Client System Component for MVS

Configuration Guide

Release 5.1

313486702

Proprietary Information Statement

This document and its contents are proprietary to Storage Technology Corporation and may be used only under the terms of the product license or nondisclosure agreement. The information in this document, including any associated software program, may not be reproduced, disclosed or distributed in any manner without the written consent of Storage Technology Corporation.

Limitations on Warranties and Liability

This document neither extends nor creates warranties of any nature, expressed or implied. Storage Technology Corporation cannot accept any responsibility for your use of the information in this document or for your use of any associated software program. You are responsible for backing up your data. You should be careful to ensure that your use of the information complies with all applicable laws, rules, and regulations of the jurisdictions in which it is used.

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

Restrictive Rights

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227─7013 or subparagraphs (c) (1) and (2) of the Commercial Computer Software — Restricted Rights at 48 CFR 52.227─19, as applicable.

First Edition, December 2002
Part Number 313486702
EC 128601

This edition applies to Release 5.1 of the Client System Component for MVS (MVS/CSC) software. Information in this publication is subject to change. Comments concerning the contents of this manual should be directed to:

Storage Technology Corporation
Manager, Software Information Development
One StorageTek Drive
Louisville, Colorado 80028-5209

or

sid@stortek.com

Document Effectivity

EC Number	Date	Doc Kit Number	Type	Effectivity
128601	December, 2002	---	First Edition	This document applies to the Client System Component for MVS (MVS/CSC), Version 5.1.

Contents

Figures

Tables

What's New With This Release?

MVS/CSC 5.1 includes the following changes and enhancements:

- SMP/E APPLY and ACCEPT installation steps for MVS/CSC are now included in the NCSAPPLY and NCSACCPT sample members, respectively.
- The following ALTER operator command parameters and/or MVS/CSC startup parameters are no longer honored:

  - ALOCtime
  - GDGAll
  - JES3set
  - SMSAcsr
  - SMSMOD
  - UNITAff
  - UXPrms
  - X02sub
  - X08sub
  - XJ3sub.

  The SMC provides allocation functions. Refer to the *SMC Configuration and Administration Guide* for more information.

- The allocation and job processing components of the Trace operator command have been moved to the SMC TRACE command. Specifically, support has ended in the MVS/CSC for the ALLCdata parameter and the following component names:

  - AL (allocation enhancement)
  - JP (job processing)
  - J3 (JES3)

  Refer to the *SMC Configuration and Administration Guide* for more information.

- Volume ranges can be specified either as alphabetic or numeric ranges for all commands and utilities that allow ranges to be entered.
- Message changes, additions and deletions.

About this Guide

This guide describes configuration procedures for release 5.1 of the StorageTek Client System Component for MVS (MVS/CSC).

Intended Audience

Part 1, “MVS/CSC System Overview” is intended for all users of the MVS/CSC product.

Part 2, “MVS/CSC Configuration Planning” is intended for systems administrators or system programmers responsible for configuring the MVS/CSC environment.

Part 3, “MVS/CSC Communications in a Non-Sysplex Environment” is intended for system administrators or system programmers responsible for configuring communications between the MVS/CSC and Library Control System (LCS), or server, in a non-sysplex environment.

Part 4, “MVS/CSC Communications in a Sysplex Environment” is intended for system administrators or system programmers responsible for configuring communications between the MVS/CSC and LCS in a sysplex environment.

Part 5, “MVS/CSC Configuration Verification and Startup Procedures” is intended for system administrators responsible for verifying configuration and starting the MVS/CSC.

Part 6, “Appendices” is intended for system programmers and StorageTek Software Support personnel.

Reader’s Comments

We’d like to know what you think about this guide. E-mail your comments to Software Information Development directly. Our Internet address is:

```
sid@stortek.com
```

Be sure to include the number and title of the guide you are referencing.

About the Software

MVS/CSC Release 5.1 is supported by this guide.

How this Guide is Organized

This guide contains the following chapters and appendices:

Part 1. "MVS/CSC System Overview"

- **Chapter 1, "Introduction"** describes the features and functions provided by the MVS/CSC.

Part 2. "MVS/CSC Configuration Planning"

- **Chapter 2, "Configuring the MVS/CSC Environment"** describes MVS/CSC configuration tasks.
- **Chapter 3, "Defining MVS/CSC Startup Parameters"** describes startup parameters used during MVS/CSC initialization.
- **Chapter 4, "Configuring the MVS/CSC License Key"** describes configuration procedures for the StorageTek MVS/CSC license key.
- **Chapter 5, "Defining Mixed Media and Devices"** describes how to define mixed media and mixed device characteristics for the MVS-based and UNIX-based LCS.

Part 3. "MVS/CSC Communications in a Non-Sysplex Environment"

- **Chapter 6, "Configuring Communications with a Unix-Based LCS"** describes configuration procedures for communications between the MVS/CSC and a UNIX-Based LCS.
- **Chapter 7, "Setting Up Communications With a VM-Based LCS"** describes configuration procedures for communications between the MVS/CSC and a VM-Based LCS.

Part 4. "MVS/CSC Communications in a Sysplex Environment"

- **Chapter 8, "Configuring Communications In a Base or Parallel Sysplex Environment"** describes configuration procedures for communications between the MVS/CSC and an MVS-Based LCS in a base or parallel sysplex environment.

Part 5. "MVS/CSC Configuration Verification and Startup Procedures"

- **Chapter 9, "Verifying MVS/CSC Configuration"** describes the utility used to verify that the MVS/CSC environment is configured correctly.
- **Chapter 10, "Starting and Stopping the MVS/CSC"** describes MVS/CSC initialization and termination procedures.

Chapter 6. "Appendices"

- **Appendix A, "MVS/CSC Samples, Source Code Modules, and Macros"** describes the samples, source code modules, and macros included on the installation base tape.
- **Appendix B, "Updating the JES3 Initialization Deck"** provides information about JES3 initialization deck statements used to define cartridge transports.
- **Appendix C, "Considerations When Setting Network Timeout Parameters"** describes guidelines for setting MVS/CSC network timeout parameters.
- **Appendix D, "Third-Party Software Restrictions"** describes restrictions when using third-party software products.
- **Appendix E, "Gathering Diagnostic Materials"** describes diagnostic materials that might be requested by Software Support for problem resolution.
- **Appendix F, "Migration and Coexistence"** provides migration and coexistence guidelines.
- **Appendix G, "List of Abbreviations"** defines commonly used abbreviations associated with the MVS/CSC.

A glossary and index are also included.

Conventions Used in this Guide

Typographic

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

Symbols

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

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

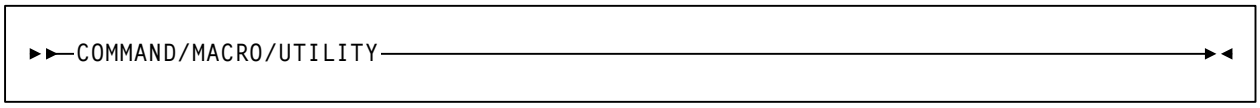
Warning: Information necessary to keep you from damaging your hardware or software.

Syntax Flow Diagrams

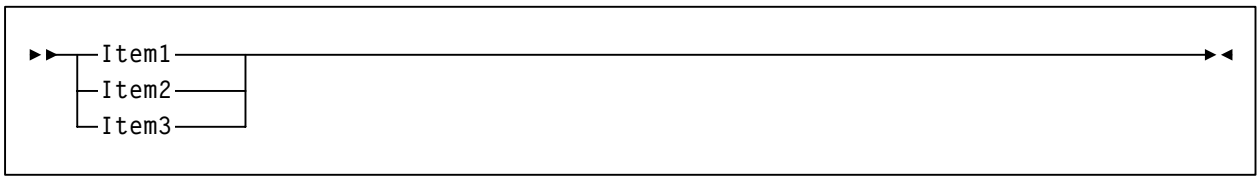
Syntax flow diagramming conventions include the following:

Flow Lines

Syntax diagrams consist of a horizontal base line, horizontal and vertical branch lines, and the text for a command, control statement, macro, or utility.



or

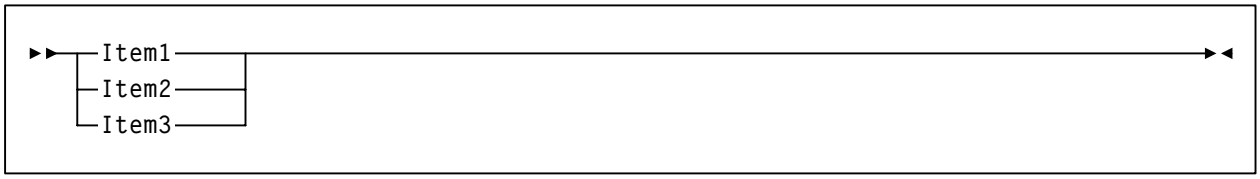


Diagrams are read left to right and top to bottom. Arrows indicate flow and direction.

