

# StorageTek Client System Component for MVS Environments

---

Configuration Guide

Version 7.0

**ORACLE**

Part Number: E38048\_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>9</b>
<b>1. Introduction</b>	<b>17</b>
Overview	17
MVS/CSC Operating Environment	18
MVS/CSC Basic Functions	18
MVS/CSC System Interfaces	19
MVS/CSC Configurations	19
Security Administration Considerations for Communication	21
StorageTek Library Control System (LCS) Software Products	21
Third-Party Software Interaction	23
Tape Management Systems	23
Multi-image Manager (MIM)	23
Data Facility Hierarchical Storage Manager (DFHSM)	23
System Authorization Facility (SAF)	24
MVS/CSC Interaction With Fault Analyzer for z/OS	24
Communication Methods	25
User Policy Definition for Mixed Media and Devices	28
Device Preferencing	28
DFSMS/MVS Storage Management Subsystem Support	28
<b>2. Configuring the MVS/CSC Environment</b>	<b>29</b>
Overview	29
MVS/CSC Configuration Checklist	30
Defining Device Addresses and Esoteric Names	31

Running MVS/CSC and MVS/HSC on the Same MVS System	32
Defining Tape Request Attributes	32
Allocating the TAPEREQ Definition Data Set	33
Allocating the Event-Log and Trace Data Sets	33
Setting Up Communications	34
Defining the MVS/CSC Startup Parameter File	35
Modifying the MVS/CSC Startup Procedure	36
Pre-Initializing and Starting MVS/CSC	38
Migration Considerations	38
<b>3. Defining MVS/CSC Startup Parameters</b>	<b>39</b>
Overview	39
MVS/CSC Startup Parameter File	40
Parameter File Syntax Conventions	41
Parameter Syntax	41
Sample Startup Parameter File	42
Startup Parameter Descriptions	43
Common Startup Parameters	44
Communications Startup Parameters	59
<b>4. Defining Mixed Media and Devices</b>	<b>73</b>
Overview	73
OPTION TITLE Control Statement	74
Syntax	74
Control Statement Name	74
Parameters	74
Example	74
<b>5. Configuring MVS/CSC Communications</b>	<b>75</b>
Overview	75
Configuring Communications with a Unix-Based LCS	76
Defining TCP/IP as the Communications Method	76
Configuring Communications with a VM-Based LCS	84
Defining TCP/IP as the Communications Method	84
Defining VTAM "3270 BISYNC" as the Communications Method	86

<b>6. Verifying MVS/CSC Configuration</b>	<b>91</b>
Overview	91
Running the Configuration Verification Utility	91
CONFIGV Control Statement	92
<b>7. Starting and Stopping the MVS/CSC</b>	<b>97</b>
Overview	97
Pre-Initializing the MVS/CSC	98
Starting the MVS/CSC	99
Stopping the MVS/CSC	100
Communications Considerations When Stopping the MVS/CSC	101
<b>A. Considerations When Setting Network Timeout Parameters</b>	<b>103</b>
Overview of MVS/CSC Network Transaction	104
Guidelines for Setting Network Timeout Parameters	105
Considerations When Setting ALOCTIME	106
Considerations When Setting a Low ALOCTIME Value	106
Considerations When Setting a High ALOCTIME Value	107
LCS Considerations	107
Considerations When Setting ALOCTIME and REQTIME	107
Considerations When Setting ALOCTIME and REQTIME in a Single-LCS Environment	108
<b>B. Gathering Diagnostic Materials</b>	<b>109</b>
Overview	109
MVS Diagnostic Materials	109
Tape Format	110
<b>C. Migration and Coexistence</b>	<b>111</b>
Migration	111
Migrating From MVS/CSC Release 4.0 or later to Release 7.0	111
Reverse Migration From MVS/CSC Release 7.0 to Release x.x	112
Coexistence	112
Compatibility With SMC	112
<b>D. List of Abbreviations</b>	<b>113</b>

**Glossary 117**

# Figures

---

FIGURE 1-1	MVS/CSC to LCS Configuration	20
FIGURE 1-2	Communications Using TCP/IP and SNA LU6.2 (Unix-Based LCS)	26
FIGURE 1-3	Communications Using TCP/IP and VTAM “3270 BISYNC” (VM-Based LCS)	27
FIGURE 2-1	Sample MVS/CSC Started Task Procedure	37
FIGURE 3-1	Sample Startup Parameter File for ACSLS Server with TCP/IP Communications	42
FIGURE 6-1	Sample JCL for Configuration Verification Utility	93
FIGURE 6-2	Configuration Verification Utility Sample Output-Input Control Statement	93
FIGURE 6-3	CONFIGV Sample Output - All Parameters Valid (VM-Based LCS with TCP/IP)	94
FIGURE 6-4	CONFIGV Sample Output - Some Invalid Parameters (VM-Based LCS with TCPIP)	95





# 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 Messages and Codes Guide*
- *MVS/CSC Operator's Guide*
- *MVS/CSC Syntax Quick Reference*
- *MVS/CSC System Programmer's Guide*

### 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.

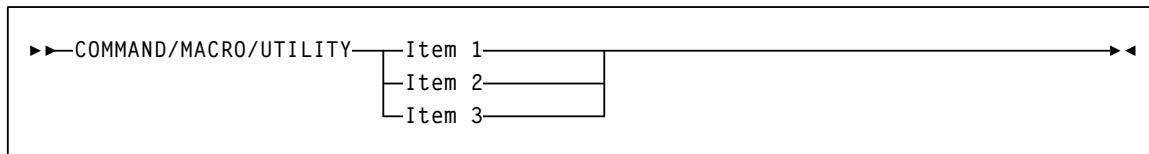
The use of mixed upper and lower case characters for commands, control statements, and parameters indicates that lower case letters may be omitted to form abbreviations. For example, you may simply enter POL when executing the POLicy command.

## Syntax Flow Diagrams

Syntax flow diagramming conventions include the following:

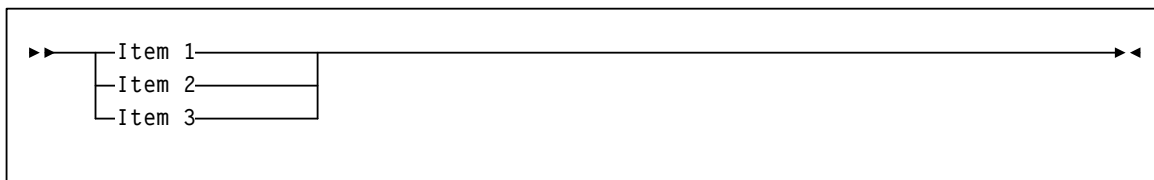
### Flow Lines

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.



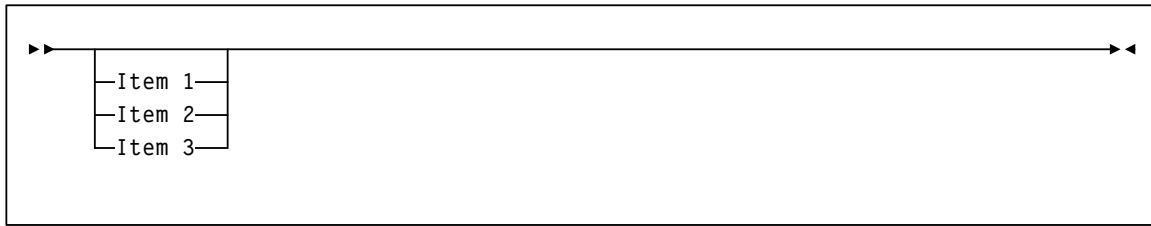
### Single Required Choice

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.



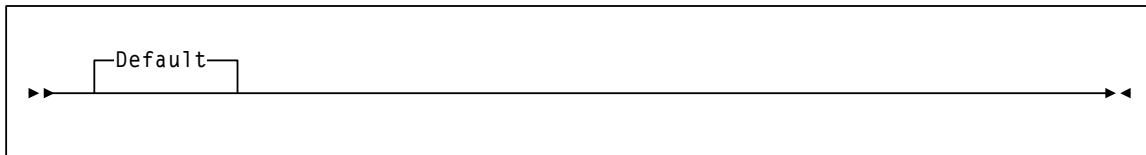
## Single Optional Choice

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

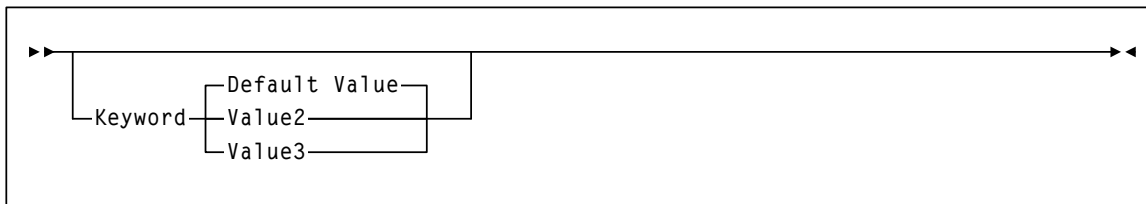


## Defaults

Default values and parameters appear above the baseline.

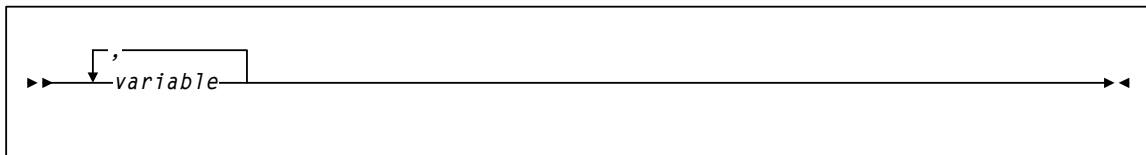


Some keyword parameters provide a choice of values in a stack. When the stack contains a default value, the keyword and the value choices are placed below the baseline to indicate that they are optional, and the default value appears above the keyword line.



## 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:

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

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

- The character positions of the incremental portion of both range elements must match.
- The non incremental characters of the first element must be identical to those of the second element.
- You cannot increment two portions of a range element. If 111AAA is the first element, you cannot specify 112AAB for the second element.
- The alphabetic portion of the VOLSER range is defined as being from character A to Z. To increment multi-character sequences, each character increments to Z. For instance, ACZ is part of the AAA-AMM range. Examples are:

<u>A00A0</u> - <u>A99A0</u>	increments VOLSERs A00A0 through A09A0, then A10A0 through A99A0.
9 <u>AA9A</u> -9 <u>ZZ9A</u>	increments VOLSERs 9AA9A through 9AZ9A, then 9BA9A through 9ZZ9A.
111 <u>AAA</u> -111 <u>ZZZ</u>	increments VOLSERs 111AAA through 111AAZ, then 111ABA through 111ZZZ
999 <u>AM8</u> -999 <u>CM8</u>	increments VOLSERs 999AM8 through 999AZ8, then 999BA8 through 999CM8
A3 <u>BZZ9</u> -A3 <u>CDE9</u>	increments VOLSERs A3BZZ9 through A3CAA9, then A3CAB9 through A3CDE9
<u>AAAAAA</u> - <u>AAACCC</u>	increments VOLSERs AAAAAA through AAAAAZ, then AAAABA through AAACCC
<u>CCNNN</u> - <u>DDNNN</u>	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 StorageTek LibraryStation. In an MVS-only environment, you must use StorageTek SMC and its HTTP server component 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.

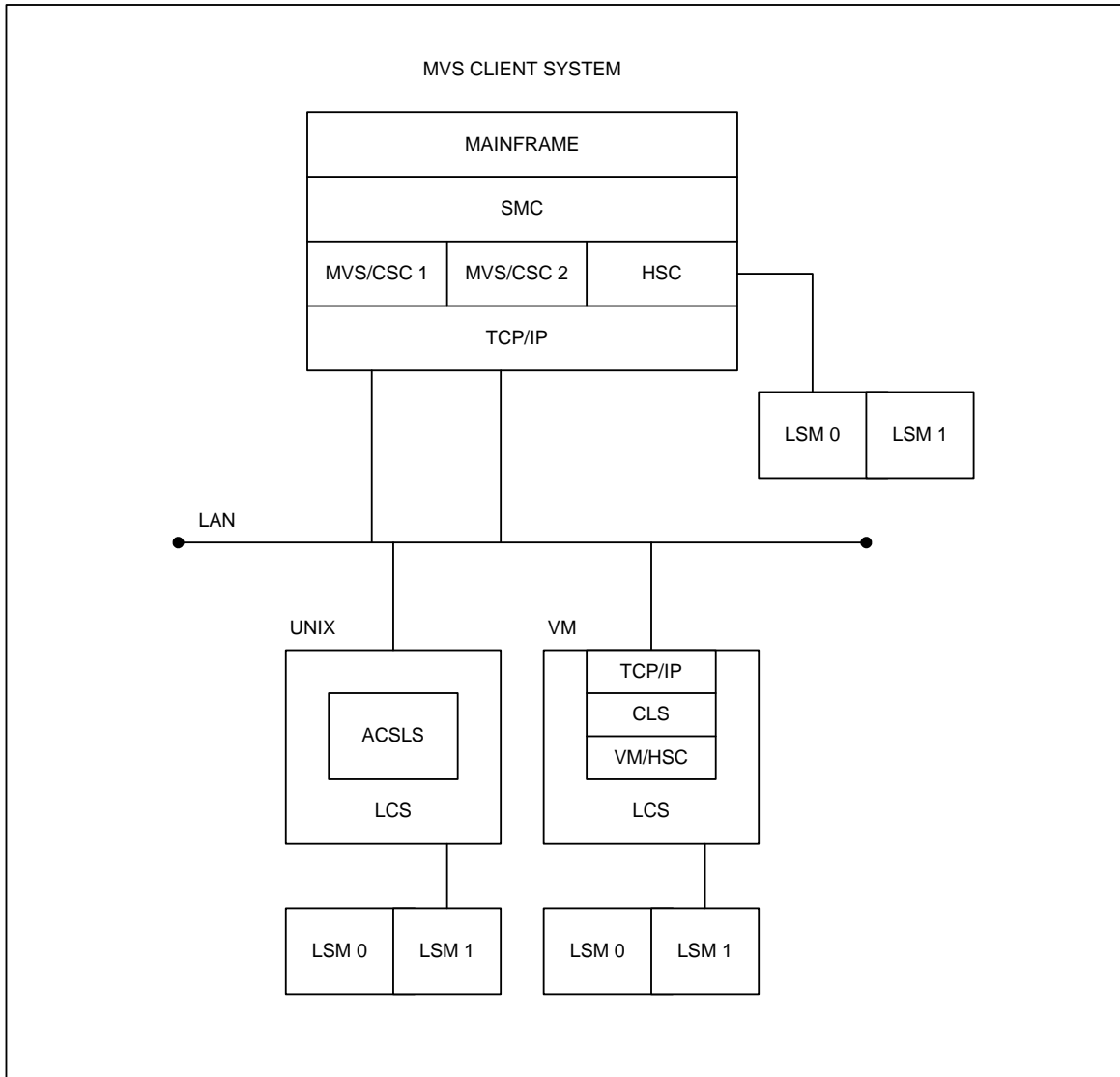


FIGURE 1-1 MVS/CSC to LCS Configuration

---

# 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 on 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 Oracle's 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. See [Appendix C, “Migration and Coexistence”](#) 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.

