINGLE VOLUME AND IS REPEATED FOR EACH VOLUME ASSOCIATED WITH A REQUEST. THIS ELEMENT MAY APPEAR IN REPLIES TO THE FOLLOWING REQUESTS:					
QDRLIST - RETURN DRIVE INFORMATION, ORDERED BY PREFERENCE					
QVOLUME - RETURN VOLUME INFORMATION					
SELSCR - SELECT A SCRATCH VOLUME					
EJECT - EJECT VOLUMES					
0	(0)	STRUCTURE		SCXSVOL	
0	(0)	CHARACTER	3	SCXVID	SECTION IDENTIFIER
3	(3)	BITSTRING	1	SCXVSTA	VOLUME STATUS:
		1... .... X'80'		SCXVILB	VOLUME IS IN LIBRARY
		.1. .... X'40'		SCXVOHST	RESERVED FOR FUTURE USE
		..1. .... X'20'		SCXVSCR	RESERVED FOR FUTURE USE
		...1 .... X'10'		SCXVMAL	RESERVED FOR FUTURE USE
		.... 1... X'08'		SCXVDSC	RESERVED FOR FUTURE USE
		.... .1.. X'04'		SCXVMNT	VOLUME IS MOUNTED ON A DRIVE
		.... ..1. X'02'		SCXVERR	RESERVED FOR FUTURE USE
		.... ...1 X'01'		RESERVED	RESERVED FOR FUTURE USE
4	(4)	CHARACTER	6	SCXVSER	VOLUME SERIAL
10	(A)	A-ADDR	1	SCXVLC	VOLUME LOCATION CODE:
0	(00)	CONST		SCXVUNK	LOCATION DATA UNAVAILABLE
1	(01)	CONST		SCXVCEL	LOCATION DATA IS A CELL ADDRESS
2	(02)	CONST		SCXVDRV	LOCATION DATA IS A DRIVE ADDRESS
11	(B)	AREA	5	SCXVLOC	VOLUME LOCATION DATA:
11	(B)	HEXSTRING	1	SCXVACS	ACS ADDRESS
12	(C)	HEXSTRING	1	SCXVLSM	LSM ADDRESS
13	(D)	HEXSTRING	3	SCXVPNL	PANEL, ROW, COL
13	(D)	HEXSTRING	2	SCXVDRIV	DRIVE ADDRESS (0CUU)
15	(F)	HEXSTRING	1	SCXVFILL	RESERVED

TABLE A-1 SCSXREQM Macro (Continued)

Dec	Hex	Type	Len	Label	Description
16	(10)	CHARACTER	8	SCXVMED	TYPE OF MEDIA. POSSIBLE VALUES: 'LONGItud', 'STANDARD', 'ECART', 'ZCART', 'DD3A', 'DD3B', 'DD3C', 'DD3D', 'STK1U', 'STK1R', 'VIRTUAL'
24	(18)	CHARACTER	8	SCXVTSNN	VTSS NAME
32	(20)		8		RESERVED FOR FUTURE USE
40	(28)	LENGTH		SCXSVOLL	LENGTH OF SCXSVOL
DRIVE INFORMATION ELEMENT					
THIS ELEMENT SUPPLIES INFORMATION ABOUT A SINGLE LIBRARY TAPE DRIVE AND IS REPEATED FOR EACH DRIVE ASSOCIATED WITH A REQUEST. THIS ELEMENT MAY APPEAR IN REPLIES TO THE FOLLOWING REQUEST:					
QDRIVES - RETURN DRIVE AND LSM INFORMATION					
0	(0)	STRUCTURE		SCXSDRV	
0	(0)	CHARACTER	3	SCXDID	SECTION IDENTIFIER
3	(3)	BITSTRING	1	SCXDSTA	LIBRARY STATUS
		...1 .... X'10'		SCXDMANU	RESERVED FOR FUTURE USE
		.... 1... X'08'		SCXDDISC	RESERVED FOR FUTURE USE
4	(4)	HEXSTRING	1	SCXQDEAC	ACS ADDRESS
5	(5)	HEXSTRING	1	SCXQDELS	LSM ADDRESS
6	(6)	HEXSTRING	2	SCXQDECU	DRIVE ADDRESS (0CUA)
8	(8)	CHARACTER	8	SCXQDRT	RECORDING TECHNIQUE OF DRIVE. POSSIBLE VALUES: '18TRACK', '36ATRACK', '36BTRACK', '36CTRACK', 'DD3', 'STK1R'
16	(10)	AREA	8	-RESERVED-	ALIGN
20	(20)	LENGTH		SCXSDRVL	LENGTH OF SCXSDRV
CSC SUBSYSTEM NAME INFORMATION ELEMENT					
THIS ELEMENT SUPPLIES THE NAME OF AN ACTIVE CSC. IT IS REPEATED FOR EACH ACTIVE CSC FOUND. THESE ELEMENTS WILL APPEAR IN THE SAME ORDER AS THE ACTIVE CSC SUBSYSTEMS APPEAR IN THE MVS SUBSYSTEM NAME TABLE. THIS ELEMENT MAY APPEAR IN REPLIES TO THE FOLLOWING REQUEST:					
QCSC *ALL - QUERY ALL ACTIVE CSC SUBSYSTEMS					
0	(0)	STRUCTURE		SCXSNAM	
0	(0)	CHARACTER	3	SCXNID	SECTION IDENTIFIER
3	(3)	BITSTRING	1		UNUSED