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

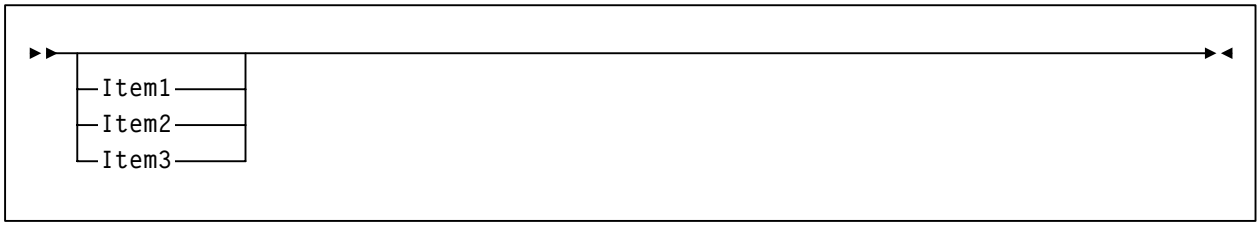
Single Required Choice

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



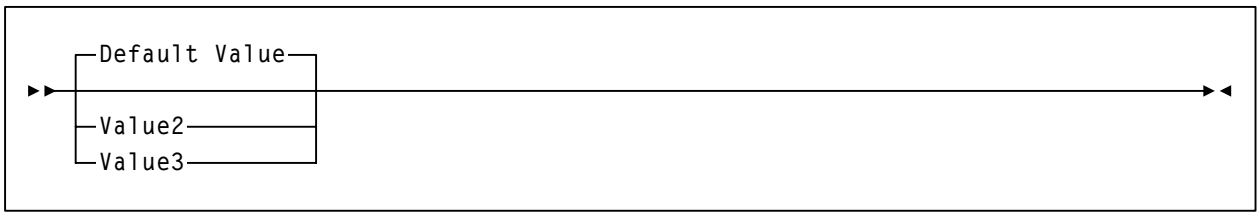
Single Optional Choice

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

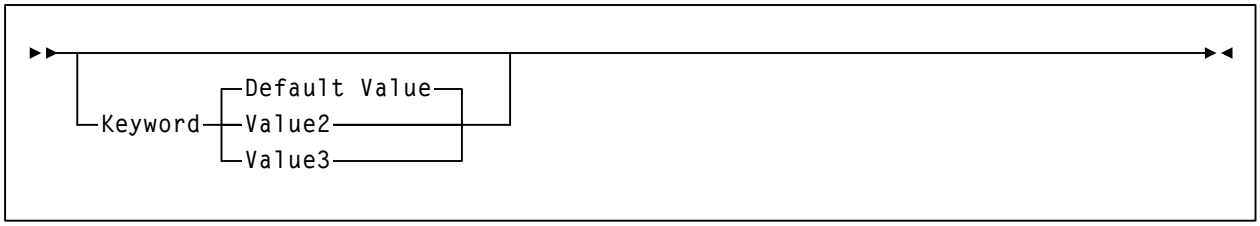


Defaults

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

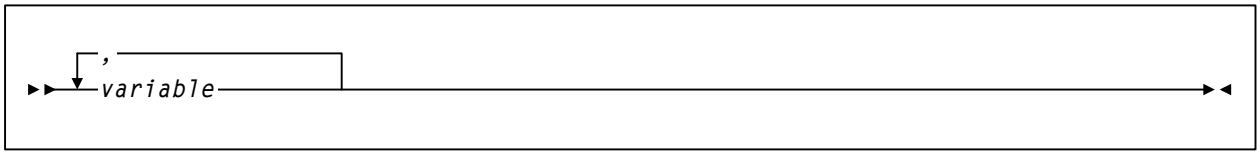


Some keyword parameters provide a choice of values in a stack. When the stack contains a default value, the keyword and the value choices are placed below the base line to indicate that they are optional, and the default value appears above the keyword line. In the following example, if the keyword is not specified with the command, keyword `(Default Value)` is used by the HSC.



Repeat Symbol

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



Keywords

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

Variables

Italic type is used to indicate a variable.

Alternatives

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

Delimiters

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

Ranges

- An inclusive range is indicated by a pair of elements of the same length and data type, joined by a dash. The first element must be strictly less than the second element.
- A hexadecimal range consists of a pair of hexadecimal numbers (for example, 0A2-0AD, or 000-0FC).
- A decimal range consists of a pair of decimal numbers (for example, 1-9, or 010-094). Leading zeros are not required. The decimal portion is referred to as an incremental range. The character positions of the incremental portion of both range elements must match, and the non-incremental characters of the first element must be identical to those of the second element.
- A numeric VOLSER range (*vol-range*) consists of a pair of VOLSER elements containing a decimal numeric portion of 1 to 6 digits (for example, ABC012-ABC025, or X123CB-X277CB). The decimal portion is referred to as an incremental range. The following additional restrictions apply:
  - The character positions of the incremental portion of both range elements must match.
  - The non-incremental characters of the first element must be identical to those of the second element.
  - You cannot increment two portions of a range element. If 111AAA is the first element, you cannot specify 112AAB for the second element.
  - If a VOLSER range contains more than one decimal portion, only the right-most portion is valid as the incremental range. For example:

    A00B00 the largest range that can be specified is A00B00 through A00B99.

    A0B0CC the largest range that can be specified is A0B0CC through A0B9CC.

    000XXX the largest range that can be specified is 000XXX through 999XXX.

- An alphabetic VOLSER range (*vol-range*) consists of a pair of VOLSER elements containing an incremental portion of 1 to 6 characters (for example, 000AAA-000ZZZ, or 9AAA55-9ZZZ55). This portion is referred to as an incremental range. The following additional restrictions apply:
  - The character positions of the incremental portion of both range elements must match.
  - The non-incremental characters of the first element must be identical to those of the second element.
  - You cannot increment two portions of a range element. If 111AAA is the first element, you cannot specify 112AAB for the second element.

- The alphabetic portion of the VOLSER range is defined as being from character A to Z. To increment multi-character sequences, each character increments to Z. For instance, ACZ is part of the AAA-AMM range. Examples are:

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

Blanks are not allowed between parameters and parentheses, or between parentheses and arguments. For example:

`LS C ID(3218)` **is a valid entry**.

`LS C ID ( 3218 )` **is not**.

Control Statements

The standard syntax conventions for control statements are as follows:

- The only valid control statement information area is from column 2 to column 72. Columns 73-80 are ignored.
- Parameters are 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.
- /* and */ can be used to enclose comments in the job stream. Comments cannot be nested.
- The maximum length for a control statement is 32,767 characters.

Related Publications

The following publications contain information about specific topics relating to the use of MVS/CSC.

StorageTek Nearline Control Solution (NCS) Publications

- *NCS (MVS/HSC, LibraryStation, MVS/CSC, SMC) Installation Guide*
- *Requesting Help from Software Support*

StorageTek Client System Component (MVS/CSC) Publications

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

StorageTek Storage Management Component (SMC) Publications

- *SMC Configuration and Administration Guide*

StorageTek Host Software Component (MVS/HSC) Publications

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

StorageTek LibraryStation Publications

- *LibraryStation Configuration Guide*
- *LibraryStation Operator and System Programmer's Guide*
- *LibraryStation Messages and Codes Guide*

StorageTek Virtual Storage Manager Publications

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

StorageTek Automated Cartridge System Library Software (ACSLS) Publications for the UNIX-Based LCS

- *ACSLS Installation and Services Manual*
- *ACSLS Programmer's Guide*
- *ACSLS System Administrator's Guide*

StorageTek Common Library Services (CLS) Publications for the VM-Based LCS

- *CLS Installation Manual*
- *CLS Messages and Codes Manual*
- *CLS Reference Manual*
- *CLS Reference Summary Card*
- *CLS User's Guide*

Technical Support

StorageTek Software Support and the StorageTek Customer Resource Center (CRC) maintain information about known NCS Release 5.1 product updates. You can contact Software Support or access the CRC for the latest information available concerning product updates (i.e. documentation, PTFs, PUTs).

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

```
http://www.support.storagetek.com
```

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

Part 1. MVS/CSC System Overview

Chapter 1. Introduction

Overview

MVS/CSC provides client functions and communications between an MVS host and the Library Control System (LCS) or server residing on another MVS or non-MVS 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

This chapter summarizes the features and functions provided by MVS/CSC, including:

- MVS/CSC operating environment
- MVS/CSC basic functions
- MVS/CSC system interfaces
- MVS/CSC configurations
- IBM Sysplex support
- Dynamic server switching capability
- StorageTek product support
- 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 MVS-based and UNIX-based LCS

MVS/CSC Operating Environment

MVS/CSC runs on any processor that supports IBM MVS/ESA SP,[1] and runs in an IBM multi-processor environment. The MVS/CSC supports both MVS/ESA SP JES2 and MVS/ESA JES3 systems. Except for noted differences, the information in this document applies to both JES2 and JES3 environments.