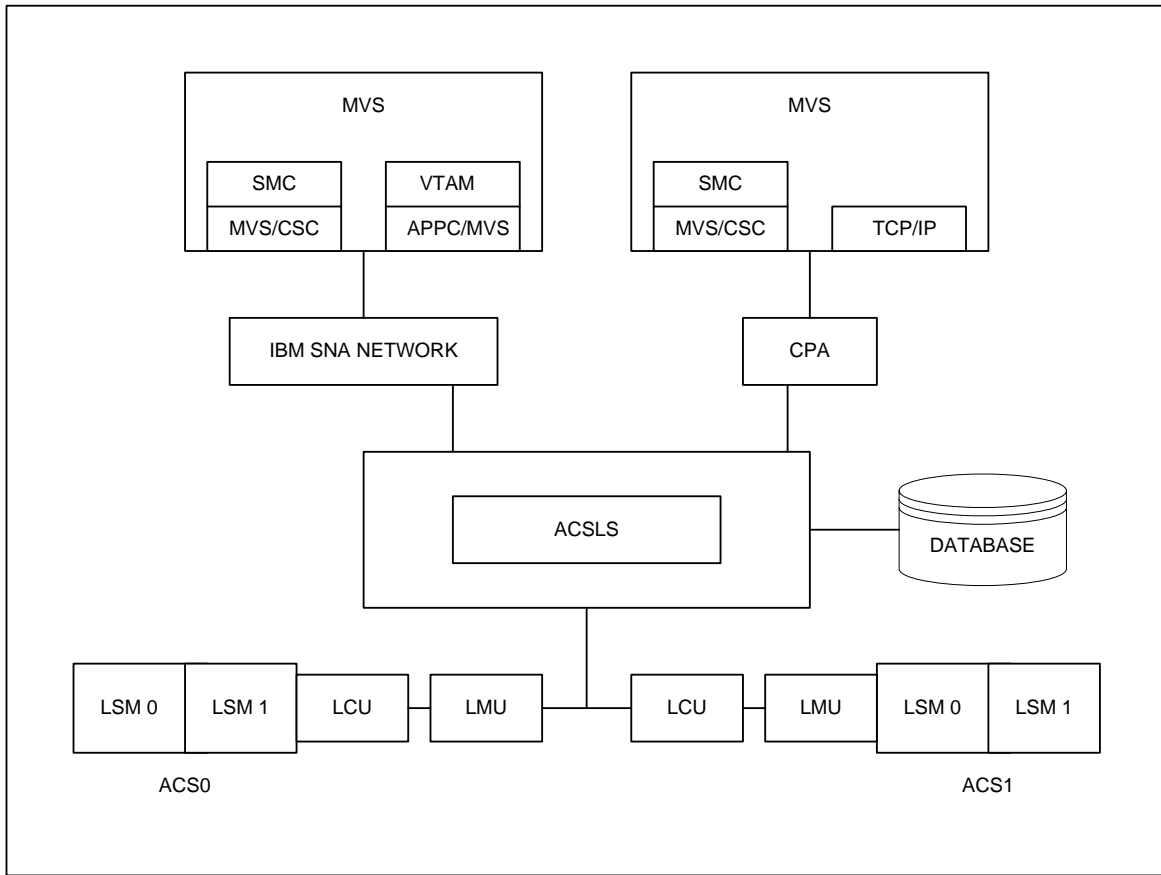
---

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.

---



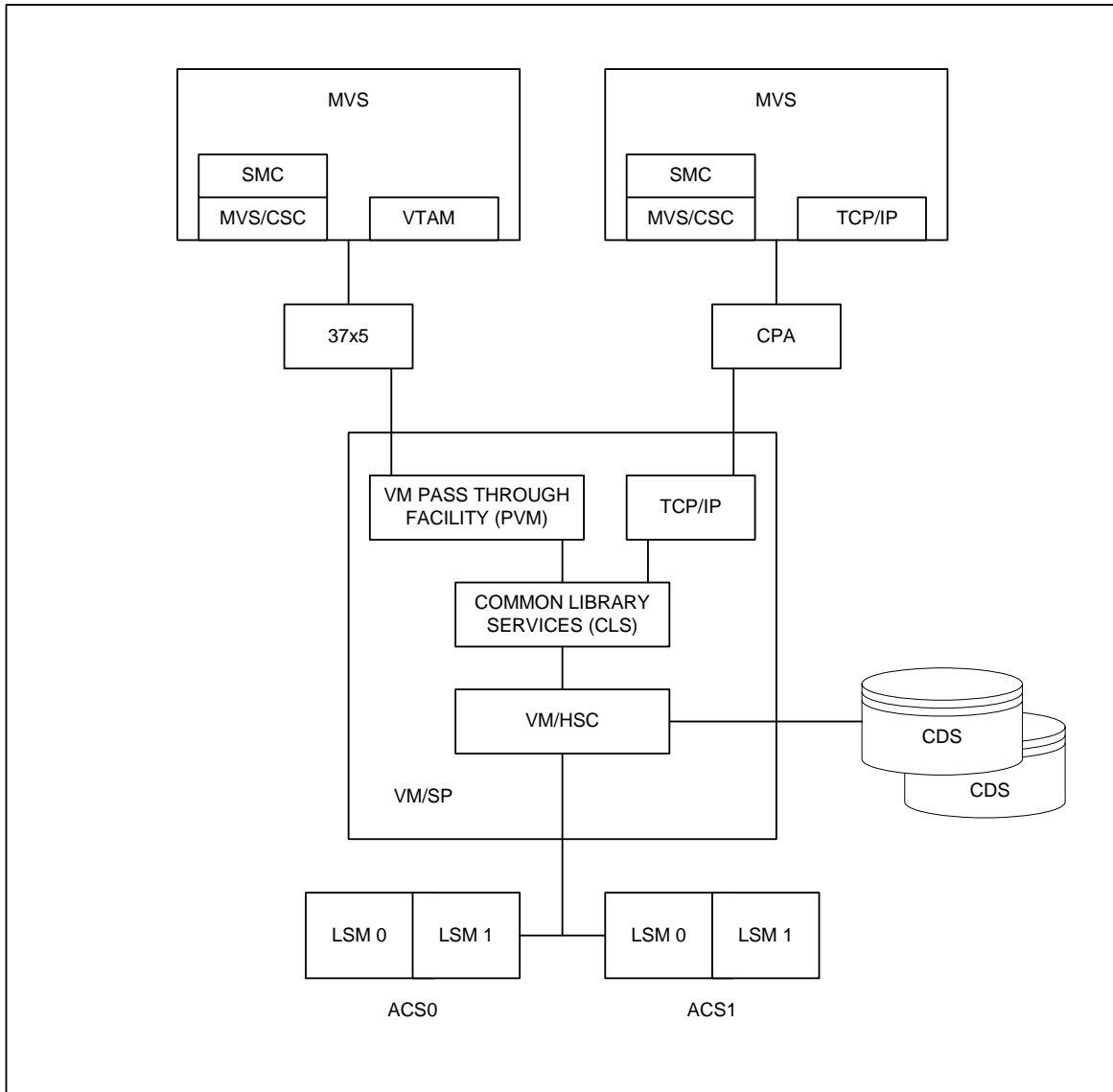
**FIGURE 1-2** Communications Using TCP/IP and SNA LU6.2 (Unix-Based LCS)

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 publication *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 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.

# Configuring the MVS/CSC Environment

---

---

## Overview

---

**Note** – MVS/CSC must be installed prior to performing MVS/CSC configuration tasks. Refer to the publication *Installing ELS* for MVS/CSC installation procedures.

---

This chapter describes the configuration tasks for the MVS/CSC. This chapter includes the following topics:

- MVS/CSC configuration checklist
- Defining device addresses and esoteric names
- Running MVS/CSC and HSC on the same MVS system
- Defining tape request attributes
- Allocating the TAPEREQ definition data set
- Allocating the event log and trace data sets
- Setting up communications
- Defining the MVS/CSC startup parameter file
- Modifying the MVS/CSC startup procedure
- Pre-initializing and starting MVS/CSC

---

**Note** – [“MVS/CSC Configuration Checklist” on page 30](#) includes a checklist detailing the MVS/CSC configuration tasks. Use this checklist to verify that you have completed all configuration tasks.

---

## MVS/CSC Configuration Checklist

Step	Description	Page(s)	Sample Member(s) Name (if applicable)	Notes
1	Define device addresses and esoteric names	31		
2	Review requirements for running MVS/CSC and MVS/HSC on the same MVS system	32		
3	Allocate event-log data set and trace data set (optional)	33		
4	Set up communications with a UNIX-based LCS (ACSL server)	34 & 76		
5	Set up communications with a VM-based LCS (CLS server)	34 & 84		
6	Define MVS/CSC startup parameter file	35 & 39	CSCPARM $x^*$	
7	Modify MVS/CSC startup procedure	36	CSCPROC	
8	Verify MVS/CSC configuration	91	JCLCFGV1 & JCLCFGV2	
9	Pre-initialize and start the MVS/CSC	38 & 97		

\*  $x$  is a number from 0 to 6.

---

# Defining Device Addresses and Esoteric Names

Use the Hardware Configuration Definition (HCD) facility to assign MVS unit addresses to the devices in your I/O Configuration. All library transports connected to the MVS host system must be defined in the MVS I/O configuration.

Because all cartridge transports might not be controlled by a particular library, you must define an esoteric name (or multiple esoteric names in a multiple ACS environment) for the pool of library transports controlled by the MVS/CSC, and an esoteric name for all other transport pools (i.e., nonlibrary transports). The LIBDev startup parameter and SMC UNITATTR statements are used to specify the esoteric names for each grouping. See [Chapter 3, “Defining MVS/CSC Startup Parameters”](#) for more information.

Separate esoteric names must be specified at MVS/CSC installation time for the following device groups:

- One for each ACS in the MVS/CSC-controlled libraries
- One for all accessible nonlibrary transports, if they exist
- One that spans all cartridge devices (recommended by IBM)

JCL for job streams does not need to be changed. You can continue to use the generic name or esoteric name that allows any 3480, 3490E, 3590, or helical-type transport to be allocated (manual 18-track, 36-track, T9840, T9940A, helical, or ACS-attached transports). The SMC, in conjunction with MVS or JES3 allocation, selects the correct transport within the specified generic or esoteric name.

For all server types, the MVS/CSC needs to know the addresses of the cartridge transports that it will use. The LIBUnit startup parameter is used to specify addresses for the real cartridge transports used by MVS/CSC. Virtual cartridge transports are configured dynamically if connected. Refer to the publication *Configuring HSC and VTCS* for more information about virtual cartridge transports.

The UNITMAP startup parameter is used to define the unit map or correlation between MVS device addresses and physical library location for the UNIX-based LCS. See [Chapter 3, “Defining MVS/CSC Startup Parameters”](#) for information about the UNITMAP startup parameter.

The CLS software running on the VM-based LCS processor maintains configuration information about the transports associated with each attached client system. The CLS defines a unique name for each transport and equates this name with the MVS device address used by the MVS/CSC. CLS maintains a table for this transport mapping within its Configuration Management subsystem. If the MVS/CSC device configuration changes, you must update the CLS mappings to reflect these changes. Refer to the *4400 Automated Cartridge System Common Library Services (CLS) User's Guide* for information about setting up a new version of the CLS configuration.

---

## Running MVS/CSC and MVS/HSC on the Same MVS System

---

**Note** – If you are running MVS/CSC and MVS/HSC on the same MVS system, you must include a STEPLIB DD statement in the started task procedure to identify the MVS/CSC load libraries. If you do not include a STEPLIB DD statement in the started task procedure to identify the MVS/CSC load libraries, unpredictable MVS/CSC operations may occur. See [“Modifying the MVS/CSC Startup Procedure” on page 36](#) for information about modifying the started task procedure.

---

One or more MVS/CSC subsystems can operate on the same MVS host system with the MVS/HSC. The MVS/HSC normally controls a local primary StorageTek library installation for the MVS system, while the MVS/CSC connects to a secondary (possibly remote) StorageTek library. This secondary library is typically used for vaulting or remote-backup operations.

Drive allocation, previously influenced by the MVS/CSC, is managed by the Storage Management Component (SMC). In determining the owning library subsystem, the SMC queries library subsystems in the order in which they are defined in the SMC TAPEPlex command. Refer to the *ELS Command, Control Statement, and Utility Reference* for more information.

Allocations may be directed to MVS/CSC library drives by specifying the TAPEREQ esoteric parameter containing only MVS/CSC drives, or by using a user exit to specify an owning library. Refer to the *ELS Legacy Interfaces Reference* for more information.

---

## Defining Tape Request Attributes

Use the SMC tape request (TAPEREQ) control statement to define the tape request characteristics for your data center. The SMC uses this information to ensure that the correct media type is used to satisfy the request and the cartridge is mounted on the appropriate device. Refer to the *ELS Command, Control Statement, and Utility Reference* for more information.



---

## Allocating the TAPEREQ Definition Data Set

The TAPEREQ definition data set is used to store the SMC TAPEREQ control statements that define the tape attributes of your data center. This data set can be sequential, or a member of a partitioned data set (PDS). You can use any valid record format to allocate the data set.

You can use the SMC TREQDEF command to load the TAPEREQ definition data set. Refer to the *ELS Command, Control Statement, and Utility Reference* for more information.

---

## Allocating the Event-Log and Trace Data Sets

The event-log data set is used to record events logged by the MVS/CSC Event-Log facility. It can also be used for trace output. The trace data set is specifically used to record trace output produced by the MVS/CSC Trace facility.

If you plan to use the event-log and trace data sets, you must specify the data sets in the startup procedure. You can set the LOG startup parameter to NO during the initial definition of the MVS/CSC startup parameters. You can then change the LOG startup parameter to YES or RESET to start logging to the event-log data set. Likewise, you can change the TRACDest startup parameter to FILE to start tracing to the trace data set.

Size definitions for the event-log data set will vary depending on your installation and the amount of activity on your system. Refer to the publication *Installing ELS* for information about DASD space and DCB parameter requirements. Refer to the *MVS/CSC System Programmer's Guide* for more information about the Event-Log and Trace facilities.