**TABLE A-1** SCSXREQM Macro (Continued)

<b>Dec</b>	<b>Hex</b>	<b>Type</b>	<b>Len</b>	<b>Label</b>	<b>Description</b>
4	(4)	CHARACTER	4	SCXNAME	CSC SUBSYSTEM NAME
8	(8)	AREA	8	-RESERVED-	ALIGN
32	(20)	LENGTH		SCXSNAMEL	LENGTH OF SCXSNAME





# Gathering Diagnostic Materials

---

---

## Overview

During problem resolution, Software Support may request that you provide specific diagnostic material. While printed format may be accepted, machine readable data (on magnetic tape) is preferred. For small amounts of data, Software Support may request that you FAX the data. Doing this may significantly reduce the time needed to resolve your problem.

---

## MVS Diagnostic Materials

The following MVS/CSC diagnostic materials may be requested by Software Support:

- Details of circumstances
- MVS SYSLOG
- SCSLOG data set
- SCSTRACE data set
- SYSxDUMP and SYS1.DUMPnn data sets
- Event Log Report (VM-based LCS)
- Event log data set (MVS-based and UNIX-based LCS)
- EREP records (software)
- MVS/CSC startup parameter file
- MVS/CSC startup procedure (cataloged procedure)
- MVSCP/IOCP definition or HCD

## Tape Format

If Software Support requests a tape containing your diagnostic materials, copy the requested files to tape using standard utility programs.

Include a description of the tape contents, including any information necessary for Software Support to retrieve the files from the tape (i.e., tape volume serial number and label attributes, number of files, file names and attributes, etc.).

## List of Abbreviations

---

abend	Abnormal end of task
ACS	Automated Cartridge System
APPC	Advanced-Program-to-Program Communications
CAP	Cartridge access port
CDS	Control data set
CLS	Common Library Services
CLSCOMM	CLS Communications
CLSCM	CLS Configuration Management
CLSLP	CLS logical port
CLSM	CLS Manager
CLSOC	CLS operator console
CMS	Conversational monitor system
CP	Control program
CPA	Control Path Adaptor
CSA	Common service area
CSC	Client System Component
CSSC	Customer Service Support Center
DASD	Direct access storage device
DFSMS	Data Facility Storage Management Subsystem
DMS/OS	DASD Management System/Operating System
EC	Engineering change
EDL	Eligible Device List
ESC	European Support Center
HCD	Hardware Configuration Definition
HSC	Host Software Component
IBM	International Business Machines Corporation
ICRC	Improved Cartridge Recording Capacity
ID	Identifier or identification

IJS	Intermediate Job Summary table
IML	Initial microcode load
I/O	Input/output
IOCP	I/O Configuration Program
IP	Internet Protocol
IPL	Initial program load
ISMF	Interactive Storage Management Facility
JCL	Job control language
JES	Job entry subsystem
JST	Job Summary Table
LAN	Local area network
LCU	Library Control Unit
LMU	Library Management Unit
LP	Logical port
LU	Logical unit
LSM	Library Storage Module
MB	Megabyte
MIM	Multi-image Manager
MVS	Multiple virtual storage
MVS/ESA	Multiple Virtual Storage/Enterprise Systems Architecture
MVS/SP	Multiple Virtual Storage/System Product
PCR	Product change request
PGMI	Programmatic interface
PIB	Product Information Bulletin
PN	Part number
PROP	Programmable operator facility
PTF	Program temporary fix
PUT	Program update tape
PVM	VM/Pass-Through Facility
RACF	Resource access control facility
RPC	Remote procedure call
SAF	System Authorization Facility
SER	Software Enhancement Request
SCP	System control program
SLK	Refers to the SCP
SMC	Storage Management Component
SMP/E	System Modification Program Extended
SMS	Storage Management Subsystem