In addition, references in this document 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.

Operating System Requirements

JES2 Environment	JES3 Environment
• MVS/ESA SP Version 5.2.2 or higher (including all OS/390 and z/OS versions) **Note:** if using TCP/IP, OS/390 version 2.7 or later is recommended	• MVS/ESA SP Version 5.2.2 or higher (including all OS/390 and z/OS versions) • JES3 Version 5.1.1 or higher (including all JES3 OS/390 and z/OS versions) **Note:** if using TCP/IP, OS/390 version 2.7 or later is recommended

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.

Note: The following functions, previously influenced by the MVS/CSC, are managed by the Storage Management Component (SMC):

- 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 *SMC Configuration and Administration Guide* 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.

1. However, if IBM has dropped support for a particular MVS/ESA SP level, then the MVS/CSC will no longer support that level. For newly announced IBM operating system levels, it is our intent to support each new level. Program Temporary Fixes (PTFs) might be available for IBM operating system levels that were not supported at the time of this version, or for products that become available after this version of the MVS/CSC. Contact StorageTek Software Support for information about the availability of PTFs for additional support. See the *Requesting Help from Software Support* guide for information about contacting StorageTek Software Support.

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)
- Cross-system coupling facility (XCF)

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 (MVS-based and 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[2] 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 *SMC Configuration and Administration Guide* 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
- MVS-based
- VM-based

2. 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 16 Library Storage Modules (LSMs).

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

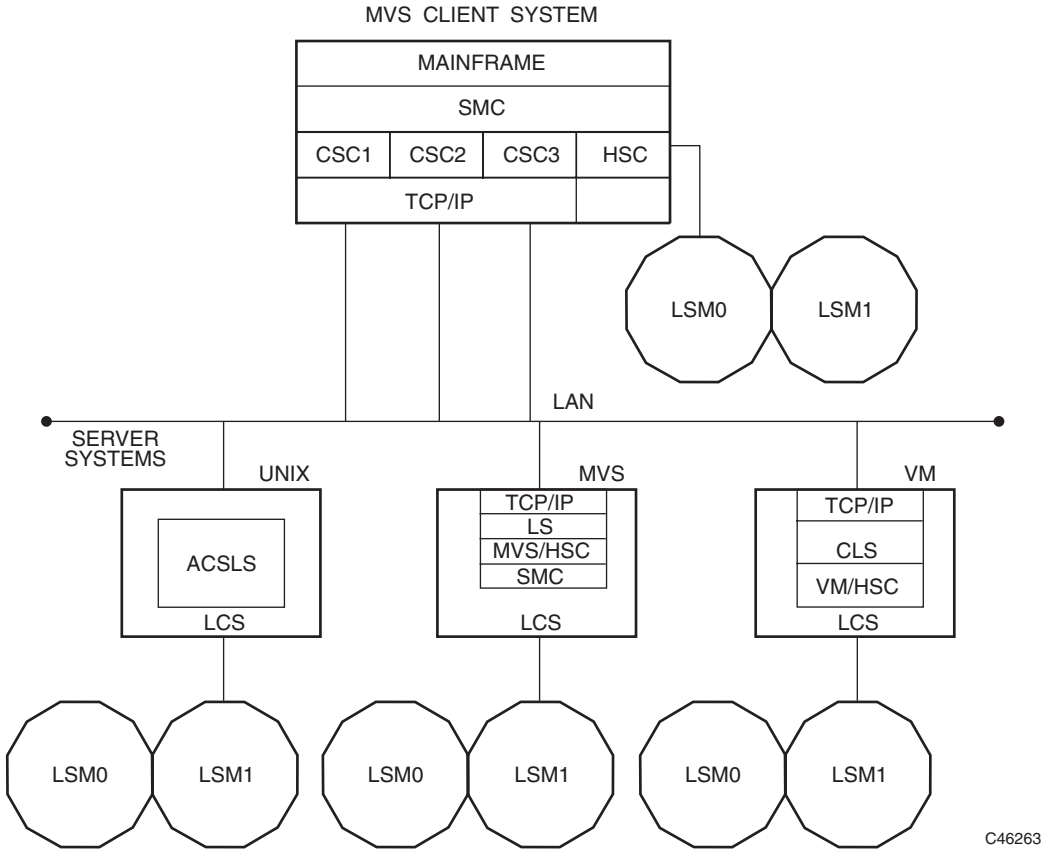


Figure 1. MVS/CSC-to-LCS Configuration

IBM Sysplex Support

The MVS/CSC supports the IBM sysplex (systems complex). A sysplex consists of multiple MVS systems cooperating to process work. In a parallel sysplex, applications that run on different MVS systems can simultaneously share data using the coupling facility. The cross-system coupling facility (XCF), SNA LU 6.2, and TCP/IP provide MVS communications for a sysplex environment.

In order to use XCF for communications between the MVS/CSC and the MVS-based LCS, the XCF group name and member name specified in the MVS/CSC startup parameters must match those defined to the MVS-based LCS.

In order to use SNA LU 6.2 for communications between the MVS/CSC and the MVS-based LCS, the partner LU specified in MVS/CSC's side information file must match the partner LU used to identify the LCS.

In order to use TCP/IP for communications between the MVS/CSC and the MVS-based LCS, you must specify the subsystem name or address space name of the TCP/IP stack, if the name was changed during the installation of the TCP/IP software.

Dynamic Server Switching

The MVS/CSC provides dynamic server switching support for multiple MVS-based LCSs that also support this capability. Dynamic server switching is supported only when the communications method is XCF or SNA LU 6.2. Dynamic server switching is not supported for the VM-based or UNIX-based LCS.

Dynamic server switching allows an MVS/CSC client to dynamically switch to an alternate LCS when it detects that the current LCS is unavailable. Dynamic server switching is initiated and controlled by the client system, and is configured using the MVS/CSC SRVRLIST startup parameter.

For each MVS/CSC client, an ordered server list is specified in the MVS/CSC SRVRLIST startup parameter. You can specify up to three MVS-based LCSs. The first LCS specified is considered to be the primary LCS. When the MVS/CSC detects that the current LCS is unavailable, the client dynamically switches connection to the next LCS specified in the list. When the alternate LCS no longer has requests outstanding, MVS/CSC periodically attempts to re-establish connection to the primary LCS.

Note: If you mount a cartridge from the current LCS before dynamic server switching occurs, you can dismount the cartridge from the new, alternate LCS.

StorageTek Library Product Support

The following sections lists the MVS/CSC Automated Cartridge Systems (ACSs), tape cartridge subsystems, and media types that the MVS/CSC supports.

StorageTek ACSs

4400 ACS	9360 stand-alone ACS	9740 ACS
• Library Storage Modules (LSMs)—4410 (standard), 9310 (PowderHorn), or 9360 (WolfCreek) • Library Control Units (LCUs) • Library Management Unit (LMUs)* • Cabling for LMUs to the LCUs and LCS	• LSMs—9360 (WolfCreek) • Integrated LMU*	• LSMs—9740 (TimberWolf)** • Integrated LMU*

* LMU Microcode Release 1.5.x or higher is required.
** LMU Microcode Release 1.6.x or higher is required for 9740 LSM and 9840 transport support.

StorageTek Tape Cartridge Subsystems

- 4480 Cartridge Subsystem (18-track)
- Silverton 4490 Cartridge Subsystem (36-track)
- TimberLine 9490 Cartridge Subsystem (36-track)
- TimberLine 9490EE Cartridge Subsystem (36-track)
- RedWood SD-3 Cartridge Subsystem (helical)
- 9840 Cartridge Subsystem
- T9840B Cartridge Subsystem
- T9940A Cartridge Subsystem

StorageTek Media Types

Standard capacity cartridge

This cartridge can be used on any longitudinal transport (i.e. 4480, 4490, 9490, or 9490EE).

Note: If data is written to the tape in 36-track mode, the data cannot be read by an 18-track 4480 transport.

Enhanced capacity cartridge (ECART)

This cartridge has a length of 1100 feet and can be used only on 36-track transports (i.e. 4490, 9490, and 9490EE). This cartridge is visually identified by a two-toned colored housing.

Extended-enhanced capacity cartridge (ZCART)

This cartridge can be used only on TimberLine 9490EE 36-track transports. These cartridges use a thinner media to provide twice the capacity of the ECART cartridge.