## Setting Up Communications

The MVS/CSC implementation requires a communications link between the MVS/CSC software and the LCS software. The following communications methods are available:

- Transmission Control Protocol/Internet Protocol (TCP/IP)
  - IBM TCP/IP
  - CA Unicenter TCPAccess Communications Server
  - CA Unicenter TCPAccess X.25 Server
- Virtual Telecommunications Access Method (VTAM “3270 BISYNC”)
- SNA LU 6.2 (APPC and VTAM are required)

You can set up communications between the MVS/CSC and the following LCSs:

- UNIX-based (ACSLs)
- VM-based (CLS)

See [Chapter 5, “Configuring MVS/CSC Communications”](#) for more information about setting up communications.

---

# Defining the MVS/CSC Startup Parameter File

After installing the MVS/CSC software using the SMP/E process, the startup configuration parameters related to the operation of the MVS/CSC must be specified. The MVS/CSC startup parameter file is a standard 80-byte flat file. It can reside as a member of SYS1.PARMLIB or another partitioned data set, or it can be maintained as a single sequential data set.

The values specified for each parameter are used each time the MVS/CSC is initialized. Once set, by specifying or accepting a default value, startup parameters remain in effect until changed. To change parameter settings, new values must be explicitly specified during a new initialization or certain parameters can be modified using the MVS/CSC ALTER operator command without re-initializing the MVS/CSC. Refer to the *MVS/CSC Operator's Guide* for information about the ALTER operator command.

The SCSPARM statement (included in the MVS/CSC started task procedure) identifies the startup parameter member that identifies the LCS as follows:

- CSCPARM0—Sample startup parameter file for VM-based (CLS) LCS using VTAM “3270 BISYNC”
- CSCPARM1—Sample startup parameter file for VM-based (CLS) dual LCSs using TCP/IP
- CSCPARM2—Sample startup parameter file for UNIX-based (ACSL) LCS using TCP/IP
- CSCPARM5—Sample startup parameter file for UNIX-based (ACSL) LCS using SNA LU 6.2

See [Chapter 3, “Defining MVS/CSC Startup Parameters”](#) for descriptions of the MVS/CSC startup parameters. The sample startup JCL is in member CSCPROC of the SAMPLIB library.

## Modifying the MVS/CSC Startup Procedure

Your system cataloged procedure library must contain a JCL startup procedure for MVS/CSC. [FIGURE 2-1 on page 37](#) shows a sample JCL startup procedure (member CSC0) to run MVS/CSC.

The JCL startup procedure is invoked after MVS is initialized but before any cartridge processing. The JCL startup procedure supplied with MVS/CSC must be modified and placed in a system cataloged procedure library.

---

### Note –

- The name of the member that contains the MVS/CSC JCL startup procedure must match the subsystem name defined in the IEFSSNyy member of SYS1.PARMLIB. For example, since CSC0 is the member name for the JCL startup procedure in [FIGURE 2-1](#), the subsystem name defined in the IEFSSNyy member must also be CSC0.
  - 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 (see [FIGURE 2-1](#)) is required. If you omit the STEPLIB DD statement in the started task procedure, unpredictable MVS/CSC operations may occur.
  - If you are using a CA Unicenter TCPAccess product for TCP/IP communications with a UNIX-based LCS, the TCPLINK data set must be included in the STEPLIB DD statement and must precede the DD for the SACLINK data set (see [FIGURE 2-1](#)). In addition, the CSLLINK data set is required for all TCP/IP communications products.
  - A region size of 4M is recommended. However, you might need to adjust the region size for your workload.
-

```

//CSCØ   PROC PROG=SCSBINIT,PRM=' '
//*
//* SAMPLE JCL START PROCEDURE
//* VALUES IN LOWER CASE MUST BE CHANGED TO CORRESPOND TO
//* LOCAL INSTALLATION.
//*
//* PRM=COLD TO REBUILD THE MVT
//* PRM=RESET TO RESET INIT/TERM SWITCHES IN SSCVT
//*
//CSCØ   EXEC PGM=&PROG,TIME=144Ø,DPRTY=(7,5),
// ACCT=your-acct-info,REGION=4ØØØK,PARM='&PRM'
//OUT    OUTPUT DEFAULT=YES,DEST=your-dest-node
//*
//* STEPLIB MUST POINT TO THE EXECUTABLE MODULES LIBRARIES.
//*
//STEPLIB DD DISP=SHR,DSN=your.SCCLINK
//        DD DISP=SHR,DSN=your.TCPLINK /* optional for CA TCP */
//        DD DISP=SHR,DSN=your.SACLINK
//        DD DISP=SHR,DSN=your.CSLLINK /* SAS/C CSL */
//*
//* SCSPARM MUST POINT TO MVS/CSC STARTUP PARAMETERS.
//*
//SCSPARM DD DISP=SHR,DSN=your.parmlib(csc-parm-member)
//*
//* SCSLOG IS OPTIONAL EVENT LOG AND MUST POINT TO A DSORG=PS
//* RECFM=VB LRECL=3Ø76 BLKSIZE=1ØØØØ DATA SET OF SUFFICIENT SIZE
//* TO HOLD EVENT RECORDS FOR TIME AND ACTIVITY.
//*
//SCSLOG DD DISP=SHR,DSN=your.log.file
//*
//* SCSTRACE IS OPTIONAL TRACE FILE AND MUST POINT TO A DSORG=PS
//* RECFM=VB LRECL=3Ø76 BLKSIZE=1ØØØØ DATA SET OF SUFFICIENT SIZE
//* TO HOLD EVENT RECORDS FOR TIME AND ACTIVITY.
//*
//SCSTRACE DD DISP=SHR,DSN=your.trace.file
//SYSABEND DD SYSOUT=*
//*

```

**FIGURE 2-1** Sample MVS/CSC Started Task Procedure



# Defining MVS/CSC Startup Parameters

---

---

## Overview

This chapter contains detailed descriptions of each MVS/CSC startup parameter, including an explanation of its function and the values associated with each parameter. It also describes the startup parameter file and syntax conventions and provides a sample startup parameter file.

The following startup parameters are provided with MVS/CSC:

- |            |            |            |
|------------|------------|------------|
| ■ ALOCTIME | ■ PORT     | ■ TRACDEST |
| ■ COMM     | ■ REQTIME  | ■ TRACE    |
| ■ COMPRFX  | ■ RETCOUNT | ■ UNITMAP  |
| ■ ENQNAME  | ■ RETTIME  | ■ USERDATA |
| ■ INTERNET | ■ SCRLABL  | ■ VAPLNAM  |
| ■ LIBDEV   | ■ SERVER   | ■          |
| ■ LIBUNIT  | ■ SYMDESTN |            |
| ■ LOG      | ■ SRVRLIST |            |
| ■ MSGCASE  | ■ TCPNAME  |            |

---

# MVS/CSC Startup Parameter File

MVS/CSC startup parameters are read during initialization of the MVS/CSC. The parameters reside in the MVS/CSC startup parameter file. The startup parameter file (1) can be a sequential data set or a member of a partitioned data set (PDS), and (2) must consist of 80-byte card-image records. The name of the startup parameter file is specified in the SCSPARM DD statement in the startup procedure.

Once set, by specifying or accepting a default value, startup parameters remain in effect until changed. To change parameter settings, new values must be explicitly specified during a new initialization or certain parameters can be modified using the MVS/CSC ALTER command without re-initializing the MVS/CSC. You can change the following parameters using the ALTER command:

- ALOCTIME
- MSGCASE
- TRACDEST

---

**Note** – Refer to the *MVS/CSC Operator's Guide* for information about the ALTER operator command.

---

Certain parameters are *persistent* (the last setting is retained). Once a value is specified, that value overrides any default value and remains in effect until it is explicitly re-specified by another initialization or is changed using the ALTER command. The following startup parameters are persistent:

- MSGCASE
- TRACDEST

---

**Note** – Refer to the *MVS/CSC Operator's Guide* for information about the ALTER operator command.

---

The following startup parameters must be specified at each initialization of the MVS/CSC:

- LIBDEV
- LIBUNIT
- SERVER



The following startup parameters must be specified depending on the environment as noted:

- INTERNET (required for TCP/IP communications)
- PORT (required for TCP/IP communications with a CLS server)
- SYMDESTN (required for SNA LU 6.2 communications with ACSLS servers)
- UNITMAP (required for ACSLS servers)
- VAPLNAM (required for VTAM “3270 BISYNC” communications)

## Parameter File Syntax Conventions

The standard syntax conventions for the startup parameter file are as follows:

- Each record must be an 80-byte card-image record.
- Using /\* and \*/ is the preferred method for enclosing comments in the job stream. The use of only an asterisk in any column to denote the beginning of a comment may not be supported in future MVS/CSC releases. Comments cannot be nested, and mixing the two comment styles (/\* and \*) is not allowed.
- Columns 73 thru 80 are ignored on all statements.
- A hyphen (-) in any column indicates a continuation character, which causes the next record (minus leading spaces) to be concatenated to the previous record, starting at the last nonblank column before the hyphen.
- A plus sign (+) in any column indicates a concatenation character, which causes the next record to be concatenated at the plus sign.
- Delimiters may be blanks or commas.
- Parameters must be specified in the following keyword format:

Keyword(value1,value2,etc.)

- Blanks between parameters and values are not allowed.
- If a string of characters is enclosed within quotation marks and exceeds the record length, you can add a concatenation character to the end of the string and continue the string on the next record. Both strings are treated as a single string of characters. Each string must be enclosed within quotation marks. For example:

```
USER('ABCD'+
      'EFG')
```

is equivalent to

```
USER('ABCDEFG')
```

- If a startup parameter is specified more than once, the last specification is used.

## Parameter Syntax

See [“Syntax Flow Diagrams” on page xiii](#).

## Sample Startup Parameter File

The following figure shows a sample startup parameter file for an UNIX-based LCS (ACSL) using TCP/IP communications. This file is in member CSCPARM2 of the SAMPLIB library.

```

*****
/* SAMPLE STARTUP PARAMETERS IDENTIFIED THROUGH THE MVS/CSC STARTUP      *
/* PROCEDURE SCSPARM DD STATEMENT.                                       *
/*                                                                           *
/* CSCPARM3- THIS SAMPLE REPRESENTS AN EXAMPLE OF AN ATTACHMENT          *
/* TO AN UNIX-BASED (ACSL) SERVER USING IBM TCP/IP COMMUNICATIONS.        *
/* THE DEVICE ADDRESSES ALSO REPRESENT 4-DIGIT DEVICE NUMBERS.          *
/* (THESE MUST BE CHANGED TO CORRESPOND TO THE LOCAL INSTALLATION.)     *
/*                                                                           *
/******
SERVER(ACSL)                               /* SERVER TYPE */
/*
/* (TWO ACS LIBRARY, MVS/CSC DEVICES IN ACS 0 AND 1, AND NONLIB DEVICES)
/*
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 */
        10B4,01:00:9:5,10B5,01:00:9:6, -   /* UNIT MAPPINGS */
        10B6,01:00:9:8,10B7,01:00:9:7)     /* UNIT MAPPINGS */
/*
COMM(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)                      /* SERVER IP ADDRESS */
TRACDEST(LOG)                                /* TRACE DESTINATION */
TRACE(AL)                                    /* TRACE ACTIVITY */
LOG(RESET)                                   /* LOGGING OPTION */
/******

```

FIGURE 3-1 Sample Startup Parameter File for ACSLS Server with TCP/IP Communications

Startup parameter files are also included for the following servers:

- UNIX-based LCS using SNA LU 6.2
- VM-based LCS using VTAM “3270 BISYNC”
- VM-based LCS using TCP/IP

See [“Defining the MVS/CSC Startup Parameter File” on page 35](#) for the member names of the sample startup parameter files.

---

## Startup Parameter Descriptions

The following sections provide detailed descriptions about each startup parameter, including an explanation of its function and associated values. The startup parameters are categorized as follows:

- Common Startup Parameters  
Startup parameters that are non-platform specific.
- Communications Startup Parameters  
Startup parameters specific to communications setup and network timeout activities.

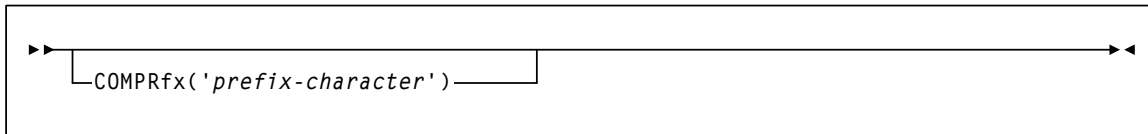
## Common Startup Parameters

This section lists the MVS/CSC startup parameters that are common across all platforms.

### COMPRFX Startup Parameter

This parameter specifies the command prefix character used to direct operator commands to the MVS/CSC. When defined, the command prefix character precedes console messages issued by MVS/CSC. This parameter is optional.

#### *Syntax*



#### *Parameter Descriptions*

This section describes the value that you can specify with the COMPRfx startup parameter.

##### *prefix-character*

Specifies the command prefix character used to direct operator commands to the MVS/CSC. The prefix character can contain only one character and must be enclosed in single quotes. Valid symbols are:

`¢ . < ( + ! & ! $ * ) ; ' - / , % _ > ? : # @ = I`

#### *Examples*

The following shows an example of the COMPRfx parameter used to specify ! as the command prefix character.

```
COMPR (' ! ')
```

The following example shows how to enter the MVS/CSC Display command using the defined command prefix character (!):

```
!DISPLAY ALL
```

If a command prefix character is not entered with the command, the MVS/CSC subsystem name must be entered with each command to direct the command to the MVS/CSC. The following example shows the Display command for the MVS/CSC subsystem CSC0, which was defined in member IEFSSNYY of the SAMPLIB library.

```
CSC0 DISPLAY ALL
```

---

**Note** – The MVS/CSC subsystem name can be used even when a COMPRfx value has been specified.

---



## LIBDEV Startup Parameter

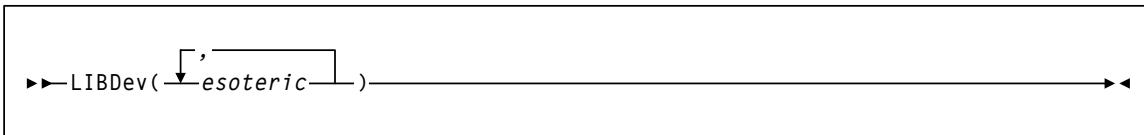
This parameter specifies the esoteric names for library transports for each ACS in the MVS/CSC-controlled library. This parameter is required and there are no defaults.

---

**Note** – The esoteric names must contain only those devices in each ACS.

---

### *Syntax*



### *Parameter Descriptions*

This section describes the values that you can specify with the LIBDev startup parameter.

#### *esoteric*

Specifies one or more esoteric names. Valid esoteric names can consist of from one- to eight-alphanumeric characters and must begin with an alphabetic character.