SNA	Systems Network Architecture
SP	System Product
SSR	System Support Representative
Sysplex	<u>System complex</u>
TCP/IP	Transmission Control Protocol/Internet Protocol
TLMS	Tape library management system
TMI	Tape management interface
TMS	Tape management system
VLR	Volume location record
VM	Virtual machine
VOLSER	Volume serial number
VSM	Virtual Storage Manager
VTAM	Virtual Telecommunications Access Method
VTCS	Virtual Tape Control System
VTSS	Virtual Tape Storage Subsystem
WSC	World Wide Support Center
WTO	Write-to-operator
WTOR	Write-to-operator with reply



# Glossary

---

This glossary defines terms and abbreviations used in this publication.

---

## A

**Abnormal end of task (abend)** A software or hardware problem that terminates a computer processing task.

**ACS-id** A method used in the LIBGEN process to identify ACSs by using hexadecimal digits, 00 to nn.

**ACS** *See* Automated Cartridge System.

**ACS library** A library is composed of one or more Automated Cartridge Systems (ACSs), attached cartridge drives, and cartridges residing in the ACSs.

**address** Coded representation of hardware id, or the destination or origination of data.

**allocation** The assignment of resources to a specific task.

**asynchronous transmission** Character-oriented data transmission (as distinct from IBM's block-mode transmission).

**Automated Cartridge System (ACS)** A fully-automated, cartridge storage and retrieval library subsystem consisting of one or more Library Storage Modules (LSMs) connected by pass-thru ports.

**Automated Cartridge System Library Software (ACSL)** The library control software, which runs in the UNIX®-based Library Control System.

**automatic mode** A relationship between an LSM and all attached hosts. LSMs operating in automatic mode handle cartridges without operator intervention. This is the normal operating mode of an LSM that has been modified online. The opposite situation is "manual mode." *See* manual mode.

---

## B

**bar code** A code consisting of a series of bars of varying widths. This code appears on the external label attached to the spine of a cartridge and is equivalent to the volume serial number (volser). This code is read by the robot's machine vision system.

**BISYNC** Binary Synchronous Communications. An early low-level protocol developed by IBM and used to transmit data on a synchronous communications link. It is a form of data transmission in which synchronization of characters is controlled by timing signals generated at the sending and receiving stations.

---

## C

**CAPid** A CAPid uniquely defines the location of a CAP by the LSM on which it resides. A CAPid is of the form "AAL" where "AA" is the acs-id and "L" is the LSM number.

**cartridge** The plastic housing around the tape. It is approximately 4 inches (100 mm) by 5 inches (125 mm) by 1 inch (25 mm). The tape is threaded automatically when loaded in a transport. A plastic leader block is attached to the tape for automatic threading. The spine of the cartridge contains an OCR/Bar Code label listing the VOLSER (tape volume identifier).

**Cartridge Access Port (CAP)** An assembly that allows several cartridges to be inserted into or ejected from an LSM without human entry into the LSM.

**cartridge drive (CD)** A hardware device containing two or four cartridge transports and associated power and pneumatic supplies.

**cartridge tape I/O driver** Operating system software that issues commands (for example, read, write, and rewind) to cartridge subsystems. It is the software focal point for attaching a particular type of control unit. (An example is Oracle's StorageTek CARTLIB product.)

**cartridge transport** *See* transport.

**CDS** *See* Control Data Set.

**cell** A receptacle in the LSM in which a single cartridge is stored.

**CGI** Common Gateway Interface

**channel** A device that connects the host and main storage with the input and output devices' control units. A full-duplex channel has two paths (that is, 2 wires, or one wire with signals at two frequencies). A half-duplex channel requires that one port receives while the other transmits.

**channel-to-channel (CTC)** Refers to the communication (transfer of data) between programs on opposite sides of a channel-to-channel adapter.(I)

**client** The ultimate user of the ACS services as provided by the Library Control System.

**client link** The communications link between the LCS and a client.



**client-server** A model of interaction in a distributed system in which a program at one site serves a request to a program at another site and awaits a response. The requesting program is called a client; the program satisfying the request is called a server.

**client system** The system to which the LCS provides an interface to a StorageTek Automated Cartridge System.

**Client System Component (CSC)** Software that provides an interface between the Client Computing System's operating system and the StorageTek Library Control System (LCS).

**coaxial cable** A transmission medium used in data transmissions for networks using synchronous communications, as opposed to twisted-pair, the primary medium for asynchronous RS-232 communications.