9840 cartridge

This cartridge can be used on 9840 or T9840B transports. Physically it is the same size as a standard 3480 cartridge, however, it has a 20 GB media capacity.

T9840B cartridge

This cartridge can be used only on T9840B transports. Physically it is the same size as a standard 3480 cartridge, however, it has a 20 GB media capacity.

T9940A cartridge

This cartridge can be used only on T9940A transports. It has a 60 GB media capacity. There are two types of T9940A cartridges: STK2P and STK2W.

Helical cartridge

This cartridge can be used only on RedWood (SD-3) transports. It is identified by the leader block on the left side of the cartridge. There are four types of helical cartridges: DD3A, DD3B, and DD3C, and DD3D.

In MVS-based and UNIX-based LCS environments, an ACS can contain mixed library transports (i.e. 4480, 4490, 9490, 9490EE, 9840, and SD-3 cartridge transports), and mixed media (i.e. standard, ECART, ZCART, 9840, and helical). In addition, 3480-, 3490E-, 3590, and helical-type cartridge transports can be attached to the MVS system outside the library.

StorageTek Library Control System (LCS) Software Products

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

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

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

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

- UNIX-based LCS, which consists of the Automated Cartridge System Library Software (ACSLS)
- MVS-based LCS, which consists of the Host Software Component for MVS (MVS/HSC) with LibraryStation
- 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: See the *NCS Installation Guide* for specific LCS software release levels.

UNIX-Based LCS

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

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

MVS-Based LCS

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

- MVS/HSC
- LibraryStation (a feature of MVS/HSC)

Host Software Component (HSC) controls the ACS. It runs as a subsystem on the MVS server system. The library database records cell status, characteristics, and disposition of all cartridges stored in the library.

LibraryStation is a software communications interface feature of the MVS/HSC; it resides on the MVS server system as a component of the MVS/HSC. LibraryStation provides software support and an interface for the Open Systems Nearline Network protocol. This includes an Open Network Computing Remote Procedure Call (ONC RPC 3.0) client, a Systems Network Architecture (SNA LU 6.2) client, an MVS cross-system coupling facility (XCF) client, and a TCP/IP client. Additionally, LibraryStation provides an operator command set for controlling LibraryStation operation through the MVS/HSC operator console.

The MVS-based LCS software can reside on an MVS processor running MVS/ESA SP. The MVS/CSC using the MVS-based LCS requires that the MVS/HSC, LibraryStation, and communications software be installed.

VM-based LCS

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

- Host Software Component for VM (VM/HSC)
- 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. The MVS/CSC should not prevent other third-party software from functioning. However, there are certain restrictions on using third-party software. See Appendix D, “Third-Party Software Restrictions” on page 171.

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 *SMC Configuration and Administration Guide* 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 D, "Third-Party Software Restrictions" on page 171 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.

Communications 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, UNIX-based, or MVS-based LCS. You can use the following software for TCP/IP communications:
  - IBM TCP/IP
  - Interlink TCPaccess
  - Interlink CISCO IOS
- 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 or MVS-based LCS
- Cross-system coupling facility (XCF) is used only by the MVS-based LCS for sysplex environments

Note: See the *NCS Installation Guide* for the supported communications 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.

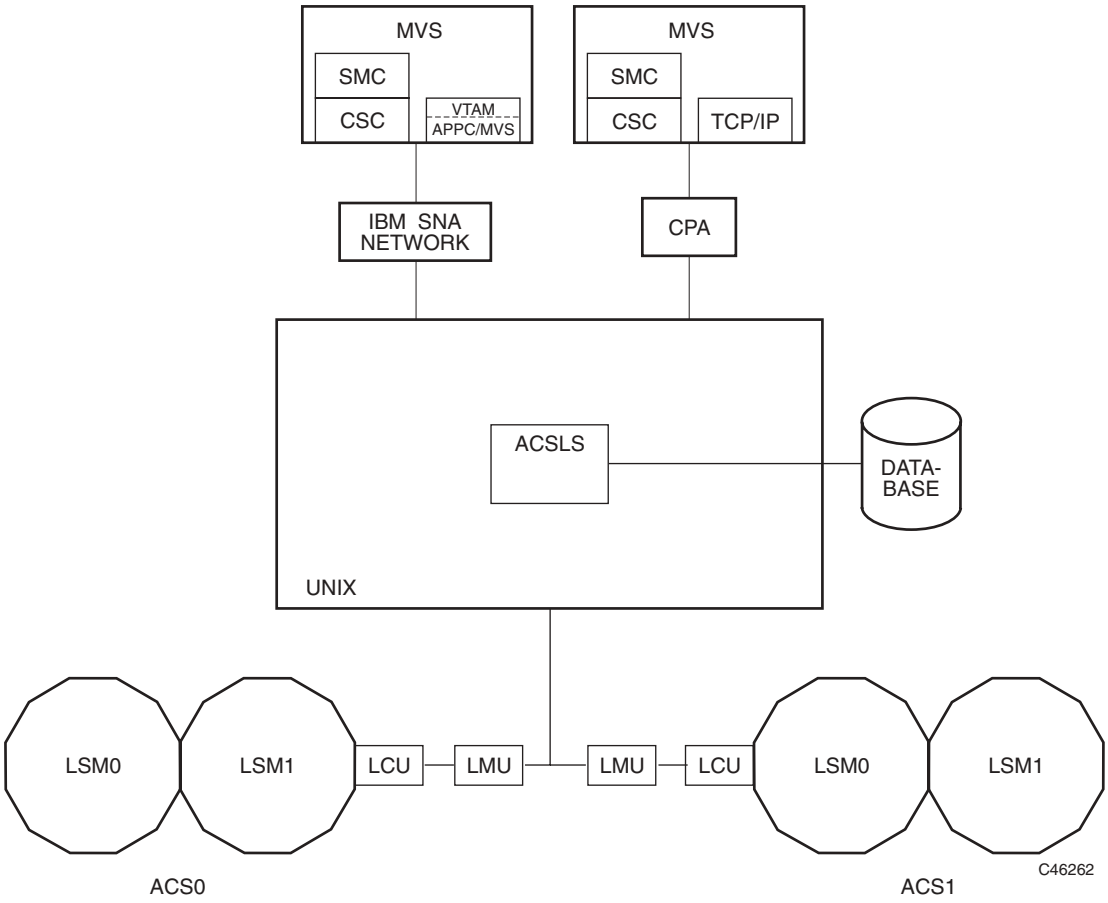


Figure 2. Communications Using TCP/IP and SNA LU6.2 (Unix-Based LCS)

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

Note: The data path is not shown in this illustration.

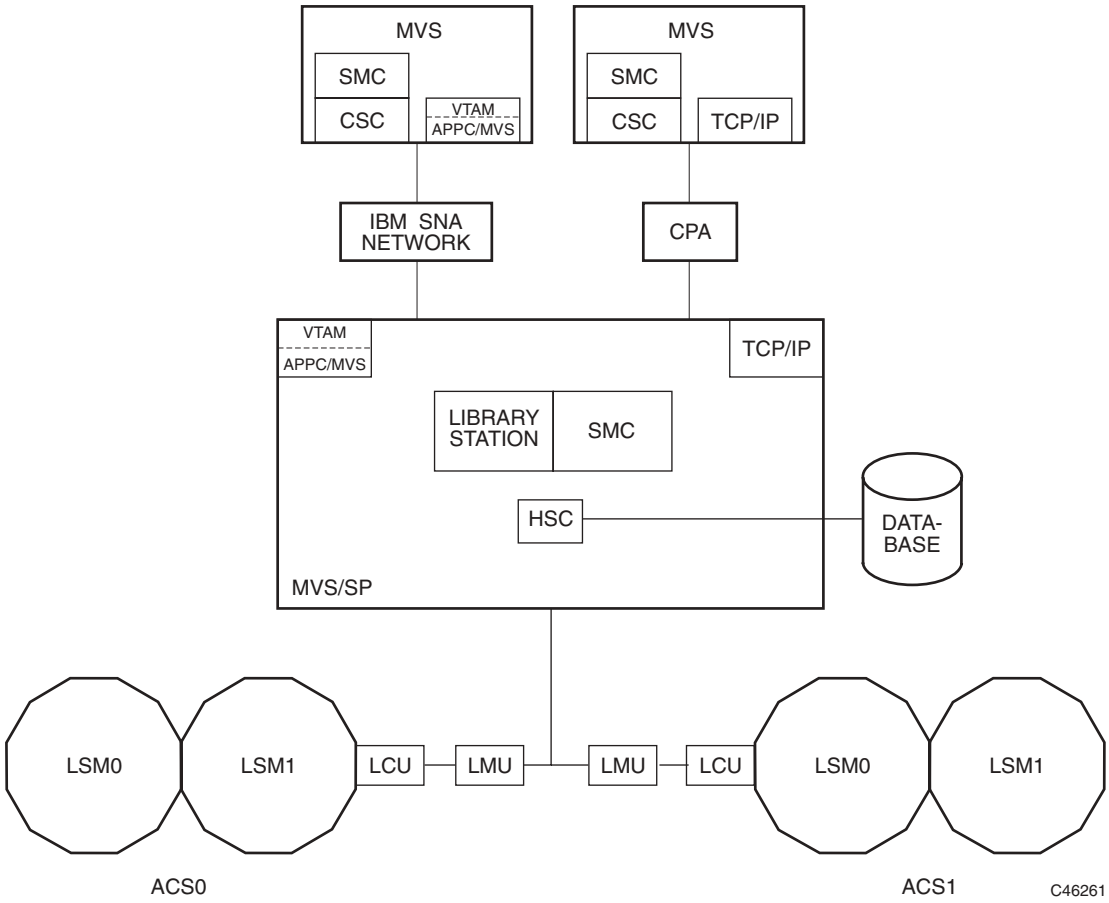


Figure 3. Communications Using TCP/IP and SNA LU 6.2 (MVS-Based LCS)

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.