### *Example*

The following shows an example of the LIBDev parameter used to specify TACSO as the esoteric name for a library transport.

```
LIBD(TACSO)
```

The esoteric names must be specified in the order that the ACS is specified to the server. If no devices in an ACS are controlled by the MVS/CSC, the positional ACS esoteric name for the MVS/CSC must be left blank. For example, if the MVS/CSC has devices in ACS0 and ACS2 but none in ACS1, the LIBDev parameter would be specified as follows:

```
LIBD(TACSO, , TACS2)
```

## LIBUNIT Startup Parameter

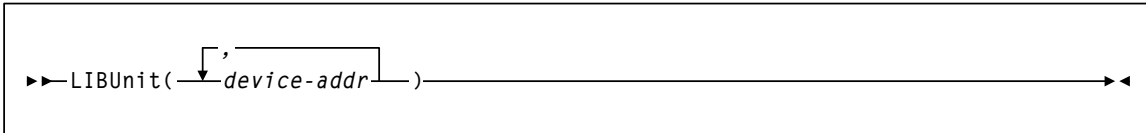
This parameter defines the MVS device addresses of the library transports controlled by an MVS/CSC subsystem. This parameter is required.

---

**Note** – All transports that reside in the MVS/CSC-controlled library and that are online to the MVS host system *must* be specified in this device list.

---

### Syntax



### Parameter Descriptions

This section describes the values that you can specify with the LIBUnit startup parameter.

#### *device-addr*

Specifies one or more MVS device addresses of library transports. Valid device addresses must be specified as four-hexadecimal character device addresses, because the MVS/CSC recognizes these formats as device addresses in MVS mount and dismount console messages.

---

**Note** – Ranges are not allowed.

---

### Example

The following shows an example of the LIBUnit parameter used to specify device addresses A001–A007 for library transports.

```
LIBU(A001,A002,A003,A004,A005,A006,A007)
```



## LOG Startup Parameter

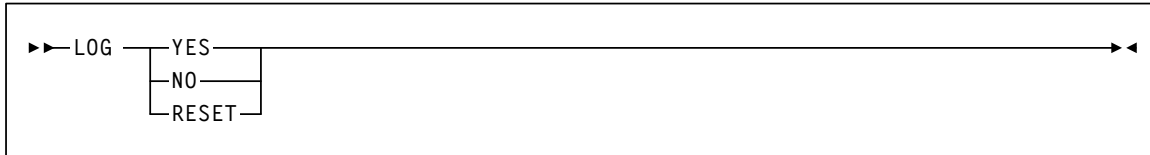
This parameter allows event logging to be turned on or off and defines how the event log is handled. This parameter is optional.

---

**Note** – The event log must be specified in the MVS/CSC startup procedure (SCSLOG DD) if event logging will be used.

---

### *Syntax*



### *Parameter Descriptions*

This section describes the values that you can specify with the LOG startup parameter.

#### **YES**

Specifies that logging of MVS/CSC activities begins at the current location in the event-log data set.

#### **NO**

Specifies that MVS/CSC activities are not logged in the event-log data set. This is the default value.

#### **RESET**

Specifies that activities are logged at the beginning of event-log data set.

### *Example*

The following shows an example of the LOG parameter used to specify that logging of MVS/CSC activities begins at the current location in the event-log data set.

```
LOG (YES)
```

## MSGCASE Startup Parameter

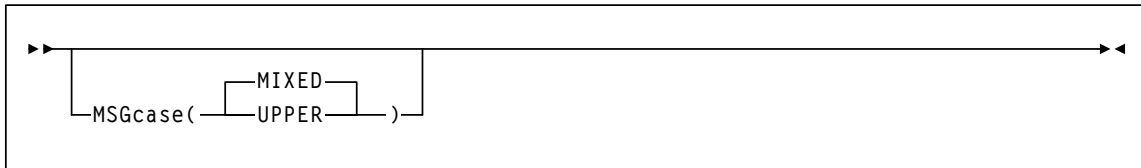
This parameter determines whether MVS/CSC message output appears in upper-case or mixed-case format. This parameter is optional.

---

**Note** – This parameter is persistent. Once a value is specified, that value overrides any default value and it remains in effect until it is explicitly re-specified by another initialization or ALTER command.

---

### *Syntax*



### *Parameter Descriptions*

This section describes the values that you can specify with the MSGcase startup parameter.

#### **UPPER**

Specifies that all message output appears in upper case only.

#### **MIXED**

Specifies that all message output appears in mixed case. This is the default value.

### *Example*

The following shows an example of the MSGcase parameter used to specify that all message output appears in upper-case format.

```
MSG (UPPER)
```

## SCRLABL Startup Parameter

This parameter specifies the magnetic label type of scratch mounts that the MVS/CSC automates. This parameter is optional.

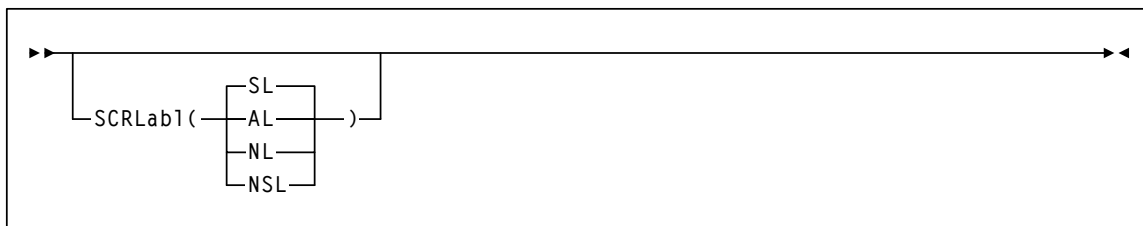
The MVS/CSC assumes that nonspecific requests with other than the SCRLABL label type are outside the library. If a nonspecific cartridge is requested with the label type specified, it is considered a scratch cartridge.

---

**Note** – Automated mounts of scratch cartridges having a label type different than the type specified on the SCRLABL label type can be performed using user exits. Refer to the *ELS Legacy Interfaces Reference* for more information about MVS/CSC user exits.

---

### Syntax



### Parameter Descriptions

This section describes the values that you can specify with the SCRLabl startup parameter.

#### SL

Specifies a standard label. This is the default value.

#### AL

Specifies an ANSI label.

#### NL

Specifies no label (nonlabeled).

#### NSL

Specifies a nonstandard label.

### Example

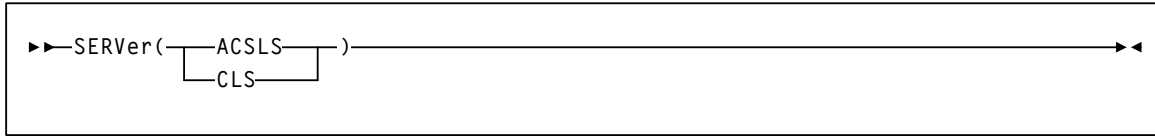
The following shows an example of the SCRLabl parameter used to specify the magnetic label type of a scratch volume that the MVS/CSC automates.

```
SCRL (AL)
```

## SERVER Startup Parameter

This parameter specifies the server protocol to be used during LCS communications. This parameter is required.

### *Syntax*



### *Parameter Descriptions*

This section describes the values that you can specify with the SERVER startup parameter.

#### **ACSL**

Specifies the connection to the ACSL server using ACSL RPC connection-less protocol.

#### **CLS**

Specifies the connection to the CLS server using CLS connection-mode protocol.

### *Example*

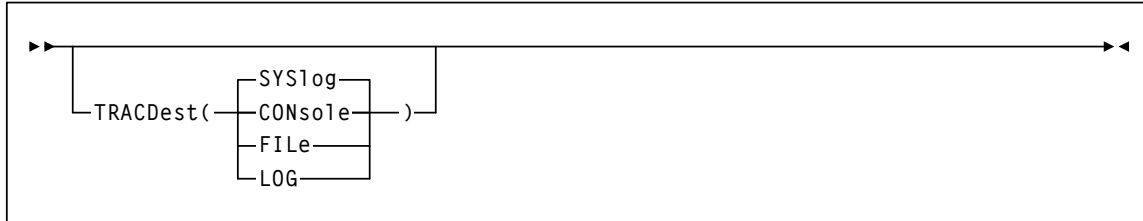
The following shows an example of the SERVER parameter used to specify ACSL as the server protocol used during LCS communications.

```
SERVER (ACSL)
```

## TRACDEST Startup Parameter

This parameter specifies the destinations for trace output. This parameter is optional.

### *Syntax*



### *Parameter Descriptions*

This section describes the values that you can specify with the TRACDest startup parameter.

#### **CONsole**

Specifies that the trace output be sent to the MVS operator console.

#### **FILE**

Specifies that the trace output be sent to the MVS/CSC trace data set. The trace data set must be specified in the startup procedure (using the SCSTRACE DD).

#### **LOG**

Specifies that the trace output be sent to the MVS/CSC event-log data set. The event-log data set must be specified in the startup procedure (using the SCSLOG DD) and event logging must be active.

#### **SYSlog**

Specifies that the trace output be sent to the MVS system log. This is the default value.

### *Example*

The following shows an example of the TRACDest parameter used to specify the MVS operator console as the destination for trace output.

```
TRACD (CON)
```

## TRACE Startup Parameter

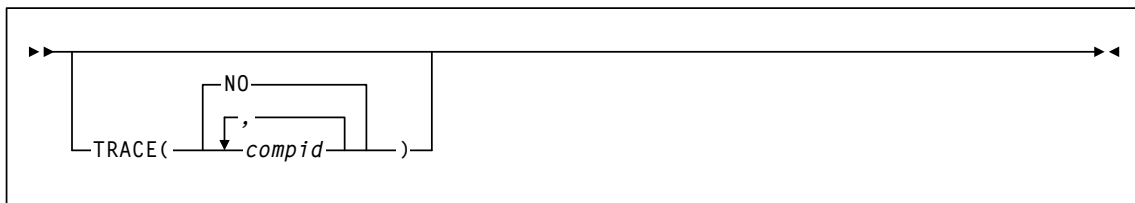
This parameter allows tracing to be turned on or off for a component. This parameter is optional.

---

### Note –

- This parameter is persistent. Once a value is specified, it overrides any default value and remains in effect until it is explicitly re-specified by another initialization or turned off using the Trace command.
  - The TRACE startup parameter should be used only as directed by StorageTek Software Support.
- 

### Syntax



### Parameter Descriptions

This section describes the values that you can specify with the TRACE startup parameter.

#### *compid*

Specifies one or more component identifiers of the MVS/CSC software components to be traced. Component identifiers must be specified as two-alphanumeric characters. Any or all of the following components can be specified:

<b>AS</b>	Address Space Communication
<b>AL</b>	Allocation support
<b>CF</b>	Configuration Manager
<b>CS</b>	Communications Server
<b>IT</b>	Initiation/Termination
<b>MD</b>	Mount/Dismount
<b>MH</b>	Message Handler
<b>OC</b>	Operator Commands
<b>RE</b>	Recovery
<b>SV</b>	Services
<b>UT</b>	Utilities
<b>PG</b>	Programmatic Interface

**NO**

Specifies that no software components be traced. This is the default value.

*Example*

The following shows an example of the TRACE startup parameter used to specify tracing of operator commands.

```
TRACE (OC)
```

## UNITMAP Startup Parameter

This parameter defines the unit map or correlation between MVS device addresses and the physical library location.

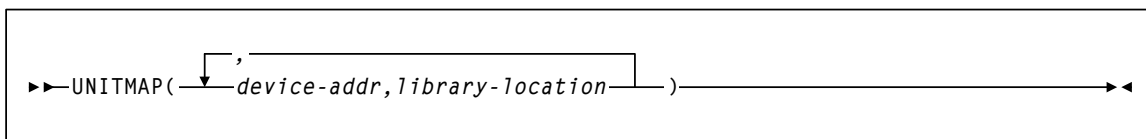
---

**Note** – This parameter is required when the server type is ACSLS. Otherwise, this parameter is ignored.

---

The unit map consists of a list of ordered pairs of MVS device addresses and library locations that map the MVS addresses to physical library locations. Every MVS device address specified in the LIBUnit parameter must also be specified in the UNITMAP parameter.

### Syntax



### Parameter Descriptions

This section describes the values that you can specify with the UNITMAP startup parameter.

#### *device-addr*

Specifies the MVS device address (or library unit address). The address must be a valid MVS device address.

#### *library-location*

Specifies the library transport address, which must be specified in the following format:

*acsid:lsmid:panelnum:devicenum*

#### *acsid*

Specifies the ACS identifier, which consists of one or two hexadecimal characters. Valid values can range from X'00' through X'7E'.

#### *lsmid*

Specifies the LSM identifier, which consists of one or two hexadecimal characters from X'0' through X'17'.

#### *panelnum*

Specifies the panel number, which consists of one or two decimal digits. Valid values can range from 1 to 10.



*devicenum*

Specifies the device number of the transport in the specified panel. The device number consists of one or two decimal numbers.

For 4480, 4490, 9490, 9490EE, and SD-3 model transports, valid device number values are 0 through 3. For T9840 model transports, valid device number values are 0 through 19. Verify that you specify device numbers that correspond to your transport models.

The device addresses and library locations must be specified as a list of ordered pairs. Each pair consists of the device address and the library location (*acs:lsm:panel:device*). Pairs must be separated by commas.

*Example*

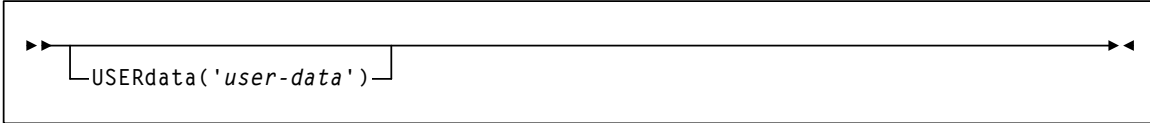
The following example shows the LIBUnit and UNITMAP startup parameters used to define the unit mapping for T9840 transports.

```
LIBUNIT (A500, A511, A522, A533)
UNITMAP (A500, 0:0:9:6, A511, 0:0:10:6, A522, 0:1:9:7, A533, 0:0:10:7)
```

## USERDATA Startup Parameter

This parameter specifies data to be defined to the user exits, which allows for pattern matching or job or data set name matching. This parameter is optional.

### *Syntax*



### *Parameter Descriptions*

This section describes the value that you can specify with the USERdata startup parameter.

#### *user-data*

Specifies user data, which can be any string of up to 100 characters in length. Beginning and ending single quotes must be specified.