**complex** A system composed of other systems, specifically the ACS server system and the client system.

**connected mode** A relationship between a host and an ACS. In this mode, the host and an ACS are capable of communicating (in the sense that at least one station to this ACS is online).

**connection number** The unique identifier on the server for a communications path. The number is assigned by TCP/IP to identify the unique connection between the server node and a specific port on the server, and the client node and a specific port on the client. The connection number exists only as long as the connection exists.

**console** The primary I/O device to control a session on a system.

**control data set (CDS)** The data set used by the host software to control the functions of the automated library. Also called a library database.

**Control Path Adaptor (CPA)** A Bus-Tech, Inc. hardware device that allows communications between a host processor's block multiplexer channel and a local area network.

**Control Unit (CU)** A microprocessor-based unit situated locally between a channel and an I/O device. It translates channel commands into device commands and sends device status to the channel.

**CTC** Channel-to-channel.

**Data Path Adapter** A hardware device which translates from a client computing system's data protocol to the data protocol of the StorageTek Control Unit or IMU. An example is DEC's TC44-AA/BA STI-to-4400 ACS Interconnect.

---

## D

**data set** A set of records treated as a unit.

**data sharing** The ability of concurrent subsystems or application programs to directly access and change the same data while maintaining data integrity.(I)

**device number** A four-digit hexadecimal number that uniquely identifies a device attached to a processor.

**device preferencing** The process of preferring one 36-track transport type over another 36-track transport type.

**device separation** *See* drive exclusion.

**direct access storage device (DASD)** IBM's term for a disk drive storage device.

**directed allocation** *See* drive prioritization.

**disconnected mode** A relationship between a host and an ACS. In this mode, the host and the ACS are not capable of communicating (there are no online stations to this ACS).

**dotted-decimal notation** The syntactic representation of a 32-bit integer that consists of four 8-bit numbers written in base ten with periods (dots) separating them. In TCP/IP descriptions, dotted-decimal notation is used for Internet addresses.

**drive exclusion** (previously referred to as *device separation*) refers to the Storage Management Component (SMC) function of excluding drives for an allocation request based on SMC exclusion criteria.

**drive panel** An LSM wall containing tape transports. The drive panel for a T9840 transport can contain either 10 or 20 transports. The drive panel for a non-T9840 transport can contain a maximum of 4 transports.

**drive prioritization** (previously referred to as *directed allocation*) refers to the Storage Management Component (SMC) function of influencing selection of a particular drive based on allocation criteria, including volume location.

**Dual LMU** A hardware/microcode feature that provides a redundant LMU capability.

**dump** A printed representation of the contents of main storage at time *t*. This representation is used for debugging purposes.

**dynamic server switching** The capability of switching server processors when a system failure occurs on the active server.

---

## E

**ECART** Enhanced Capacity Cartridge.

**Enhanced Capacity Cartridge** A cartridge that has a length of 1100 feet and can be used only on 36-track transports (i.e., 4490, 9490, and 9490EE).

**Enterprise Systems Connection (ESCON)** A set of products and services that provides a dynamically connected environment using optical cables as a transmission medium.(I)

**error codes (EC)** Numeric codes displayed by messages indicating the type of problem that caused an error.

**error recovery procedures (ERP)** Procedures designed to help isolate and, where possible, to recover from errors in equipment.

**ESCON** Enterprise Systems Connection.

**Ethernet** One LAN architecture using a bus topology that allows a variety of computers to be connected to a common shielded coaxial spine. The Ethernet architecture is similar to the IEEE 802.3 standard.

---

## F

**file** A set of related records treated as a unit.

**File Transfer Protocol (FTP)** A TCP/IP command that provides a way to transfer files between machines connected through TCP/IP.

**foreign socket** One of two end-points in a TCP/IP connection-oriented protocol. Specifies the address of a foreign host that can connect to the server.

---

## G

**GB** 1,073,741,834 bytes of storage

---

## H

**handshake** A flow-of-control signal sent by one process to another.

**helical cartridge** A high capacity, helical scan cartridge that can hold up to 50GB of uncompressed data. This cartridge can be used only on RedWood (SD-3) transports.

**host computer** A computer that controls a network of computers.

**Host Software Component (HSC)** Software running on the Library Control System processor that controls the functions of the ACS.

**Host Software Component utilities** Utilities provided by the VM/HSC that can be executed from the HSCUTIL virtual machine. *See* client-initiated utilities.

**HSC** *See* Host Software Component.

**HTTP** Hypertext Transfer Protocol.

---

## I

**IEEE 802.3** A standard produced by the IEEE and accepted worldwide for local area networks using CSMA/CD (Carrier Sense Multiple Access with Collision Detection).