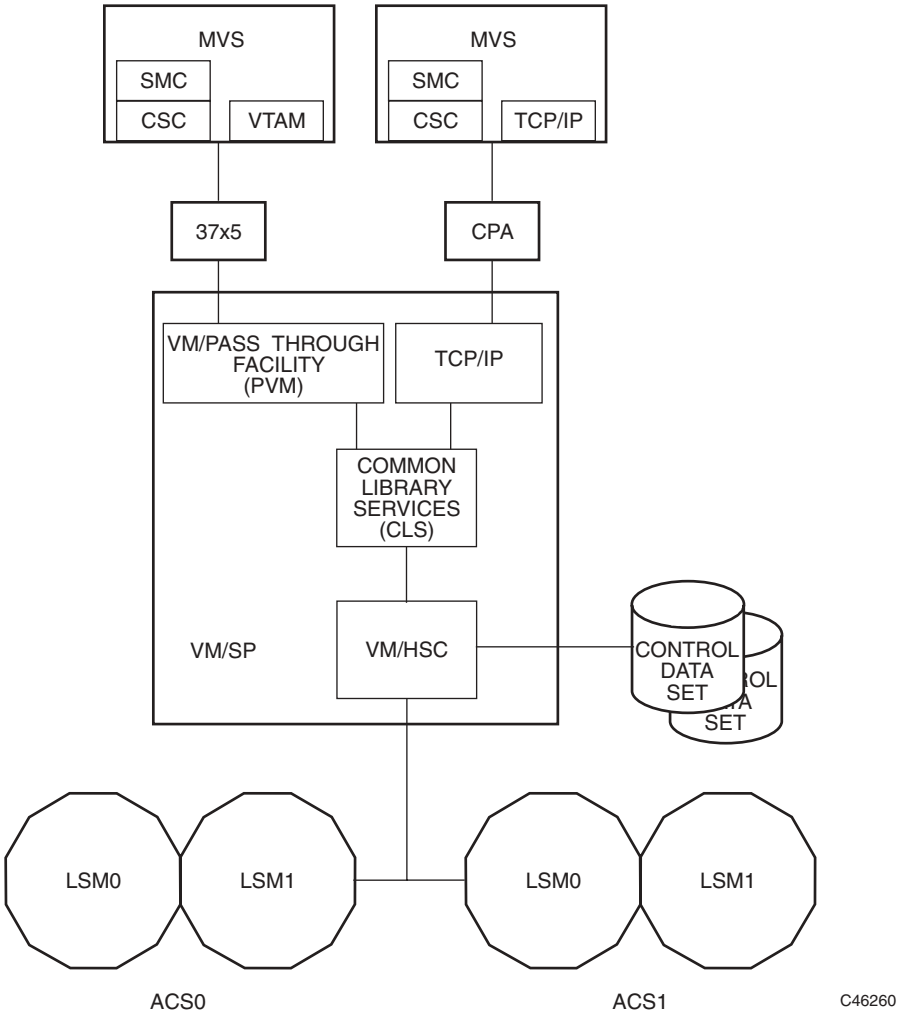


Figure 4. Communications Using TCP/IP and VTAM “3270 BISYNC” (VM-Based LCS)

The following figure shows the XCF communications protocol in a sysplex environment with either channel-to-channel (CTC) or coupling facility links for dynamic server switching.

Note: The data path is not shown in this illustration

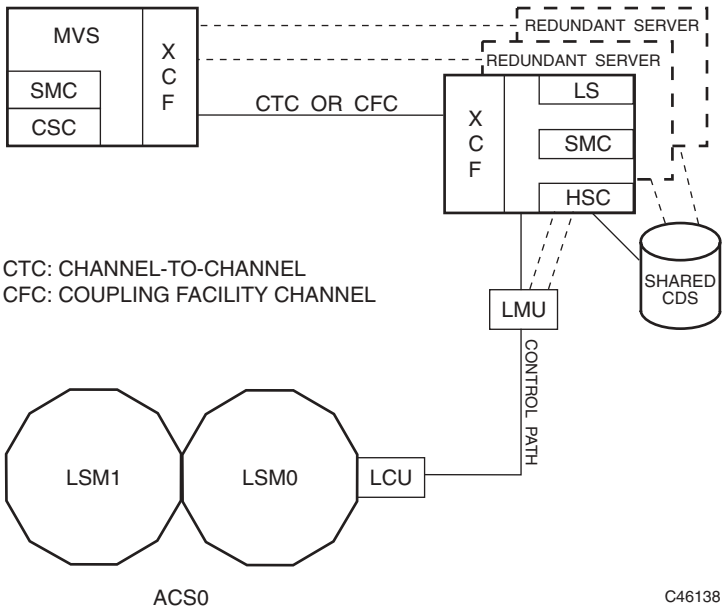


Figure 5. Communications Using XCF in a Sysplex Environment (MVS-Based LCS)

User Policy Definition for Mixed Media and Devices

The MVS/CSC supports mixed media (i.e., standard, ECART, ZCART, 9840, and helical) and mixed cartridge transport devices (i.e., 4480, 4490, 9490, 9490EE, 9840, and SD-3) in an ACS for the MVS-based and UNIX-based LCS. Mixed media and cartridge transport devices are not supported for the VM-based LCS.

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 *SMC Configuration and Administration Guide* for more information.

TAPEREQ control statements are used to specify tape request attributes to the MVS/CSC. 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.

You can define parameters for TAPEREQ control statements requesting that a data set be created using the following recording techniques:

- 18-track, thereby requiring that a 4480 transport be assigned to the request
- 36-track, thereby requiring that a 4490, 9490, or 9490EE transport be assigned to the request
- 9840, thereby requiring that a 9840 transport be assigned to the request
- T9940A, thereby requiring that a T9940A transport be assigned to the request
- Helical, thereby requiring that a SD-3 transport be assigned to the request

Device Preferencing

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

Part 2. MVS/CSC Configuration Planning

Chapter 2. Configuring the MVS/CSC Environment

Overview

Warning: MVS/CSC must be installed prior to performing MVS/CSC configuration tasks. See the *NCS Installation Guide* 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
- Configuring the MVS/CSC license key
- Pre-initializing and starting MVS/CSC

Note: "MVS/CSC Configuration Checklist" on page 24 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	25		
2	Update the JES3 Initialization Deck (JES3 environments)	163		
3	Review requirements for running MVS/CSC and MVS/HSC on the same MVS system	27		
4	Define tape request (TAPEREQ) attributes (optional)	27 & 93	TREQSAM1 & TREQSAM2	
5	Allocate the tape request (TAPEREQ) definition file (optional)	28		
6	Allocate event-log data set and trace data set (optional)	28		
7	Set up communications with a UNIX-based LCS (ACSLS server)	28 & 115		
8	Set up communications with a VM-based LCS (CLS server)	28 & 123		
9	Set up communications in a base or parallel sysplex environment (LibraryStation server(s))	28 & 133		
10	Define MVS/CSC startup parameter file	29 & 39	CSCPARM*x**	
11	Modify MVS/CSC startup procedure	30	CSCPROC	
12	Configure the MVS/CSC license key	87		
13	Verify MVS/CSC configuration	147	JCLCFGV1 & JCLCFGV2	
14	Pre-initialize and start the MVS/CSC	37 & 153		

* *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. In addition, you must define JES3-managed devices in the JES3 initialization deck if you are running the MVS/CSC in a JES3 environment with TAPE SETUP processing. See Appendix B, "Updating the JES3 Initialization Deck" on page 163 for information about defining JES3-managed devices.