### *Example*

```
USER('JOB=CSC0*,DSN=BKUP*')
```

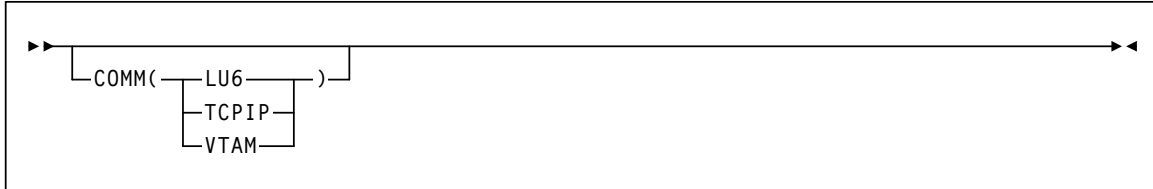


## COMM Startup Parameter

This parameter specifies the communications method used to connect the MVS/CSC subsystem to the Library Control System (LCS). This parameter is optional and the default value depends on the server environment.

This parameter is specific to SNA LU 6.2, VTAM "3270 BISYNC", and TCP/IP communications.

### *Syntax*



### *Parameter Descriptions*

This section describes the values that you can specify with the COMM startup parameter.

#### **LU6**

Specifies that the communications method used is SNA LU 6.2. This value applies to a UNIX-based LCS.

#### **TCPIP**

Specifies that the communications method used is TCP/IP. This is the default value for the UNIX-based LCS. This value can also be specified for a VM-based LCS.

#### **VTAM**

Specifies that the communications method used is VTAM "3270 BISYNC". This is the default value for the VM-based LCS. This value applies only to the VM-based LCS.

*Example*

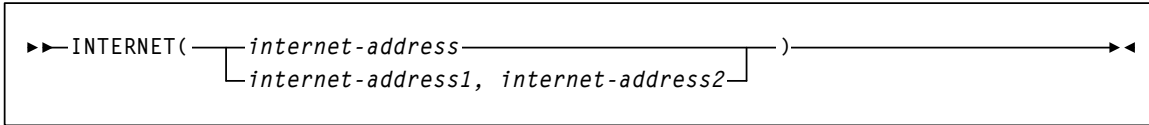
The following example shows how to specify SNA LU 6.2 as the communications method using the COMM parameter.

```
COMM(LU6)
```

## INTERNET Startup Parameter

This parameter specifies the TCP/IP Internet address of the LCS network node. This parameter is required when the communications method is TCP/IP.

### Syntax



### Parameter Descriptions

This section describes the values that you can specify with the INTERNET startup parameter.

#### *internet-address*

Specifies the Internet address of the primary local area network (LAN) or server. The Internet address is in standard internet dotted-decimal format (for example, 128.2.33.9). Valid values can range from 0 to 255.

#### *internet-addressn*

Specifies two Internet addresses for dual servers. The secondary address (*address2*) is used to connect or reconnect to a server if the MVS/CSC is not able to connect using the primary address (*address1*). The Internet address is in standard internet dotted-decimal format (for example, 128.2.33.9). Valid values can range from 0 to 255.

---

**Note** – This value applies only to the VM-based LCS.

---

### Example

The following shows an example of the INTERNET parameter used to specify the TCP/IP Internet address.

```
INTERNET(128.2.33.9)
```

## PORT Startup Parameter

This parameter specifies the TCP/IP port number assigned to a Common Library Services Logical Port (CLSLP) for TCP/IP communications.

---

### Note –

- This parameter is required when the communications method is TCP/IP and the server type is CLS. Otherwise, this parameter is ignored.
  - A systems programmer should determine the value required for this parameter based on the installation.
- 

### Syntax

```
▶▶ PORT ( port-number ) ▶▶
         [ port-number1, port-number2 ]
```

### Parameter Descriptions

This section describes the values that you can specify with the PORT startup parameter.

#### *port-number*

Specifies the primary port number. The port number corresponds to the LCS Internet address specified in the INTERNET startup parameter. The port number is specified as decimal digits. Valid values range from 256 to 65535.

#### *port-numbern*

Specifies two port numbers for dual servers. The port numbers correspond to the primary and secondary LCS Internet addresses specified in the INTERNET startup parameter. The secondary port number (*port-number2*) corresponds to the logical port number of the CLSLP in the backup server. If this port number is the same as the logical port number of the CLSLP on the primary server, only one port number is required.

The port number is specified as decimal digits. Valid values range from 256 to 65535.

---

### Note –

- Port numbers 0-255 are “well known” ports generally reserved for TCP/IP services. Ports in this range cannot be specified for the PORT startup parameter.
  - Port numbers below 1024 are generally reserved for privileged users. Specifying these ports for the PORT startup parameter may cause conflicts in certain environments.
-

*Example*

The following shows an example of the PORT parameter used to specify the TCP/IP port number assigned to the CLSLP.

```
PORT (3000)
```



## REQTIME Startup Parameter

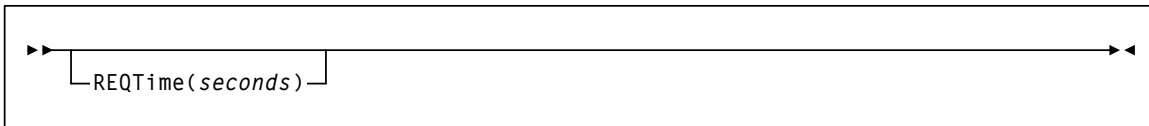
This parameter defines the number of seconds that the MVS/CSC network interface waits for the ACSLS server to complete a request before it assumes that the server is not available. After this time, the MVS/CSC either retries or fails a transaction, depending on the type of transaction.

---

**Note** – This parameter is optional and is valid only when the server type is ACSLS. Otherwise, this parameter is ignored.

---

### *Syntax*



### *Parameter Descriptions*

This section describes the value that you can specify with the REQTime startup parameter.

#### *seconds*

Specifies the number of seconds (in decimal digits) that the MVS/CSC waits for a response. Values can range from 60 to 86,399 seconds. The default value is 900 seconds.

---

**Note** – See [“Guidelines for Setting Network Timeout Parameters”](#) on page 105 for more information about setting values for this parameter. See [“Considerations When Setting ALOCTIME and REQTIME”](#) on page 107 for considerations when setting both the REQTime and ALOCTime timeout parameters.

---

### *Example*

The following shows an example of the REQTime parameter used to specify the number of seconds the MVS/CSC network interface waits for completion of the server request.

```
REQT(900)
```



## RETTIME Startup Parameter

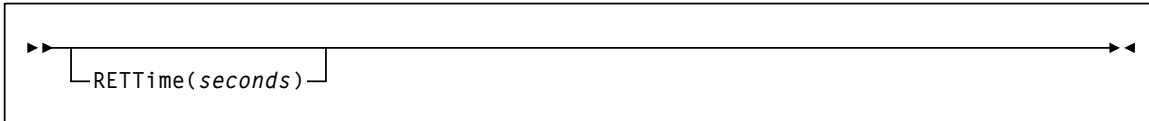
This parameter defines the minimum number of seconds that the MVS/CSC will wait between attempts to retransmit an ACSLS server request.

---

**Note** – This parameter is optional and is valid only when the communications method is TCP/IP and the server type is ACSLS. Otherwise, this parameter is ignored.

---

### *Syntax*



### *Parameter Descriptions*

This section describes the value that you can specify with the RETTime startup parameter.

#### *seconds*

Specifies the number of seconds (in decimal digits). Values can range from 0 to 999,999,999. The default value is 4.

This value multiplied by the value specified in the RETCount startup parameter defines the period for which any request not accepted by the server will cause the MVS/CSC to assume that the server is not available.

---

**Note** – See [“Guidelines for Setting Network Timeout Parameters” on page 105](#).

---

### *Example*

The following shows an example of the RETTime parameter used to specify the minimum number of seconds that the MVS/CSC will wait between attempts to retransmit an ACSLS server request.

```
RETT(10)
```

## SYMDESTN Startup Parameter

This parameter specifies the symbolic destination name used to identify the LCS when SNA LU 6.2 is the communications method. The symbolic destination name is the name of an entry in the APPC/MVS side information file, which includes the partner LU name, transaction program (TP) name, and logon mode name. APPC/MVS uses this entry to establish a conversation with the LCS application (ACSL).

---

**Note** – This parameter is required when the communications method is SNA LU 6.2.

---

### *Syntax*

▶▶—SYMDESTN(*symdestname*)—————▶▶

### *Parameter Descriptions*

This section describes the value that you can specify with the SYMDESTN startup parameter.

#### *symdestname*

Specifies the symbolic destination name. The name that you specify can consist of one- to eight-alphanumeric characters and must begin with an alphabetic character. Valid alphanumeric characters are 0-9 and upper case A-Z.

The symbolic destination name that you specify must also match the name specified on the DESTNAME keyword in the active APPC/MVS side information file.

### *Example*

The following example shows how to specify the symbolic destination name using the SYMDESTN parameter.

SYMDESTN(CSCLU1)

## TCPNAME Startup Parameter

This parameter specifies the subsystem name or address space name of the TCP/IP stack used for TCP/IP communications. Valid TCP/IP communications software includes IBM TCP/IP, CA Unicenter TCPaccess Communications Server, and CA Unicenter TCPaccess X.25 Server.

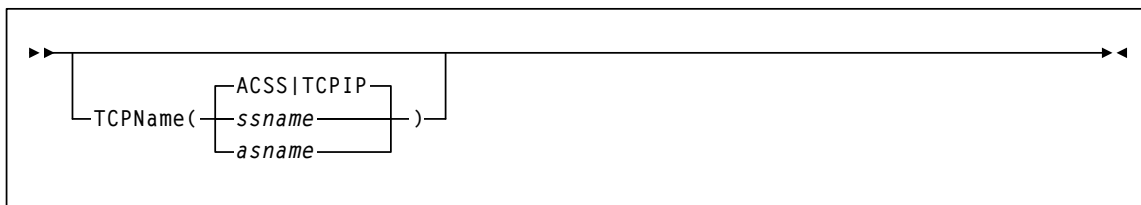
This parameter should be specified if the subsystem name or address space name was changed during the installation of the TCP/IP software.

---

**Note** – This parameter is optional and is valid only when the communications method is TCP/IP. Otherwise, this parameter is ignored.

---

### Syntax



### Parameter Descriptions

This section describes the values that you can specify with the TCPName startup parameter.

#### *ssname*

Specifies the subsystem name of the TCP/IP stack that was specified during the installation of the TCP/IP communications software. Valid subsystem names must be from one- to four-alphanumeric characters, including national characters: \$, #, and @, and must begin with an alphabetic character or national character.

#### *asname*

Specifies the address space name of the TCP/IP stack that was specified during the installation of the TCP/IP communications software. Valid address space names must be from one- to eight-alphanumeric characters, including national characters: \$, #, and @, and must begin with an alphabetic character or national character.

#### **ACSS**

Specifies the default subsystem name for CA Unicenter TCPaccess Communications Server or CA Unicenter TCPaccess X.25 Server.

#### **TCPIP**

Specifies the default subsystem name for IBM TCP/IP.

*Example*

The following shows an example of the TCPName parameter used to specify TCP1 as the subsystem name for CA Unicenter TCPaccess Communications Server.

```
TCPN(TCP1)
```

## VAPLNAM Startup Parameter

This parameter specifies the VTAM application name by which the MVS/CSC VTAM application program is known in the local VTAM network.

---

**Note** – This parameter is required when the communications method is VTAM “3270 BISYNC”.

---

### *Syntax*

▶▶—VAPLNam(*vtam-application-name*)————▶▶

### *Parameter Descriptions*

This section describes the value that you can specify with the VAPLNam startup parameter.

#### *vtam-application-name*

Specifies the one- to eight-character alphanumeric name of the VTAM application; the name must begin with an alphabetic character. Valid alphanumeric characters are 0–9 and upper case A–Z. The VTAM application name must be defined in the VTAM configuration tables.

### *Example*

The following shows an example of the VAPLNam parameter used to specify VTAM0 as the VTAM application name.

VAPLNAM (VTAM0)





# Defining Mixed Media and Devices

---

---

## Overview

The SMC TAPEREQ control statement and MVS/CSC OPTion TITLE control statement are used to support mixed media and mixed devices. Mixed media and mixed devices are supported for the UNIX-based LCS. You must use a definition data set to define these control statements; control statements cannot be issued as an operator command.

The SMC tape request (TAPEREQ) control statement is used to define the tape request characteristics for your data center. The SMC uses this information to ensure that the correct media type is used to satisfy the request, and that the cartridge is mounted on the appropriate device. Refer to the *ELS Command, Control Statement, and Utility Reference* for more information about the SMC TAPEREQ control statement.

The MVS/CSC OPTion TITLE control statement, optionally defines an identifying string, which describes the definition data set that contains these definitions.

The SMC TREQDEF command is used to load or dynamically reload the definition data set. Refer to the publication *ELS Command, Control Statement, and Utility Reference* for more information about the SMC TREQDEF command.

The following section describes the MVS/CSC OPTion TITLE control statement.

---

## OPTION TITLE Control Statement

The OPTion TITLE control statement is used to specify an identifying string for a definition data set. The identifying string can be any information that helps describe the contents of the definition data set.

The OPTion TITLE statement must be placed in the definition data set; it cannot be issued as an operator command. If more than one OPTion statement is specified in the definition data set, only the identifying string of the last OPTion statement encountered is retained.

### Syntax

▶▶—OPTion—TITLE(*identifying-string*)—————▶▶

---

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

---

### Control Statement Name

#### OPTion

Initiates the OPTion control statement.

### Parameters

#### TITLE

Specifies an identifying string for the definition data set. If this statement is omitted, the definition data set has no identifying string associated with it.

#### *identifying-string*

Specifies the identifying string. The maximum length of the identifying string is fifty characters. If the identifying string includes one or more spaces, or any characters other than alphabetic, numeric, or national (i.e., \$, @, #), it must be enclosed in quotes.

### Example

The following example illustrates the use of the OPTion TITLE control statement.

OPTion TITLE('SAMPLE IDENTIFYING STRING')

# Configuring MVS/CSC Communications

---

---

## Overview

This chapter describes the setup procedures required for the MVS/CSC to communicate with an LCS in the following scenarios:

- MVS/CSC communications with a Unix-based LCS
- MVS/CSC communications with a VM-based LCS
- MVS/CSC communications in a base or parallel Sysplex environment

---

# Configuring Communications with a Unix-Based LCS

This section describes the communications setup procedures required for the MVS/CSC to communicate with a Unix-based LCS. TCP/IP and SNA LU 6.2 are the communications methods supported for the UNIX-based LCS. Refer to your ACSLS publications for additional communications setup procedures for the server.

## Defining TCP/IP as the Communications Method

Use the COMM startup parameter (COMM (TCPIP)) to select TCP/IP as the communications method. You must also specify the INTERNET startup parameter as described below. You can also specify the TCPName startup parameter, as described in [“Specifying the TCPNAME Startup Parameter” on page 77](#), to define the subsystem name or address space name of the TCP/IP stack being used.