**ICRC** Improved Cartridge Recording Capacity. A compression and compaction feature that increases the amount of data that can be stored on a 1/2-inch cartridge.

**initial program load (IPL)** A process that activates a machine reset.

**Interactive Storage Management Facility** A series of applications for defining DFSMS/MVS storage groups and classes.

**Internet** A collection of networks using TCP/IP that functions as a virtual network.

**Internet address** The numbering system used to specify a network or host on that network for TCP/IP communications. Standard Internet address notation is dotted-decimal format.

**Internet Protocol (IP)** Formal description of messages and rules two networks use to exchange messages.

**ISMF** Interactive Storage Management Facility.

---

## J

**job control language (JCL)** A problem oriented language designed to describe a job's processing requirements to an operating system.

**JES** Job entry subsystem.(I)

**JES2** An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In an installation with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing. *See also* JES3.(I)

**JES3** An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for execution, processes their output, and purges them from the system. In complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them via a common job queue. *See also* JES2.(I)

---

## L

**LAN** *See* local area network.

**LCS** *See* Library Control System.

**LCS processor console** The Library Control System processor console is used to control the VM operating system (for the VM-based LCS).

**LCU** *See* Library Control Unit.

**LIBGEN** The process of defining the configuration of a library to the VM/HSC.

**library** *See* TapePlex.

**library cartridge transport** *See* transport.

**library complex** A library complex consists of one HSC Control Data Set (CDS) and may contain up to 256 Automatic Cartridge Systems (ACSs), each of which may contain up to 24 Library Storage Modules (LSMs).

**library control component** Software that controls the mounting and dismounting of cartridges in an ACS.

**library control platform** The hardware and software that provides the proper environment for the Library Control System.

**library control processor** Properly configured computer hardware that supports the operation of the Library Control System.

**Library Control Software** A library control component, the client system interface, and library utilities.

**Library Control System (LCS)** The library control platform and the Library Control Software.

**Library Control Unit (LCU)** The portion of an LSM that controls the movements of the robot.

**library database** A file or data set containing information about the location and status of the removable media volumes, such as cell location, scratch status. Also called a control data set (CDS).

**library drive** A cartridge drive in the ACS, as distinct from a stand-alone cartridge drive.

**Library Management Unit (LMU)** A hardware and software product that coordinates the activities of one or more LSMs/LCUs.

**library mode** The operation of a 4480 Cartridge Subsystem as part of a 4400 Automated Cartridge System, as opposed to manual mode, in which the operator inserts cartridges into the transports. *See* manual mode.

**Library Storage Module (LSM)** The standard LSM (4410) a twelve-sided structure with storage space for up to around 6000 cartridges. It also contains a free-standing, vision-assisted robot that moves the cartridges between their storage cells and attached transports. *See* also PowderHorn, SL3000, SL8500, and WolfCreek.

**LMU** *See* Library Management Unit.

**local area network (LAN)** A network in a small (local) geographic area.

**local port** The designation of a given application or process among many that are available for a TCP/IP-capable host processor.

**local socket** The address combination of a TCP/IP-capable host's network address and a specific port for an application process.

**logical port (LP)** CLS software that interfaces with the client system. The CLSLP is one of the software components used to pass data between the client system and the VM/HSC.

**LP** *See* logical port.

**LSM** *See* Library Storage Module.

**LSM-id** An LSM-id is composed of the ACS-id joined to (concatenated with) the LSM number.

**LSM number** A method used to identify an LSM. An LSM number is the result of defining the SLIACS macro LSM parameter during a LIBGEN. The first LSM listed in this parameter acquires the LSM number of 00 (hexadecimal) the second LSM listed acquires a number of 01, and so forth, until all LSMs are identified (up to a maximum of 24 or hexadecimal 17).

---

## M

**manual mode** Operation of a cartridge drive apart from an ACS. *See* library mode.

**master LMU** The LMU currently controlling the functional work of the ACS in a dual LMU configuration.

**mixed configuration** A configuration that contains different types of cartridge drives in both manual and library modes.

**modem** A device that enables digital data to be transmitted over an analog transmission facility.

**multi-client** The environment where more than one (homogenous or heterogeneous) client system is connected to one LCS.

**MVS system console** The MVS/CSC provides an operator interface through the MVS system console.

---

## O

**OCR label** Optical character recognition label. An external label attached to the spine of a cartridge that is both human and machine readable.