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. non-library transports). The LIBDev and NONLib startup parameters are used to specify the esoteric names for each grouping. See Chapter 3, "Defining MVS/CSC Startup Parameters" on page 39 for information about the LIBDev and NONLib startup parameters.

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 non-library 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, 9840, 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 *VTCS Installation, Configuration and Administration Guide* 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 and MVS-based LCS. See Chapter 3, "Defining MVS/CSC Startup Parameters" on page 39 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. See *4400 Automated Cartridge System Common Library Services (CLS) User's Guide* for information about setting up a new version of the CLS configuration.

9840 Coexistence Requirements for MVS/CSC Release 3.1 or Higher with Release 3.0

9840 transports are not supported for MVS/CSC Release 3.0. If you are running both MVS/CSC Release 3.1 or higher and MVS/CSC Release 3.0 in the same library environment, all 9840 transports must remain offline to MVS systems running MVS/CSC Release 3.0. In addition, all MVS/CSC Release 3.0-defined esoterics for library and non-library transports must exclude 9840 transports. See the LIBDev startup parameter for information about specifying esoteric names for library transports. See the NONLib startup parameter for information about specifying esoteric names for non-library transports.

In order for 9840 transports to coexist in a library environment where both MVS/CSC Release 3.1 or higher and MVS/CSC Release 3.0 run, you must either define separate hardware configurations for each release, or define a different set of esoteric names for each release. The following sections give a brief description of each method.

Defining Separate Hardware Configurations

If you are running both MVS/CSC Release 3.1 or higher and MVS/CSC Release 3.0 in the same library environment, you can use the MVS Hardware Configuration Definition (HCD) facility to define separate hardware configurations for each release. This method allows you to use the same set of esoteric names across MVS/CSC releases.

Defining Different Esoteric Names

If you are running both MVS/CSC Release 3.1 or higher and MVS/CSC Release 3.0 in the same library environment, you can define different esoteric names for the transports that are supported for each release. This method allows you to define only one hardware configuration for both releases, but define different esoteric names for the transports supported for each release.

For example, you can define one esoteric name that maps to 9840 transports (MVS/CSC Release 3.1), and another esoteric name that excludes 9840 transports (MVS/CSC Release 3.0). You would then use the LIBDev startup parameter to specify the appropriate esoteric name for each ACS, and the NONLib startup parameter to specify the appropriate esoteric name for non-library transports.

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 30 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 first queries the MVS/HSC, followed by all active MVS/CSCs. Refer to the *SMC Configuration and Administration Guide* for details.

Scratch allocation requests can be directed to the MVS/HSC using HSC's Scratch Allocation User Exit (SLSUX02) for MVS allocation, or Scratch Allocation User Exit (SLSUX04) for JES3 allocation. Scratch allocation requests can be directed to the desired MVS/CSC using either Scratch Allocation User Exit (SCSUX02) for MVS allocation, or Scratch Allocation User Exit (SCSUX04) for JES3 allocation. See the *MVS/CSC System Programmer's Guide* for more information about MVS/CSC user exits.

Specific volume allocation requests can be directed to the MVS/HSC using HSC's Specific Volume Allocation User Exit (SLSUX08) for MVS allocation, or Specific Volume Allocation User Exit (SLSUX13) for JES3 allocation. Specific volume allocation requests can be directed to the desired MVS/CSC using either Specific Volume Allocation User Exit (SCSUX08) for MVS allocation, or Specific Volume Allocation User Exit (SCSUX13) for JES3 allocation. See the *MVS/CSC System Programmer's Guide* for more information about MVS/CSC user exits.

Defining Tape Request Attributes

Use the tape request (TAPEREQ) control statement to define the tape request characteristics for your data center to the MVS/CSC. The MVS/CSCC 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. See Chapter 5, "Defining Mixed Media and Devices" on page 93 for the parameters you can specify on the TAPEREQ control statement.

Allocating the TAPEREQ Definition Data Set

The TAPEREQ definition data set is used to store the TAPEREQ control statements that define the tape attributes of your data center to the MVS/CSC. 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 load the TAPEREQ definition data set using the TREQDEF startup parameter. In addition, you can dynamically load or reload the definition data set using the TREQDEF operator command.

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. See the *NCS Installation Guide* for information about DASD space and DCB parameter requirements. See 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, Interlink TCPaccess, or Interlink CISCO IOS for OS/390 is required for TCP/IP communications.)
- Virtual Telecommunications Access Method (VTAM "3270 BISYNC")
- SNA LU 6.2 (APPC and VTAM are required)
- Cross-system coupling facility (XCF)

You can set up communications in a non-sysplex environment, which includes communications between the MVS/CSC and the following LCSs:

- MVS-based (LibraryStation)
- UNIX-based (ACSLS)
- VM-based (CLS)

You can also set up communications in a sysplex environment, which includes communications between the MVS/CSC and MVS-based LCS in either a base sysplex or parallel sysplex environment.

See the following chapters for more information about setting up communications.

- Chapter 6, “Configuring Communications with a Unix-Based LCS” on page 115
- Chapter 7, “Setting Up Communications With a VM-Based LCS” on page 123
- Chapter 8, “Configuring Communications In a Base or Parallel Sysplex Environment” on page 133

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. See 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 (ACSLS) LCS using TCP/IP
- CSCPARM3—Sample startup parameter file for MVS-based (LibraryStation) LCS using TCP/IP
- CSCPARM4—Sample startup parameter file for MVS-based (LibraryStation) LCS using SNA LU 6.2
- CSCPARM5—Sample startup parameter file for UNIX-based (ACSLS) LCS using SNA LU 6.2
- CSCPARM6—Sample startup parameter file for MVS-based (LibraryStation) LCS using XCF

See Chapter 3, “Defining MVS/CSC Startup Parameters” on page 39 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 6 on page 31 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.

Notes:

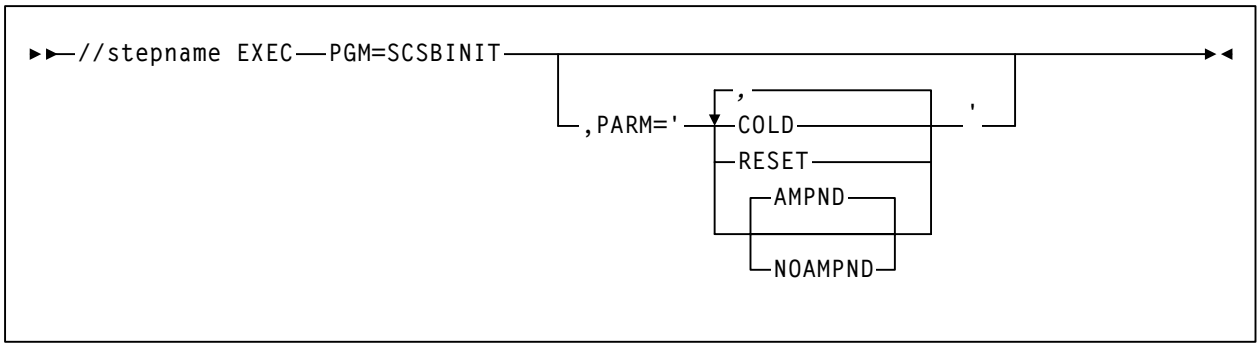
- The name of the member that contains the MVS/CSC JCL startup procedure must match the subsystem name defined in the IEFSSNyy member of SYS1.PARMLIB. For example, since CSC0 is the member name for the JCL startup procedure in Figure 6, the subsystem name defined in the IEFSSNyy member must also be CSC0.
- 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 6) is required. If you omit the STEPLIB DD statement in the started task procedure, unpredictable MVS/CSC operations may occur.
- If you are using the TCP/IP stack for either Interlink TCPaccess or Interlink CISCO IOS for TCP/IP communications with an MVS-based or 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 6). In addition, the CSLLINK data set is required for all TCP/IP communications (IBM TCP/IP, Interlink TCPaccess, and Interlink CISCO IOS for OS/390).
- If you installed the MVS/CSC into the HSC and LibraryStation CSI, all references to SACLINK must point to the LibraryStation’s SACLINK data set.
- StorageTek recommends a region size of 4M. 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.SCSLINK
//          DD DISP=SHR,DSN=your.TCPLINK /* optional for Interlink 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=1ØØ4 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 6. Sample MVS/CSC Started Task Procedure

Automation of Pending Mounts During MVS/CSC Startup

During initialization, the MVS/CSC automates pending mounts for configured drives (those specified in the MVS/CSC LIBUnit startup parameter). This function can be controlled by specifying AMPND or NOAMPND in the PARM field of the MVS/CSC started task procedure EXEC statement. The syntax for the EXEC statement is:



PARM=

defines the list of parameters passed to the MVS/CSC initialization routine.