One of the following communications software products must be installed on your MVS system for TCP/IP communications:

- IBM TCP/IP
- CA Unicenter TCPAccess Communications Server
- Interlink CISCO IOS

Refer to the publication *Installing ELS* for the correct release levels.

---

**Note** – If you are using TCP/IP communications between ACSLS and the MVS/CSC, the TCP/IP Portmapper must be active on the MVS/CSC host.

---

## Specifying the INTERNET Startup Parameter

The following example shows the syntax for specifying the INTERNET startup parameter (where *internet-address* is the address of the LCS).

```
INTERNET (internet-address)
```

See [“INTERNET Startup Parameter” on page 62](#) for more information about specifying the INTERNET startup parameter.

## Specifying the TCPNAME Startup Parameter

The following example shows the syntax for specifying the TCPName startup parameter (where *ssname* is the subsystem name of the TCP/IP stack).

```
TCPN(ssname)
```

See [“TCPNAME Startup Parameter” on page 69](#) for more information about specifying the TCPName startup parameter.

### *MVS/CSC and IBM TCP/IP*

If you are using IBM TCP/IP for TCP/IP communications, use the following guidelines:

- If the TCPName parameter in the MVS/CSC startup PROC specifies a subsystem name other than TCPIP, the MVS/CSC startup PROC must contain a SYSTCPD DD statement, and this data set must include a TCPIPJOBNAME parameter specifying the alternate subsystem name. The TCPIPJOBNAME parameter must match the JCL procedure name for the alternate TCP/IP stack. Refer to the *IBM TCP/IP Configuration Guide* for more information.
- An OMVS segment for the MVS/CSC started task must be defined in RACF (or the equivalent security facility) prior to using MVS/CSC with IBM TCP/IP 3.4 or later.
- If you are using TCP/IP communications between ACSLS and the MVS/CSC, the TCP/IP Portmapper must be active on the MVS/CSC host.

### *MVS/CSC and CA Unicenter TCPaccess Communications Server*

If you are using CA Unicenter TCPaccess Communications Server for TCP/IP communications, use the following guidelines:

- Add the TCPaccess load library as the first data set in the STEPLIB concatenation of the MVS/CSC startup PROC.
- When using multiple TCP/IP stacks, if a TCPName value other than ACSS is specified in the MVS/CSC startup parameter file, the alternate subsystem name must be defined in member DNRALCxx of the TCPaccess parameter library. This alternate name must match the subsystem name specified for the alternate TCP/IP stack at startup.

## Defining SNA LU 6.2 as the Communications Method

APPC/MVS and VTAM must be installed and set up on both the client system and LCS for SNA LU 6.2 communications. See [“Setting Up VTAM and APPC/MVS for SNA LU 6.2 Communications” on page 79](#) for information about setup procedures.

SNA LU 6.2 is selected as the communications method using either the COMM startup parameter, or the SRVRLIST startup parameter. If you use the COMM startup parameter to specify SNA LU 6.2, you must also specify the symbolic destination name using the SYMDESTN startup parameter.

### *Specifying the SYMDESTN Startup Parameter*

Use the SYMDESTN startup parameter to specify the symbolic destination name (which represents the transaction program, logon mode, and partner logical unit) for SNA LU 6.2 communications. The following example shows how to specify the SYMDESTN startup parameter (where *symdestname* is the symbolic destination name).

```
SYMDESTN ( symdestname )
```

The symbolic destination name you specify can consist of one- to eight-alphanumeric characters and must begin with an alphabetic character. Valid alphanumeric characters are 0-9 and upper case A-Z.

The symbolic destination name you specify must also match the name specified on the DESTNAME keyword in the APPC/MVS side information file (see [“Creating the Side Information File” on page 81](#) for information about the APPC/MVS side information file).

### *Setting Up VTAM and APPC/MVS for SNA LU 6.2 Communications*

**Note to Existing APPC/MVS Users:** If you currently have a system base LU defined to APPC/MVS and a local LU defined to VTAM, you need only to add a side information entry to the existing APPC side information file that identifies the LCS (see [“Creating the Side Information File” on page 81](#)). You should also verify that your VTAM application definition statement used to define the local LU is consistent with the definition statement shown in [“Setting Up VTAM” on page 80](#).

The following sections describe the procedures for setting up VTAM and APPC/MVS on the MVS client system for SNA LU 6.2 communications. (Refer to the appropriate LCS documentation for server setup procedures.) These procedures are specifically for sites that are not currently utilizing APPC/MVS services. It is recommended that an experienced network systems programmer or administrator who is familiar with VTAM, APPC, and the physical network hardware used to connect the client system to the LCS perform these procedures.

Refer to the following publications for additional information about setting up APPC/MVS and VTAM.

- *IBM MVS/ESA Planning: APPC Management*
- *IBM VTAM Customization*
- *IBM VTAM Network Implementation Guide*
- *IBM VTAM Operations*
- *IBM VTAM Programming for LU 6.2*
- *IBM VTAM Resource Definition Reference*

## Setting Up VTAM

To set up VTAM for SNA LU 6.2 communications:

### 1. Define a local LU to VTAM.

You must define a local LU to VTAM with a VTAM application (APPL) definition statement. APPLs reside in members of the SYS1.VTAMLST system library. The following figure shows a sample APPL statement used to define a local LU named CSCLU1 to VTAM. A sample APPL statement for defining a local LU to VTAM resides in sample library member LU6APPL.

```

APPLCSC      VBUILD TYPE=APPL
CSCLU1      APPL ACBNAME=CSCLU1,
              APPC=YES,
              AUTOSES=0,
              DDRAINL=NALLOW,
              DLOGMOD=APPCHOST,
              DMINWNL=5,
              DMINWNR=5,
              DRESPL=NALLOW,
              DSESLIM=10,
              LMDENT=19,
              MODETAB=LOGMODES,
              PARSESS=YES
              SECACPT=CONV,
              SRBEXIT=YES,
              VPACING=1
  
```

The local LU name defined to VTAM must also be defined to APPC/MVS as the system base LU. See [“Setting Up APPC/MVS” on page 81](#) for information about defining the local LU to APPC/MVS.

### 2. Define an APPC logon mode.

Logon mode entries, which are required for LU 6.2 sessions, must be compiled into the logon mode table that exists in SYS1.VTAMLIB. Member ATBLJOB of SYS1.SAMPLIB contains sample JCL to create a logon mode table.

The logon mode entries required are SNASVCMG and APPCHOST. These entries reside in member ATBLMODE of SYS1.SAMPLIB. Refer to *IBM MVS/ESA Planning: APPC Management* for more information about defining the local LU and logon mode entry.

Additional VTAM setup is required to define the physical connection between the MVS/CSC and the LCS. An experienced network systems programmer should be involved in defining the connection, and this setup should be completed before installing the MVS/CSC.



## Setting Up APPC/MVS

Define a system base LU that APPC/MVS uses to establish sessions between the MVS/CSC and the LCS. To accomplish this task, create an APPCPMxx member (xx is the two-character suffix used on the MVS START APPC command) in SYS1.PARMLIB, and add the following statements:

```
LUADD ACBNAME(CSCLU1) BASE NOSCHED TPDATA(side_info_file)
SIDEINFO DATASET(side_info_file)
```

where the name specified on the ACBNAME keyword is the system base LU, and *side\_info\_file* is the name of the VSAM key sequenced data set (KSDS) that contains side information for the installation (see [“Creating the Side Information File” on page 81](#) for information about defining the side information). The name specified on the ACBNAME keyword must match the name of the local LU defined to VTAM (for example, CSCLU1). An example for defining a system base LU to APPC/MVS resides in member APPCPMY of the SAMPLIB library.

For more information about the APPCPMy member, refer to *IBM MVS/ESA Planning: APPC Management*.

## Creating the Side Information File

Create a side information file and add an entry that identifies the LCS. The following figure shows sample JCL used to create a side information file. Sample JCL for creating a side information file resides in member DEFAPPC of the SAMPLIB library.

```
//DEFAPPC JOB job card information
//DEFSIDE EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
    DEFINE CLUSTER (NAME(side_info_file) -
        VOLUME(volser) -
        INDEXED REUSE -
        SHAREOPTIONS(3 3) -
        RECORDSIZE(248 248) -
        KEYS(112 0) -
        RECORDS(5 5)) -
    DATA -
        (NAME(side_info_file.DATA)) -
    INDEX -
        (NAME(side_info_file.INDEX))
//INITSIDE EXEC PGM=ATBSDFMU
//SYSPRINT DD SYSOUT=*
//SYSSDLIB DD DSN=side_info_file,DISP=SHR
//SYSSDOUT DD SYSOUT=*
//SYSIN DD *
    SIADD
        DESTNAME(LIBSTAT)
        TPNAME(CSCI)
        PARTNER_LU(LSLU)
        MODENAME(APPCHOST)
```

The values in lowercase represent information that you must supply. The following list describes the values specific to SNA LU 6.2 communications.

**side\_info\_file**

Specifies the name used on the SIDEINFO DATASET keyword in the APPCPMxx member of SYS1.PARMLIB.

**DESTNAME**

Specifies the symbolic destination name of the entry. This name can consist of one- to eight-alphanumeric characters and must begin with an alphabetic character. Valid alphanumeric characters are 0-9 and upper case A-Z. The name you specify must also match the symbolic destination name specified on the MVS/CSC startup parameter.

**TPNAME**

Specifies the name of the transaction program used on the LCS. This name must match the transaction program name included in the side information entry that was defined for the LCS (i.e., CSCI).

**PARTNER\_LU**

Specifies the partner LU name that is used to identify the LCS. This name must match the partner LU name included in the side information entry that was defined for the LCS. This LU name must also be defined locally on the LCS system and must be available to the MVS/CSC.

**MODENAME**

Specifies the name of the logon mode that controls the session between the MVS/CSC application and the server application. This name must match the logon mode name included in the side information entry that was defined for the LCS (i.e., APPCHOST).

Refer to *IBM MVS/ESA Planning: APPC Management* for more information about the side information file.

*Starting APPC/MVS and VTAM*

Prior to starting the MVS/CSC, APPC/MVS and VTAM should be running, and all physical and logical units used to connect the MVS/CSC to the LCS should be active. You can do this either manually with operator commands, or automatically at IPL.

When starting APPC using the MVS START APPC command, specify the last two characters of the APPCPMxx member name on the START command. For example:

```
START APPC , SUB=MSTR , APPC=XX
```

where xx is the two-character suffix from the APPCPMxx member name. For more information about the APPCPMxx member, starting and stopping APPC, and showing the status of APPC, refer to *IBM MVS/ESA Planning: APPC Management*. For information about starting VTAM and activating logical units, refer to *IBM VTAM Operations*.

---

# Configuring Communications with a VM-Based LCS

This section describes the communications setup procedures required for the MVS/CSC to communicate with a VM-based LCS. TCP/IP and VTAM “3270 BISYNC” are the communications methods supported for the VM-based LCS. Refer to your CLS publications for additional communications setup procedures for the server.

## Defining TCP/IP as the Communications Method

Use the COMM startup parameter (COMM (TCPIP)) to select TCP/IP as the communications method. You must also specify the INTERNET startup parameter as described below. You can also specify the TCPName startup parameter, as described in [“Specifying the TCPNAME Startup Parameter”](#), to define the subsystem name or address space name of the TCP/IP stack being used.

One of the following communications software products must be installed on your MVS system for TCP/IP communications:

- CA Unicenter TCPAccess Communications Server
- CA Unicenter TCPAccess X.25 Server

Refer to the publication *Installing ELS* for the correct release levels.

## Specifying the INTERNET Startup Parameter

The following example shows the syntax for specifying the INTERNET startup parameter (where *internet-address* is the address of the LCS).

```
INTERNET(internet_address)
```

See [“INTERNET Startup Parameter” on page 62](#) for more information about specifying the INTERNET startup parameter.

## Specifying the TCPNAME Startup Parameter

The following example shows the syntax for specifying the TCPName startup parameter (where *ssname* is the subsystem name of the TCP/IP stack).

```
TCPN(ssname)
```

See [“TCPNAME Startup Parameter” on page 69](#) for more information about specifying the TCPName startup parameter.

## Specifying the PORT Startup Parameter

You must also specify a TCP/IP port address for the CLS Logical Port (CLSLP) to which the MVS/CSC will connect for CLS servers. This is done by specifying the PORT startup parameter (where *port-number* is the TCP/IP port number for use by TCP/IP communications), as shown below:

```
PORT (port-number)
```

See [“PORT Startup Parameter” on page 63](#) for more information about specifying the PORT startup parameter.

## MVS/CSC and CA Unicenter TCPaccess Communications Server

If you are using CA Unicenter TCPaccess Communications Server for TCP/IP communications, use the following guidelines:

- Add the TCPaccess load library as the first dataset in the STEPLIB concatenation of the MVS/CSC startup PROC.
- When using multiple TCP/IP stacks, if a TCPName value other than ACSS is specified in the MVS/CSC startup parameter file, the alternate subsystem name must be defined in member DNRALCxx of the TCPaccess parameter library. This alternate name must match the subsystem name specified for the alternate TCP/IP stack at startup.

## Defining VTAM “3270 BISYNC” as the Communications Method

VTAM “3270 BISYNC” is the default communications method for the VM-based LCS. It is assumed that VTAM is installed on your client system.

VTAM “3270 BISYNC” is selected as the communications method using the COMM startup parameter (COMM (VTAM)). If VTAM “3270 BISYNC” is the selected communications method, the VAPLnam startup parameter and the VTAMLST data set must be specified (see “Specifying the VAPLNAM Startup Parameter” and “Specifying the VTAMLST Data Set”).

### Specifying the VAPLNAM Startup Parameter

The VTAM application name must be specified for communications with VTAM “3270 BISYNC”. This is done by specifying the VAPLnam startup parameter (where *vtam-application-name* is the name of the VTAM application) as shown below:

```
VAPLNAM(vtam-application-name)
```

See “[VAPLNAM Startup Parameter](#)” on page 71 for more information about specifying the VAPLnam startup parameter.

### Specifying the VTAMLST Data Set

The VTAMLST data set must be specified to add information to the VTAM application major node and to the Network Control Program (NCP) stage-one SYSGEN macro sections.

### Application Major Node Definitions

The following statement must be added in the application major node definitions section of the VTAMLST data set:

```
vtam-application-name APPL AUTH=(NVPACE), EAS=1
```

The APPL statement defines an application program node and some of the communications capabilities of the application program. The APPL statement defines the logical unit (LU). The EAS parameter specifies the number of concurrent sessions a subsystem may have with other logical units (LU-LU sessions).

In the example above, *vtam-application-name* is the VTAM application name that is designated in the MVS/CSC startup parameter file. The VTAM application name specified here must be the same as the name specified in the VAPLnam startup parameter.