**operator console** In this publication, the operator console refers to the MVS client system console.

**operating system (OS)** Software that controls the execution of programs that facilitate overall system operation.

---

## P

**Pass-thru Port (PTP)** A mechanism that allows a cartridge to be passed from one LSM to another in a multiple LSM ACS.

**physical port** The communications hardware required to support a server/client link.

**physical volume** A physically bound unit of data file media. *See* cartridge.

**pool** A collection of tape cartridges having one or more similar features or attributes, such as a pool of scratch tapes.

**PowderHorn (9310)** The high-performance version of the standard LSM.

**pre-configured package** A storage server package including all hardware, software, and configuration parameter settings delivered by the vendor.

**product change request (PCR)** A request for enhancement to a product. Normally, this request comes from a client, but may come from Oracle.

**program temporary fix (PTF)** A software release designed to remedy one or a series of defects.

**program update tape (PUT)** One or more tapes containing updates to, or new versions of, the MVS/CSC system software.

**protocol** A formal description of message formats and the rules two or more machines must follow to exchange these messages.

---

## R

**Resource Access Control Facility (RACF)** An IBM-licensed program that provides access control by identifying and verifying the users to the system.

**recovery** Automatic or manual procedures to resolve problems in the server system.

**reel-id** Identifier of a specific tape volume. Equivalent to volume serial number (VOLSER).

**request** Term used to refer to commands issued to the 4400 ACS to perform a tape-related function.



---

## S

**scratch tape** A tape that is available to any user because it is not owned.

**scratch tape subpool** A defined subset of all scratch tapes. Subpools are composed of one or more ranges of volumes with similar physical characteristics (type of volume—reel or cartridge, reel size, length, physical location, and so on). Some installations may also subdivide their scratch pools by other characteristics such as label type.

**SD-3** The StorageTek helical cartridge transport. Also known as RedWood.

**server** A library control system such as HSC. In SMC a server is represented by a named SERVER path to a named TAPEPLEX. While the StorageTek HTTP server software component is required as the middle ware on the remote host, the server, as far as SMC is concerned is the library control system operating on the remote host.

**socket** A unique address on a network plus a node address plus the id of one specific application on a specific network. An abstraction used by TCP/IP.

**standard capacity cartridge** A cartridge that can be used on any longitudinal transport (i.e., 4480, 4490, 9490, or 9490EE).

**standby** The status of a station that has been varied online but is connected to the standby LMU of a dual LMU ACS.

**standby LMU** The redundant LMU in a dual LMU configuration that is ready to take over in case of a Master LMU failure or when the operator issues a SWITCH command.

**station** A hardware path between the host computer and an LMU over which the VM/HSC and LMU send control information.

**Storage Management Component (SMC)** Software interface between IBM's z/OS operating system and StorageTek real and virtual tape hardware. SMC performs the allocation processing, message handling, and SMS processing for the ELS solution.

**storage server** A set of hardware and software products designed to enable heterogeneous computer systems to use automated tape cartridge library services.

**switchover** The assumption of master LMU function by the standby LMU.

**synchronous** *See* BISYNC.

**synchronous LAN** Local area network built on synchronous communications.

**sysplex** A set of MVS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads.(I)

**Systems Network Architecture (SNA)** A description of the logical structure, formats, protocols, and operational sequences for transmitting information units through and controlling the configuration and operation of networks.

---

## T

**tape drive** A tape processing device consisting of up to four transports in a cabinet. A drive can refer to an individual transport.

**tape library management system (TLMS)** TLMS, as used in this publication, refers to any tape library management system, not to CA-1.

**TapePlex** (formerly “library”), a single StorageTek hardware configuration, normally represented by a single HSC Control Data Set (CDS). A TapePlex may contain multiple Automated Cartridge Systems (ACSs) and Virtual Tape Storage Subsystems (VTSSs).

**TCP/IP** Transmission Control Protocol/Internet Protocol.

**trace event type** Types of event traced through the system when tracing is enabled.

**trace file** A file that contains information useful for debugging the system.

**transaction** A specific set of input that triggers the execution of a specific process.

**Transmission Control Protocol (TCP)** An inter-network standard protocol that provides a full-duplex stream service.

**transport** An electro-mechanical device used to thread, position, and read or write from a tape.

---

## U

**UCB** Unit Control Block.

**userid** Sometimes referred to as the VM userid, the userid is the name that identifies a specific “virtual machine” user or client.

**utility** Program that performs a function ancillary to the chief function(s) of a computer system.

---

## V

**virtual machine (VM)** A functional simulation of a computer and its associated devices. Each virtual machine is controlled by a suitable operating system.

**virtual storage** A feature of the OS where main storage requirements are allocated by segments (or pages) as needed by programs, thus creating the apparent existence of unlimited or virtual storage.