COLD

specifies that any permanent in-memory data structures previously allocated by MVS/CSC are reallocated and re-initialized.

RESET

specifies that all subsystem status flags in the MVS Subsystem Communications Vector Table (SSCVT) for the MVS/CSC are unconditionally reset.

AMPND

specifies that the MVS/CSC automate pending mounts for configured drives during initialization. This setting is the default and need not be entered.

Notes:

- Configured drives are defined as those listed in the MVS/CSC LIBUNIT startup parameter.
- SCRPOOL and TAPEREQ requests are not honored for pending mounts processed during initialization.
- Automation of pending mounts for Virtual Tape Drives (VTDs) is not supported.

NOAMPND

specifies that the MVS/CSC bypass automation of pending mounts during initialization.

Note: AMPND and NOAMPND are mutually exclusive. Specifying both results in messages SCS0004I and SCS0511I.

Examples

The following examples illustrate the possible effects of automating pending mounts during MVS/CSC initialization.

Notes:

- In all cases, if the NOAMPND parameter is specified on the MVS/CSC started task EXEC statement, all mounts are left pending and operator intervention is required.
- Allocation and job processing policies including TAPEREQ statements, user exits, and startup parameters, are not honored for pending mounts processed during initialization.

Example 1:

This example assumes the following scenario:

- Single MVS/CSC
- Single ACS
- Drives 580, 581, 585, 586, and 587 are attached to the ACS and configured to the MVS/CSC
- Drives A54, A56, and A57 are manual drives
- Volumes ABC123, ABC222, ABC555, and ABC999 reside in the ACS
- Volumes XYZ001 and XYZ444 reside in external racks adjacent to the manual drives

While the MVS/CSC is down, an operator starts four jobs requesting various mounts. The following tables list the mounts requested for each job, and the result when the MVS/CSC is started with the mounts pending.

JOB 1:

Mounts Requested	Result When MVS/CSC is started with pending mounts
SCRATCH volume on unit 580	Mount request is transmitted to the server. TAPEREQ and User Exit processing are not performed.
Volume ABC123 on unit 581	Mount request is transmitted to the server.
Volume ABC222 on unit 585	Mount request is transmitted to the server.

JOB 2:

Mounts Requested	Result When MVS/CSC is started with pending mounts
SCRATCH volume on unit A54	SCRATCH mount is left pending; operator action required.
Volume ABC999 on unit 586	Mount request is transmitted to the server.

JOB 3:

Mounts Requested	Result When MVS/CSC is started with pending mounts
Volume ABC555 on unit A56	Mount is left pending; operator action required. If the ACS is co-located with the manual drives, the volume may be removed from the ACS (EJECT) to allow the operator to complete the mount.
Volume XYZ444 on unit A57	Mount is left pending; operator action required.

JOB 4:

Mounts Requested	Result When MVS/CSC is started with pending mounts
Volume XYZ001 on unit 587	Mount request is transmitted to the server. However, because XYZ001 is not resident in the ACS, the MVS/CSC issues message SCS0918D. If XYZ001 is available at the ACS location, it may be placed in the ACS (ENTER) and you can reply “R” to message SCS0918D.

Notes:

- ENTER and EJECT commands are not supported by the MVS/CSC. These commands must be issued from the LCS host (HSC, ACSLS, VM/CLS).
- When a cartridge’s location does not allow movement to the proper drive, ENTER or EJECT is not an option, or a scratch request cannot be satisfied within the current ACS, the affected job(s) must be cancelled and restarted while the MVS/CSC is active.

Example 2:

This example assumes the following scenario:

- Multiple MVS/CSCs (2)
- Multiple ACSs (3)
- Drives 580, 581, 585, and 586 are attached to ACS0 and configured to MVS/CSC0
- Drives 680 and 682 are attached to ACS1 and configured to MVS/CSC0.
- Drives 911, 912, and 913 are attached to ACS0 and configured to MVS/CSC1
- Drives A54 and A56 are manual drives
- Volumes ABC123 and ABC999 reside in ACS0, controlled by MVS/CSC0
- Volumes GHP045, GHP511, GHP222, and GHP595 reside in ACS1, controlled by MVS/CSC0
- Volumes MNA444 and MNA654 reside in ACS0, controlled by MVS/CSCb
- The customer uses the following CA-1 subpools (Each CA-1 subpool is mapped to a corresponding HSC subpool id)
    - SP0 Default
    - SP1 Manual Library
    - SP2 MVS/CSC0 ACS volume serial number range
    - SP3 MVS/CSC1 ACS volume serial number range

While both MVS/CSCs are down, an operator starts three jobs requesting various mounts. The following tables list the mounts requested for each job, and the result when the MVS/CSCs are started with the mounts pending.

JOB 1:

Mounts Requested	Result When MVS/CSCs are started with pending mounts
SCRATCH volume from CA-1 subpool 2 on unit 580	Mount request is transmitted to the server at MVS/CSC0 startup. TAPEREQ and User Exit processing are not performed.
Volume ABC123 on unit 581	Mount request is transmitted to the server at MVS/CSC0 startup
Volume GHP222 on unit 682	Mount request is transmitted to the server at MVS/CSC1 startup.
Volume MNA654 on unit 913	Mount request is transmitted to the server at MVS/CSC1 startup.

JOB 2:

Mounts Requested	Result When MVS/CSCs are started with pending mounts
SCRATCH volume from CA-1 subpool 1 on unit A54	Mount is left pending. Because SP1 is the subpool for manual mounts, an operator can select an available scratch and mount it.
Volume ABC999 on unit 586	Mount request is transmitted to the server at MVS/CSC0 startup
Volume GHP511 on unit 585	Mount request is transmitted to the server at MVS/CSC0 startup. However, because GHP511 resides in another ACS, MVS/CSC0 issues message SCS0918D. You must do the following: 1. Remove the volume from MVS/CSC0's ACS1 (EJECT) 2. Place the volume in MVS/CSC0's ACS0 (ENTER) 3. Reply "R" to message SCS0918D
Volume GHP045 on unit 680	Mount request is transmitted to the server at MVS/CSC0 startup.

JOB 3:

Mounts Requested	Result When MVS/CSCs are started with pending mounts
Volume GHP595 on unit A56	Mount is left pending; operator action required. If the ACS is co-located with the manual drives, the volume can be removed from the ACS (EJECT), allowing an operator to complete the mount.
SCRATCH volume from CA-1 subpool 2 on unit 911	Mount request is transmitted to the server at MVS/CSC1 startup. However, because there was no HSC subpool passed on the MRQST macro issued during MVS/CSC startup, the LCS attempts to satisfy the request from the default subpool. When the mount is completed, CA-1 rejects the volume and the mount is re-issued. This time, TAPEREQ and User Exit processing occur. The correct CA-1 subpool is selected, and is translated to an LCS subpool id. Because there are no volumes available from this subpool, the LCS returns an error. MVS/CSC issues message SCS0923E and the mount request is cancelled. Operator intervention is required.
Volume MNA444 on unit 912	Mount request is transmitted to the server at MVS/CSC1 startup.

Notes:

- ENTER and EJECT commands are not supported by the MVS/CSC. These commands must be issued from the LCS host (HSC, ACSLS, VM/CLS).
- When a cartridge's location does not allow movement to the proper drive, ENTER/EJECT is not an option, or a scratch request cannot be satisfied within the current ACS, the affected job(s) must be cancelled and restarted while the MVS/CSC is active.

Configuring the MVS/CSC License Key

MVS/CSC 5.1 requires a valid license key for initialization. Product license keys are validated during initialization and immediately after midnight each day. MVS/CSC will **not** initialize without a valid license key.

License keys can be obtained through the StorageTek Customer Resource Center (CRC) at **www.support.storagetek.com**, or by contacting your StorageTek Software Manufacturing Distribution Representative, Marketing Representative, or Systems Engineer. License Keys are generally issued within 48 hours of receipt of the request.

Visit the Customer Resource Center at the address listed above for more information about obtaining a license key.

Once a license key is assigned by StorageTek, you must make the license key information available to the MVS/CSC license key validation service. This is accomplished using the LKEYDEF startup parameter and LKEYINFO control statement. See Chapter 4, "Configuring the MVS/CSC License Key" on page 87 for more information.

Pre-Initializing and Starting MVS/CSC

The MVS/CSC must be initialized before the library can be accessed. The MVS/CSC is pre-initialized by the MVS subsystem pre-initialization routine during the initial program load (IPL) of the MVS host system. (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 subsystems. See Chapter 10, "Starting and Stopping the MVS/CSC" on page 153 for more information.