If you need to create an application major node, the following statement must be included and should precede the VTAM application name definition:

```
node-id VBUILD TYPE=APPL
```

The VBUILD statement describes the node, and the TYPE parameter specifies that the VBUILD statement defines a local major node.

If you need to assign multiple VTAM application names for several MVS/CSC subsystems, repeat the VTAM application name definitions. Each definition must contain a unique MVS/CSC name. For example, to declare three VTAM application names for separate MVS/CSC subsystems, the following definitions would be specified in the application major node section:

```
appl-id1 APPL AUTH=(NVPACE), EAS=1  
appl-id2 APPL AUTH=(NVPACE), EAS=1  
appl-id3 APPL AUTH=(NVPACE), EAS=1
```

The only difference in the application definitions is the MVS/CSC *appl-id* name.

---

**Note** – The *appl-id* must be unique for each application defined.

---

## Network Control Program (NCP) Requirements

Add the section shown in the following figure to the NCP stage-one SYSGEN MACRO sections. The NCP example defines the communications configurations enabling VTAM to process communications between the MVS/CSC and the CLS systems. In the example, lowercase items indicate that your own applicable system identifiers must be specified. For example, where group-name appears, replace that with your system group-name identifier.

```

group-name GROUP DIAL=NO, LNCTL=BSC,
                TYPE=NCP, DUPLEX=HALF,
                PAUSE=2,
                POLIMIT=(20, QUEUE),
                SERVPRI=OLD,
                POLLED=YES,
                CUTYPE=3271,
                CRETRY=255,
                OWNER=MVS host-id, (from PCCU macro, if applicable)
                RETRIES=(10,1,1),
                DLOGMOD=D4B32782, (IBM Default)
                USSTAB=USS table name (your USSTABLE name)

*****
*****

VTAM line-name LINE ADDRESS=(37x5 LIB or port number)
                SESSION=1,
                SERVLIM=1

Service order-name SERVICE ORDER=(PU-name, LU-name:)

PU-name        CLUSTER GPOLL=40407F7F.
                XMITLIM=1

LU-name        TERMINAL TERM=3277,
                ADDR=60604040,
                POLL=40404040,
                CONV=YES,
                FEATUR2=(MODEL2)

```

## Verifying Communications Setup

If after issuing the MVS Start command, error messages are issued that indicate communications problems, use the procedures described in the following sections to test your communications setup for the VM-based LCS (CLS server) using either TCP/IP or VTAM "3270 BISYNC".

---

**Note** – No verification of communications parameters is done for the UNIX-based or MVS-based LCS.

---



## Testing CLS Communications Parameters for TCP/IP

If (1) a communications error is indicated at startup and (2) the server type is CLS and the communications method is TCP/IP, make sure that the addresses entered in the CLS configuration database and those specified in the MVS/CSC startup parameters are identical.

The MVS/CSC startup parameters used when COMM(TCPIP) is specified are INTERNET and PORT. The values specified for INTERNET and PORT **must** be the same as the values you specify for the LOCAL SOCKET definition using the CLS Configuration Management (CLSCM) program (for CLS). If these addresses are not the same, an error condition occurs and a message is returned.

## Testing CLS Communications Parameters for VTAM “3270 BISYNC”

Because the Library Control System (LCS) is considered a “real terminal” by the client system, the signon sequence imitates that of a real terminal in operation. If the default signon script does not work, execute each step of the signon script manually, observe the actual responses, and note the exact sequence that an operator would use to sign on to the MVS/CSC application.

Use the CLS Configuration Management program to recreate or update the signon script.

1. Do the following to verify that the individual components are active:
  - a. Enter QUERY CLSCOMM to verify that the VM/Pass-Through (PVM) service machine CLSCOMM is active.
  - b. Enter SMSG CLSCOMM STATUS LINE “vln” (where vln is the PVM virtual line) to verify that the CSC application line is active.

---

### Note –

- The PVM status command is only accepted by PVM from a VM userid identified as authorized (in the PVM CONFIG file) to issue restricted PVM commands.
  - The PVM virtual line, which is defined in PVM CONFIG and started by PROFILE PVM, is attached using the command CP ATTACH rln \* vln (where rln is the address of the REAL line).
- 

- c. Enter SMSG CLSCOMM QUERY SYSTEM to verify that the link from CLSCOMM to 37x5 is active.
2. Use the 3270 terminal to sign on to the MVS/CSC system. From a terminal attached to the VM Library Control System which is not logged on, enter DIAL CLSCOMM. A menu displaying the client node names is displayed.

Following the instructions on the menu screen, select the appropriate client node and press <Enter>. A message “USER CONNECTED TO PORT 0” is briefly displayed, followed by the client system prompt screen.

The cursor and data field position are important to the VTAM 3270 terminal handler when data is input using an attention key such as <Enter>.

Press <Clear> to get another prompt, then enter LOGON APPLID(CSC0).

3. Record the response, the position of cursor (column and row) and data field (column and row) when you entered the LOGON command.
4. Encode the response in a script and enter these coordinates in the signon script. Also supply the VM/Pass-Through service machine name, the MVS/CSC node name, and the line number as defined to VM/Pass-Through.

The MVS/CSC now tries to establish further communications with the terminal, but the process is finished. Press the <Back Tab> key to return to the beginning of a field. Then enter the Pass-Through termination string (usually #####).

5. Execute the configured signon script. Use any resulting diagnostic messages to correct possible errors.

# Verifying MVS/CSC Configuration

---

---

## Overview

Before the initial startup of the MVS/CSC, you can invoke the Configuration Verification utility to verify that the configuration definition and the initial setup of the MVS/CSC is operable. The Configuration Verification utility validates each parameter against the MVS environment.

---

## Running the Configuration Verification Utility

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

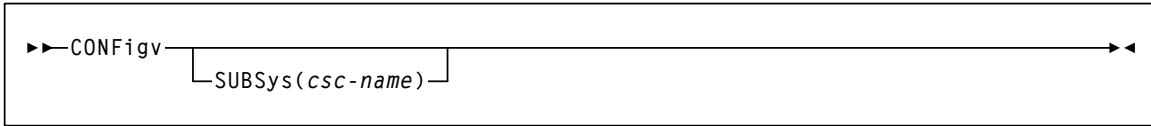
- User-selected startup parameters
- Device esoteric definitions
- LCS configuration compatibility

The Configuration Verification utility is primarily used to verify the values specified for the startup parameters at initialization time. It is also 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

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 mods */
// 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 6-1** 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 HSC on the same MVS system, the STEPLIB DD statement that identifies the MVS/CSC load libraries is required (see FIGURE 6-1). 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 below.
- A formatted printout of MVS/CSC startup parameters verified (when all parameters are valid). Sample output is shown in [FIGURE 6-3 on page 94](#).
- A formatted printout of MVS/CSC startup parameters verified including error conditions for invalid parameters. Sample output is shown in [FIGURE 6-4 on page 95](#).

```
SCUADMIN (7.0.0) StorageTek MVS Client/Server System Utility PAGE 001
TIME 10:51:02 Control Card Image Listing DATE 2010-06-08

CONF SUBS(CSC1)
```

**FIGURE 6-2** 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 2009-06-08

*****
*****
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
*
COMM(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 6-3 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
*
COMM (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 6-4 CONFIGV Sample Output - Some Invalid Parameters (VM-Based LCS with TCPIP)





# Starting and Stopping the MVS/CSC

---

---

## Overview

The MVS/CSC must be operable before the library can be accessed to perform automatic tape handling. This chapter describes procedures for starting and stopping the MVS/CSC.

---

**Note** – MVS/CSC can be initialized before the SMC without producing error messages. However, an SMC subsystem must be active to influence tape allocations and intercept MVS messages.

---

## Pre-Initializing the MVS/CSC

Both the MVS/CSC and the LCS must be initialized before the library can be accessed. The MVS/CSC can either be pre-initialized by the MVS subsystem pre-initialization routine during the initial program load (IPL) of the MVS host system, or by issuing the MVS SETSSI command to dynamically define the MVS subsystem name.

The subsystem pre-initialization routine is identified in the MVS IEFSSNyy member of SYS1.PARMLIB. The pre-initialization routine is executed once for each IPL of the MVS host system. The pre-initialization routine establishes unique identification of the MVS/CSC subsystems in the MVS host system. Once the IPL of MVS has completed and the pre-initialization routine has executed, you can start the MVS/CSC subsystem.

Issuing the MVS Start command invokes the subsystem initialization routine. This routine determines what parameters are in effect, initializes communications, performs any cleanup necessary (such as resource recovery), and begins normal processing.

Before initialization of the MVS/CSC, the MVS/CSC startup parameters must be specified. These parameters reside in a member of a partitioned data set or in a sequential data set. The parameters are identified by the SCSPARM DD name in the MVS/CSC startup procedure.

During MVS/CSC startup processing, the MVS/CSC synchronizes the state of its resources with the LCS and MVS using its synchronization processing. For the VM-based LCS, the MVS/CSC sends an *availability* message to the CLS during initialization. The MVS/CSC waits for a return availability message from the CLS before processing can occur.

# Starting the MVS/CSC

The MVS Start command initializes the MVS/CSC. The syntax of the Start command is:

```
START csc-proc-name [ , PRM=RESET|COLD ]
```

START or S is the keyword for the MVS Start command. The value specified for *csc-proc-name* is the name of a member in a procedure library. You can specify the following values for the PRM parameter:

## RESET

Instructs the MVS/CSC to reset its internal initialization and termination flags. This parameter may be required if the last execution on the MVS/CSC was terminated by an MVS Force command.

## COLD

Instructs the MVS/CSC to rebuild its internal control structures. This parameter is required if migrating from a prior version of the MVS/CSC to this version of MVS/CSC and no IPL of the MVS host system was performed. This parameter may also be required if an MVS/CSC PTF has been applied and no IPL of MVS was performed.

---

**Note** – The AMPND and NOAMPND startup parameters are no longer supported. Automation of pending mounts is now by provided by the SMC. Refer to the publication *Configuring and Managing SMC* form more information.

---

The MVS/CSC system responds by displaying console messages (shown in the following figure). The messages explain that the MVS/CSC subsystem started at the time shown and that initialization completed. Specific messages indicating that a session with the LCS was successfully initialized will be issued depending on the configuration and parameters specified.

```
IEF403I CSC0 - STARTED - TIME=08.45.56
SCS0500I COPYRIGHT (C) 1992, 2010, ORACLE AND/OR ITS AFFILIATES.
ALL RIGHTS RESERVED.
...
...
SCS0517I MVS/CSC subsystem CSC0 initialization complete
```

---

## Stopping the MVS/CSC

MVS/CSC processing can be stopped by causing an orderly shutdown or an immediate shutdown.

- During an orderly shutdown, the MVS/CSC waits for processing of all activities in progress to be completed before completing shutdown.
- During an immediate shutdown, the MVS/CSC stops all processing and immediately begins shutdown processing.

Any of the following MVS commands can be used to stop MVS/CSC processing:

- **STOP**

The MVS Stop command causes an orderly shutdown of the MVS/CSC.

```
P csc-proc-name
```

P is the keyword for the MVS Stop command. The value specified for *csc-proc-name* is the name of the MVS/CSC started task currently running.

- **CANCEL**

The MVS Cancel command causes all MVS/CSC operations to be cancelled and causes an immediate shutdown of the MVS/CSC.

```
Cancel csc-proc-name, DUMP
```

Cancel is the keyword for the MVS Cancel command. The value specified for *csc-proc-name* is the name of the MVS/CSC started task currently running. The optional DUMP parameter instructs the MVS host system to produce a dump of the MVS/CSC address space.

- **FORCE**

The MVS Force command causes all MVS/CSC operations to be cancelled and causes an immediate shutdown of the MVS/CSC. However, unlike the Cancel command, the Force command may cause unpredictable results when the MVS/CSC is restarted. Therefore, use of this command is not recommended.

```
FORCE csc-proc-name
```

FORCE is the keyword for the MVS Force command. The value specified for *csc-proc-name* is the name of the MVS/CSC task currently running.

---

# Communications Considerations When Stopping the MVS/CSC

The following communications access method software should be operational before starting the MVS/CSC subsystem.

- IBM TCP/IP, or CA Unicenter TCPAccess products
- VTAM for “3270 BISYNC” communications
- APPC/MVS and VTAM for SNA LU 6.2 communications

If the communications software must be stopped, the MVS/CSC should be stopped using the MVS Stop (or Cancel) command before stopping the communications software.



## Considerations When Setting Network Timeout Parameters

---

This appendix contains guidelines for setting the network timeout parameters for the MVS/CSC (RETime, RETCount, and REQTime) and the equivalent parameters on the ACSLS server. The default values will be adequate for most installations but it may be necessary to customize these parameters for some configurations.

The network timeout parameters are common to MVS/CSC and ACSLS. However, the ACSLS timeout parameters are defined with unique names, which do not match those used by MVS/CSC. These names are summarized in [TABLE A-1](#).

**TABLE A-1** Timeout Parameter Names Cross-Reference Table

MVS/CSC Name	ACSL Name	Description
RETime	CSI_RETRY_TIMEOUT	Seconds between attempts to initiate a request.
RETCount	CSI_RETRY_TRIES	Number of times a connection initiation is attempted.
REQTime	CSI_CONNECT_AGETIME	Seconds allowed for a request to terminate.

Throughout this publication, MVS/CSC parameter names are used to refer to all network timeout parameters.

In addition to the timeout parameters listed in [TABLE A-1](#), MVS/CSC provides the ALOCTime timeout parameter, which is specific to volume lookup requests issued by the MVS/CSC. See [“Considerations When Setting ALOCTIME and REQTIME” on page 107](#) for considerations when using this parameter.

---

**Note** – There are considerations that you should take into account when setting both the ALOCTime timeout parameter and REQTime timeout parameter. See [“Considerations When Setting ALOCTIME and REQTIME” on page 107](#) for more information.

---

## Overview of MVS/CSC Network Transaction

When the MVS/CSC makes a request to the server, it transmits a packet with information about the request in addition to information about the system and task making the request. The MVS/CSC then sets up an environment that allows the server to transmit the response. When the server is ready, it uses the system and task information (from the request packet) to connect back to the waiting MVS/CSC. The MVS/CSC acknowledges the server and uses the response packet to reply to the calling task or to generate another server request.

The MVS/CSC may detect a server or network failure when it tries to initialize a new transaction or when the transaction has not been completed in a reasonable amount of time. The RETCount and RETTime parameters determine the number of retries and time between retries when a network transaction is being initialized. The REQTime parameter determines the amount of time the MVS/CSC will wait for the transaction to terminate. The ALOCTime parameter determines the amount of time the MVS/CSC waits for the server to respond to a query request for volume location or volume attribute information.