**Virtual Storage Manager (VSM)** A storage solution that virtualizes volumes and transports in a VTSS buffer in order to improve media and transport use.

**Virtual Tape Control System (VTCS)** The primary host code for the Virtual Storage Manager (VSM) solution. This code operates in a separate address space, but communicates closely with HSC.

**Virtual Tape Storage Subsystem (VTSS)** The DASD buffer containing virtual volumes (VTVs) and virtual drives (VTDs). The VTSS is a StorageTek RAID 6 hardware device with microcode that enables transport emulation. The RAID device can read and write "tape" data from/to disk, and can read and write the data from/to a real tape drive (RTD).

**Virtual Telecommunications Access Method (VTAM)** IBM host-resident communications software that serves as a common interface for communications.

**VM** *See* virtual machine.

**VM/SP or VM/XA** A proprietary operating system of IBM corporation that consists mainly of two major components, CP and CMS.

**volume** A tape cartridge (data carrier) that is mounted or dismounted as a unit.

**volume serial number (VOLSER)** An identifier of a physical volume.

---

## W

**WolfCreek (9360)** The high-performance LSM with a smaller capacity than the standard LSM.

---

## Z

**ZCART** An extended-enhanced cartridge that uses a thinner media to provide twice the capacity of the enhanced capacity (ECART) cartridge. This cartridge has a length of 2200 feet and can be used only on TimberLine 9490EE 36-track transports.

---

## Numerics

**802.3** See IEEE 802.3.

**3270** IBM synchronous, block-mode, half-duplex terminals preferred for use with IBM 370 and related types of machine.

**3270 protocol** A telecommunications protocol that supports networks of 327x CRTs on IBM mainframes.

**3274** Terminal control unit used on the ACS for processor-to-LMU communications.

**3480** IBM's 18-track half-inch cartridge tape drive model.

**3490** IBM's 36-track half-inch cartridge tape drive model.

**3590** IBM's newest cartridge tape drive model that supports 128-track recording technique.

**4400 Automated Cartridge System (ACS)** A fully automated, cartridge-based, 18-track storage and retrieval library. A 4400 ACS consists of 1 to 256 LMUs with each LMU connected to from 1 to 24 LSMs.

**4410** The standard Library Storage Module (LSM).

**4411** Library Control Unit (LCU).

**4480** The StorageTek 18-track 1/2-inch cartridge transport.

**4480 Cartridge Subsystem** The StorageTek 4480 Cartridge Subsystem consists of a control unit (CU) plus cartridge drives (CDs).

**4490** The StorageTek 36-track long-tape cartridge transport with ESCON support. Also known as Silverton.

**4780** Same as a 4480, but is used for attachment to certain non-IBM computers.

**8380** The StorageTek DASD system.

**9310** The PowderHorn, a high-performance version of the standard LSM (4410)

**9360** The WolfCreek, a high-performance LSM with a smaller capacity than the standard LSM (4410).

**9490** The StorageTek 36-track cartridge transport. Also known as TimberLine.

**9490EE** The StorageTek 36-track cartridge transport. Also known as TimberLine EE.

**9740** A small, four-sided StorageTek library that supports large-style cartridge transports. This library can be configured to contain either 326 cartridges or 494 cartridges.

**SL3000** The StorageTek high performance library scalable from 200 to 3,000 cartridge slots.

**SL8500** The StorageTek high performance library scalable from 29 terabytes to 70 petabytes with up to 70,000 tape slots.

**T9840A** The StorageTek access-centric cartridge transport capable of reading and writing 9840A cartridges.

**T9840B** The StorageTek access-centric cartridge transport capable of reading and writing T9840B cartridges.

**T9840C** The StorageTek access-centric cartridge transport capable of reading and writing T9840C cartridges.

**T9840D** The StorageTek access-centric cartridge transport capable of reading and writing T9840D cartridges.

**T9940A** The StorageTek capacity-centric cartridge transport capable of reading and writing 60GB T9940A cartridges.

**T9940B** The StorageTek capacity-centric cartridge transport capable of reading and writing 200GB T9940B cartridges.

**T10000** The StorageTek high-capacity cartridge transport capable of reading and writing 120GB or 500GB T10000 cartridges.

**T10000B** The StorageTek high-capacity cartridge transport capable of reading and writing 240GB or 1TB T10000 cartridges.