The default values for these parameters will be adequate for most installations. The following situations may require customization of these variables:

- Remote installation of server and MVS/CSC
- Slow network performance due to other network utilization (the MVS/CSC puts negligible load on a network)
- Large quantities of simultaneous mounts and dismounts (for example, more than 10 per minute)



---

# Guidelines for Setting Network Timeout Parameters

The following guidelines should be considered when setting these parameters.

---

**Note** – If the first two guidelines are followed, deviations from the others may be necessary if problems persist.

---

- The MVS/CSC REQTime value must be greater than the server RETTime value multiplied by the server RETCount value. Likewise, the server REQTime value must be larger than the MVS/CSC RETTime value multiplied by the MVS/CSC RETCount value. For example, if the server has the default RETTime value of 4 and a RETCount value of 5, the MVS/CSC could have them set at 2 and 10 (or even 3 and 6) so that the product is approximately the same (in this case, 18 to 20 seconds).
- When changing the values for RETTime and RETCount, make changes to one variable or the other in small increments. For example, add 1 second to RETTime; if the problems persist, add 1 to RETCount. Then apply similar changes to the other end (server or MVS/CSC).
- The MVS/CSC REQTime value should not be set higher than necessary. Increasing the REQTime value will delay the ability of the MVS/CSC to detect server or network failures.
- If SCS3232 messages are seen regularly, the MVS/CSC may be performing unnecessary recovery. Increasing the values set for the RETTime or RETCount will eliminate unnecessary recovery.
- When the MVS/CSC host is remote from the server host, the RETTime or RETCount value may need to be increased.
- If very high rates of mounts, dismounts, and set scratch processing are occurring between one MVS/CSC and its server, it may be necessary to set different RETTime values for the server and the MVS/CSC. The product of the RETTime value multiplied by the RETCount value should be similar at both ends. For example, if the server has the default RETTime value of 4 and a RETCount value of 5, the MVS/CSC could have them set at 2 and 10 (or even 3 and 6) so that the product is approximately the same (in this case, 18 to 20 seconds). The objective in this single MVS/CSC setup is to try to keep the product of RETTime and RETCount values as small as possible and approximately the same for the server and the MVS/CSC.
- If a high rate of simultaneous mount, dismount, and set scratch processing is occurring between multiple MVS/CSCs and the server, the RETTime or RETCount value on each MVS/CSC host may need to be slightly higher than the same values for the server.

## Considerations When Setting ALOCTIME

During device allocation, the MVS/CSC issues query requests to the server for volume location and volume attribute information. Waiting for the server to respond to a query request for an extended period of time can adversely affect the response time for certain system functions, and can add to the amount of time required to run a job. The MVS/CSC provides the ALOCTime startup parameter, which lets you specify the amount of time that the MVS/CSC waits for the server to respond to a query request for volume location and volume attribute information.

If the server does not respond to a query request for volume location information before this time expires, the SMC does not modify the MVS Eligible Device List (EDL), or the JES3 Intermediate Job Summary table (IJS) to exclude ineligible devices for the allocation request. This can result in the allocation of nonlibrary devices (if any exist), or pass-thru activity in an ACS containing multiple LSMs. If the server does not respond to a query request for volume attribute information before this time expires, the SMC does not process specific recording-technique or media-type requirements for the allocation request.

When initializing the MVS/CSC for the first time, use the default value (55 seconds) for the ALOCTime parameter for the UNIX-based LCS, and modify this value when necessary. For the VM-based LCS, set the ALOCTime value to be twice the heartbeat interval, or use the default value (whichever is greater), and modify this value when necessary.

MVS/CSC issues messages SCS0746E and SCS0747E to report allocation timeouts. Refer to the publication *ELS Messages and Codes* for more information about these messages.

---

## Considerations When Setting a Low ALOCTIME Value

The time it takes to allocate a cartridge transport adds to the overall elapsed time for the job. If this time becomes excessive, setting a low value for the ALOCTime startup parameter will minimize the increase in elapsed time for the job. However, if the server does not respond to a query request within the specified time, the volume location and volume attribute information is not obtained and can result in the following:

- Nonlibrary devices (if any exist) might be allocated for scratch or specific volumes that would normally be allocated on library devices.
- Library devices might be allocated for specific volumes, but the desired recording technique and media type might not be selected. This applies only to mixed-media and mixed-device configurations.

---

## Considerations When Setting a High ALOCTIME Value

Serialization of MVS resources is another consideration when specifying the amount of wait time for the server to respond to a query request. Shared ENQs are held on the following MVS resources during device allocation:

- QNAME - SYSIEFSD
  - RNAME - Q4
  - RNAME - CHNGDEVS
  - RNAME - DDRTPUR
  - RNAME - DDRDA

An exclusive ENQ on one or more of these MVS resources is required for the following MVS processes:

- Varying devices online or offline
- MVS UNLOAD command
- DDR tape swaps
- Dynamic activation of a new I/O configuration

Since these processes require an exclusive ENQ, they cannot complete while a job is in device allocation. Since these processes are performed infrequently for most installations, you can specify a higher ALOCTime value to accommodate a longer server response time for query requests when necessary. However, if optimal response time is required for these processes, setting a lower ALOCTime value can minimize the exposure for a longer response time. This frees up the MVS resources to be used exclusively by these processes.

---

## LCS Considerations

For the UNIX-based LCS, the ALOCTime value overrides the value specified on the REQTime parameter for query requests during device allocation. For the VM-based LCS, the ALOCTime value operates independently of the heartbeat interval.

---

## Considerations When Setting ALOCTIME and REQTIME

This section describes considerations that you should take into account when setting both the ALOCTime and REQTime timeout startup parameters. It describes considerations when setting these startup parameters in a single-LCS environment, as well as in a multiple-LCS environment.

## Considerations When Setting ALOCTIME and REQTIME in a Single-LCS Environment

The following considerations should be taken into account when setting both the ALOCTime and REQTime timeout startup parameters in a single-LCS environment.

**ALOCTIME Equal To REQTIME**—When both ALOCTime and REQTime are set to the same value, the timer associated with ALOCTime should expire first. If the server does not respond to a query request for volume location and volume attribute information before the time expires on ALOCTime, MVS/CSC issues a message indicating that a timeout has occurred. Processing of query requests for volume location and volume attribute information occurs as described in [“Considerations When Setting ALOCTIME” on page 106](#).

**ALOCTIME Less Than REQTIME**—When the time set on ALOCTime is less than the time set on REQTime, and the server does not respond to a query request for volume location and volume attribute information before the time expires on ALOCTime, MVS/CSC issues a message indicating that a timeout has occurred. Processing of query requests for volume location and volume attribute information occurs as described in [“Considerations When Setting ALOCTIME” on page 106](#).

**ALOCTIME Greater Than REQTIME**—When the time set on ALOCTime is greater than the time set on REQTime, and the server does not respond to a query request for volume location and volume attribute information before the time expires on REQTime, a message is issued indicating that the server is unavailable. All outstanding query requests for volume location and volume attribute information are assigned a return code indicating that the requests failed. Processing of query requests for volume location and volume attribute information occurs as defined in [“Considerations When Setting ALOCTIME” on page 106](#).

# 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.).

## Migration and Coexistence

---

This appendix provides guidelines for migration from previous releases of the MVS/CSC to MVS/CSC Release 7.0. Reverse migration is also discussed.

In addition, this appendix provides MVS/CSC coexistence and compatibility guidelines.

---

**Note** – Your site may have specific conditions that require special precautions and procedures. If so, contact StorageTek Software Support for assistance.

---

---

## Migration

### Migrating From MVS/CSC Release 4.0 or later to Release 7.0

Perform the following steps:

- Make the MVS/CSC 7.0 LINKLIB available through the MVS LINKLIST facility or a STEPLIB DD statement in the MVS/CSC started task procedure.
- Make any parameter changes necessary to enable new functionality, if desired.
- Start MVS/CSC 7.0.

---

**Note** – When migrating to a new MVS/CSC release, you are not required to re-assemble your MVS/CSC user exits.

---

## Reverse Migration From MVS/CSC Release 7.0 to Release x.x

Perform the following steps:

- Make the MVS/CSC x.x LINKLIB available through a STEPLIB DD statement in the MVS/CSC started task procedure.
- Remove any parameters not supported by MVS/CSC x.x.
- Start MVS/CSC x.x.

---

## Coexistence

For MVS/CSC, the term “coexistence” implies the ability for two different MVS/CSC releases to execute on the same host at the same time.

An MVS/CSC subsystem cannot be started on an MVS host when another MVS/CSC subsystem at a different release level is active on that host.

---

## Compatibility With SMC

MVS/CSC and SMC must be at the same release level when residing on the same host.



## 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
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.

**coupling facility channel** A high bandwidth fiber optic channel that provides the high-speed connectivity required for data sharing between a coupling facility and the central processor complexes directly attached to it.(I)

**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.



---

## 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 HTTP server 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 systems 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.

**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

- allocating
  - event-log data set, 33
  - TAPEREQ definition data set, 33
  - trace data set, 33
- ALOCTIME startup parameter
  - considerations when specifying, 106
  - description, 59

## B

- basic functions, MVS/CSC, 18

## C

- CANCEL command (MVS), 100
- COMM startup parameter, 60
- Common Library Services (CLS), 22
- common startup parameters, 44
- communications methods
  - defining SNA LU 6.2 as, 78
  - defining TCP/IP as, 76, 84
  - NCP requirements, 88
  - overview, 25
  - specifying INTERNET startup parameter for TCP/IP, 76, 84
  - specifying TCPNAME startup parameter for TCP/IP, 77, 84
  - VTAM definitions ("3270 BISYNC"), 86
- communications startup parameters, 59
- COMPRFX startup parameter, 44
- configuration checklist, 30
- configuration tasks
  - allocating event-log data set, 33
  - allocating TAPEREQ definition data set, 33
  - allocating trace data set, 33
  - defining device addresses, 31
  - defining esoteric names, 31
  - defining startup parameter file, 35
  - modifying MVS/CSC startup procedure, 36
- CONFIGV control statement, specifying values for, 92
- considerations, migrating, 38

- considerations, when specifying ALOCTIME, 106
- control statement
  - OPTION TITLE, 74
- control statements
  - CONFIGV for Configuration Verification utility, 92
- conventions
  - control statements, 15
  - typographic, 11

## D

- Data Facility Hierarchical Storage Manager (DFHSM), 23
- defining
  - device addresses, 31
  - esoteric names, 31
  - MVS/CSC startup parameter file, 35
- device preferencing, 28
- devices, mixed, 28
- drive exclusion, 28

## E

- ENQNAME startup parameter, 46
- environment
  - MVS/CSC, 18
- esoteric names
  - defining for MVS, 31
  - LIBDEV startup parameter, 47
- event log
  - LOG startup parameter, 49
- event-log data set, allocating, 33

## F

- Fault Analyzer for z/OS, interaction with MVS/CSC, 24
- FORCE command (MVS), 100

## H

- Host Software Component (HSC)
  - coexistence with MVS/CSC, 32
  - description, 22

## I

INTERNET startup parameter, 62

## L

LIBDEV startup parameter, 47

Library Control System (LCS)

software products, 21

UNIX-based, 22

VM-based, 22

LIBUNIT startup parameter, 48

local major node definition, 87

LOG startup parameter, 49

## M

media, mixed, 28

migration considerations, 38

mixed media and mixed devices, 28

MSGCASE startup parameter, 50

MVS operating system

commands

CANCEL, 100

FORCE, 100

START, 99

STOP, 100

MVS/CSC

basic functions, 18

coexistence with HSC, 32

configuration checklist, 30

configurations, 19

migration considerations, 38

operating environment, 18

preinitializing, 38, 98

starting, 38, 99

startup parameter file, defining, 35

startup parameters, 39

startup procedure, modifying, 36

stopping, 100

system interfaces, 19

## N

Network Control Program (NCP)

defining communications, 88

specifying VTAMLST data set, 86

## O

operating environment, 18

operations

preinitializing the MVS/CSC, 98

starting MVS/CSC, 99

stopping MVS/CSC, 100

OPTION TITLE control statement, 74

## P

parameters, startup, 39

PORT startup parameter, 63

preferencing, devices, 28

preinitializing MVS/CSC, 38, 98

## R

REQTIME startup parameter, 65

RETCOUNT startup parameter, 66

RETTIME startup parameter, 67

## S

SCRLABL startup parameter, 51

SERVER startup parameter, 52

side information file, creating, 81

SL8500 library, 28

SNA LU 6.2

creating side information file, 81

defining as communications method, 78

defining system base LU, 81

setting up APPC/MVS, 79

setting up VTAM, 79

starting APPC/MVS, 83

starting VTAM, 83

software products

Automated Cartridge System Library Software (ACSL), 22

Common Library Services (CLS), 22

Host Software Component (HSC) for VM, 22

Library Control System (LCS), 21

START command (MVS), 99

starting MVS/CSC, 38, 99

startup parameter file

defining for MVS/CSC, 35

description, 40

sample file for LS server, 42

startup parameters

ALOCTIME, 59

COMM, 60

common, 44

communications, 59

COMPREFX, 44

ENQNAME, 46

INTERNET, 62

LIBDEV, 47

LIBUNIT, 48

LOG, 49

MSGCASE, 50

PORT, 63

REQTIME, 65

RETCOUNT, 66

RETTIME, 67

SCRLABL, 51

SERVER, 52

SYMDESTN, 68

TCPNAME, 69

TRACDEST, 53

- TRACE, 54
- UNITMAP, 56
- USERDATA, 58
- VAPLNAM, 71
- startup procedure, 36
- STOP command (MVS), 100
- stopping MVS/CSC, 100
- Storage Management Component (SMC), 18, 28, 32
- SYMDESTN startup parameter, 68
- System Authorization Facility (SAF), 24
- system base LU, defining, 81

## T

- tape management system (TMS), 23
- TCP/IP
  - defining as communications method, 76, 84
  - specifying INTERNET startup parameter, 76, 84
  - specifying TCPNAME startup parameter, 77, 84
  - testing communications parameters for CLS, 89
- TCPNAME startup parameter, 69
- third party software
  - DFHSM, 23
  - interactions with MVS/CSC, 23
  - MIM, 23
  - SAF, 24
  - STAM, 23
  - tape management systems, 23
- TRACDEST startup parameter, 53
- trace data set, allocating, 33
- TRACE startup parameter, 54
- transport, LIBDEV startup parameter, 47

## U

- UNITMAP startup parameter, 56
- UNIX-based Library Control System (LCS), 22
- USERDATA startup parameter, 58

## V

- VAPLNAM startup parameter, 71
- VM-based Library Control System (LCS)
  - Common Library Services (CLS), 22
  - description, 22
  - Host Software Component (HSC), 22
  - verifying communications setup, 88
- VTAM
  - application major node definition, 86
  - defining during installation, 86
  - local major node definition, 87
  - NCP requirements, 88
  - starting for SNA LU 6.2, 83
  - testing CLS communications parameters, 89