# Index

---

## A

address restrictions for transports, 94  
automatic tape switching, 94

## B

balancing scratch cartridges, 91  
basic functions, MVS/CSC, 20

## C

CADYNAM/TLMS  
MVS/CSC, 31  
cartridge  
changing scratch status, 31  
exchange activity, reducing, 93  
naming restrictions, 93  
tape transports  
automatic tape switching, 94  
sharing, 94  
Common Library Services (CLS), 24  
commands  
specifying component identifier, 36  
operator response to CLS commands, 38  
communications methods  
overview, 27  
CONFIGV control statement, specifying values  
for, 75  
considerations, performance, 91  
control statements  
CONFIGV for Configuration Verification  
utility, 75  
LOGRPT for Event Log Report utility, 72  
conventions  
control statements, 17  
typographic, 13

## D

Data Facility Hierarchical Storage Manager  
(DFHSM), 25  
device preferencing, 30  
devices, mixed, 30  
drive exclusion, 30

## E

environment  
MVS/CSC, 20  
Event Log Report utility  
description, 71  
LOGRPT control statement, 72

## F

Fault Analyzer for z/OS, interaction with  
MVS/CSC, 26

## H

Host Software Component (HSC)  
commands  
specifying component identifier, 36  
description, 24  
operator response to HSC commands, 38

## I

interfaces  
communications, 35  
MVS operator console, 36

## L

Library Control System (LCS)  
association with MVS/CSC, 94  
communications interface to MVS/CSC, 35  
software products, 23  
UNIX-based, 24  
VM-based, 24  
Library Storage Module (LSM)  
balancing scratch cartridges, 91  
maintaining scratch cartridge quantities, 92  
LOGRPT control statement, 72

## M

maintaining quantities of scratch cartridges, 92  
media, mixed, 30  
mixed media and mixed devices, 30  
MVS operating system  
operator console interface, 36

MVS operator console interface, 36  
MVS/CSC  
  association with LCS, 94  
  automatic tape switching, 94  
  basic functions, 20  
  communications interface to LCS, 35  
  configurations, 21  
  device sharing, 94  
  operating environment, 20  
  programmatic interface, 39  
  system interfaces, 21  
  tape management systems, support of, 33  
  utility functions, 65

## N

naming restrictions for cartridges, 93

## O

operating environment, 20  
operational constraints, 93  
operator response to HSC and CLS commands, 38

## P

performance considerations, 91  
port  
  VTAM requirements, 35  
preferencing, devices, 30  
programmatic interface  
  authorization checking, 41  
  description, 39  
  multiple MVS/CSCs, 41, 42  
  QCSC, 46  
  QDRIVES, 48  
  QVOLUME, 50  
  READ, 52  
  reason codes, 45  
  return codes, 43  
  samples, 54  
  SCSXREQ macro format, 42  
  SCSXREQM macro mappings, 97  
  supported requests, 46

## Q

querying  
  drives, 48  
  MVS/CSC, 46  
  volume, 50

## R

reason codes, programmatic interface, 45  
redistributing scratch cartridges, 91  
return codes  
  programmatic interface, 43

## S

schedule contention, 92  
scratch cartridge  
  maintaining quantities of, 92  
  redistributing across LSMs, 91  
Scratch Conversion utility, 83  
scratch pool  
  defining during initialization, 32  
  maintaining quantities of scratch cartridges, 92  
  specifying warning thresholds, 92  
Scratch Update utility, 80  
SCUADMIN utility program  
  description, 68  
  return codes, 70  
SCUCONDB program, 83  
SL8500 library, 30  
software products  
  Automated Cartridge System Library Software (ACSL), 24  
  Common Library Services (CLS), 24  
  Host Software Component (HSC) for VM, 24  
  Library Control System (LCS), 23  
  tape management systems, 33  
Storage Management Component (SMC), 20, 30  
System Authorization Facility (SAF), 26

## T

tape management system (TMS), 25  
  changing scratch status of cartridges, 31  
  coexistence with MVS/CSC, 31, 33  
  interfaces, 31  
TCP/IP communications link, 35  
third party software  
  DFHSM, 25  
  interactions with MVS/CSC, 25  
  MIM, 25  
  SAF, 26  
  STAM, 25  
  tape management systems, 25

## U

UNIX-based Library Control System (LCS), 24  
user exits  
  PGMI authorization checking, 41  
  tape management systems, 33  
utilities  
  Configuration, 75  
  Event Log Report, 71  
  Scratch Conversion, 83  
  Scratch Update, 80  
  SCUADMIN, 68  
utility functions, 65



## **V**

VM-based Library Control System (LCS)  
  Common Library Services (CLS), 24  
  description, 24  
  Host Software Component (HSC), 24

## VTAM

  communications link, 35

## **W**

workload balancing, 92