Migration Considerations

When migrating from a prior version of the MVS/CSC to this version of the MVS/CSC, you must either perform an IPL of the MVS host system, or issue a COLD start using the MVS START command. See Chapter 10, "Starting and Stopping the MVS/CSC" on page 153 for more information.

Chapter 3. 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.

Note: Virtual Storage Manager (VSM) support has been added for certain MVS/CSC startup parameters. Refer to the *VTCS Installation, Configuration and Administration Guide* for more information.

The following startup parameters are provided with MVS/CSC:

- ALOCTIME
- COMM
- COMPRFX
- DEFER
- DELDISP
- ENQNAME
- FETCH
- INTERNET
- LIBDEV
- LIBUNIT
- LKEYDEF (see **Chapter 4, "Configuring the MVS/CSC License Key"**)
- LOG
- MSGCASE
- NONLIB
- PORT
- REQTIME

- RETCOUNT
- RETTIME
- SCRLABL
- SERVER
- SYMDESTN
- SRVRLIST
- TCPNAME
- TRACDEST
- TRACE
- TREQDEF
- UNITMAP
- USERDATA
- VAPLNAM
- WTODESC
- XCFGROUP
- ZEROSCR

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:

- DEFER
- DELDISP
- FETCH
- MSGCASE
- TRACDEST
- WTODESC
- ZEROSCR

Note: See 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:

- DEFER
- DELDISP
- FETCH
- LKEYDEF (see **Chapter 4, "Configuring the MVS/CSC License Key"**)
- MSGCASE
- SMSACSR
- SMSMOD
- TRACDEST
- WTODESC
- ZEROSCR

Note: See 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
- LKEYDEF (see **Chapter 4, "Configuring the MVS/CSC License Key"**)
- SERVER

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

- INTERNET (required for TCP/IP communications)
- NONLIB (required if non-library devices exist)
- PORT (required for TCP/IP communications with a CLS server)
- SYMDESTN (required for SNA LU 6.2 communications with ACSLS or LibraryStation (LS) servers)
- SRVRLIST (required for cross-system coupling facility (XCF) communications; also required when specifying multiple MVS-based LCSs to eliminate an LCS as a single point of failure)
- UNITMAP (required for ACSLS or LibraryStation (LS) servers)
- VAPLNAM (required for VTAM "3270 BISYNC" communications)
- XCFGROUP (required for XCF 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 non-blank 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 xx.

Sample Startup Parameter File

The following figure shows a sample startup parameter file for an MVS-based LCS using TCP/IP communications. This file is in member CSCPARM3 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 MVS-BASED (LS) 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(LS)                                  /* SERVER TYPE */
/*
/* (TWO ACS LIBRARY, MVS/CSC DEVICES IN ACS Ø AND 1, AND NONLIB DEVICES)
/*
  LIBDEV(LIBØ,LIB1)                           /* LIBRARY ESOTERICS */
  NONLIB(NLIB)                                /* NON-LIB ESOTERIC */
  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 */
  DELDISP(NOSCRTCH)                           /* DELETE DISPOSITION */
  SCRLABL(SL)                                 /* SCRATCH LABEL TYPE */
  DEFER(NO)                                   /* DEFERRED MOUNTS */
  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 */
  TREQDEF(MY.TAPEREQ.FILE)                    /* TAPEREQ DEFINITIONS */
  LOG(RESET)                                  /* LOGGING OPTION */
  LKEYDEF(MY.LKEYINFO.FILE)                   /* LICENSE KEY INFO LOCATION */
/*********************************************************************
```

Figure 7. Sample Startup Parameter File for LibraryStation Server with TCP/IP Communications

Startup parameter files are also included for the following servers:

- MVS-based LCS using XCF
- MVS-based LCS using SNA LU 6.2
- UNIX-based LCS using TCP/IP
- 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 29 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 (see page 47).

Communications Startup Parameters
Startup parameters specific to communications setup and network timeout activities (see page 67).

Job Processing Startup Parameters
Startup parameters specific to job processing activities in both JES2 and JES3 environments (see page 83).

JES2 Startup Parameters
Startup parameters specific only to JES2 environments (see page 85).

JES3 Startup Parameters
Startup parameters specific only to JES3 environments (see page 86).

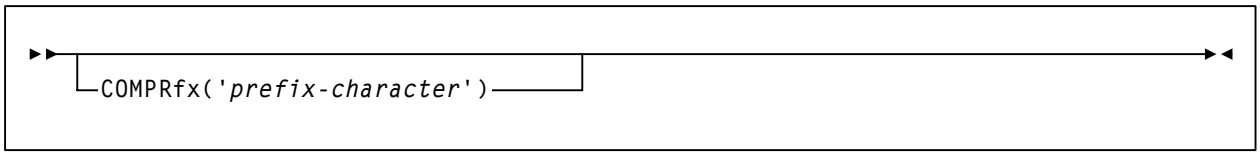
Common Startup Parameters

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

Example

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.

```
F CSC0 DISPLAY ALL
```

Notes:

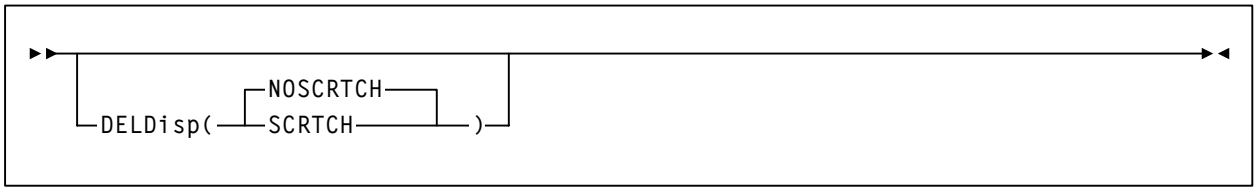
- The MVS/CSC subsystem name can be used even when a COMPRfx value has been specified.
- Each command prefix should be unique on the MVS system. The command prefixes specified for the HSC, LibraryStation, and MVS/CSC should each be unique in the MVS subsystem.

DELDISP Startup Parameter

This parameter specifies how the library is to interpret the delete disposition on a dismount message for a scratch cartridge. 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 by the ALTer command.

Syntax



Parameter Descriptions

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

SCRTCH
Specifies that the cartridge be returned to the scratch list when MVS indicates delete disposition.

NOSCRTCH
Specifies that the MVS delete disposition value is ignored. This is the default value.

Deleted cartridges are retained as non-scratch cartridges during the TMS and TLMS grace period. Therefore, users of TMS and TLMS should specify NOSCRTCH for the delete disposition.

Example

The following shows an example of the DELDisp parameter used to return the cartridge to the scratch list.

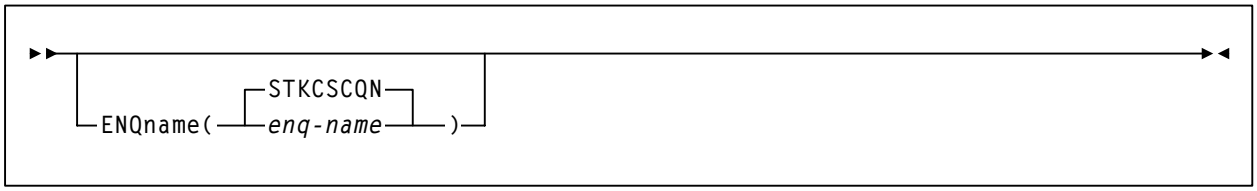
```
DELD(SCRTCH)
```

ENQNAME Startup Parameter

This parameter specifies the major ENQ name used by MVS/CSC for software serialization. This parameter is optional.

Note: The default major ENQ name is sufficient even in MVS systems using multiple MVS/CSC subsystems. ENQ name needs to be modified only if some other application is using the default major ENQ name.

Syntax



Parameter Descriptions

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

enq-name
Specifies the one- to eight-alphanumeric character name.

STKCSCQN
Specifies the default ENQ name.

Example

The following shows an example of the ENQname parameter used to specify CSCENQ as the major ENQ name used by MVS/CSC.

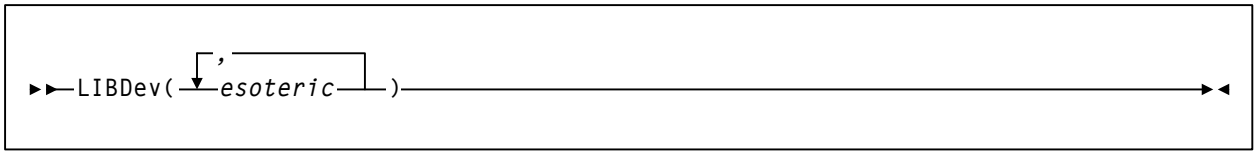
```
ENQ(CSCENQ)
```

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 TACS0 as the esoteric name for a library transport.

```
LIBD(TACS0)
```

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:

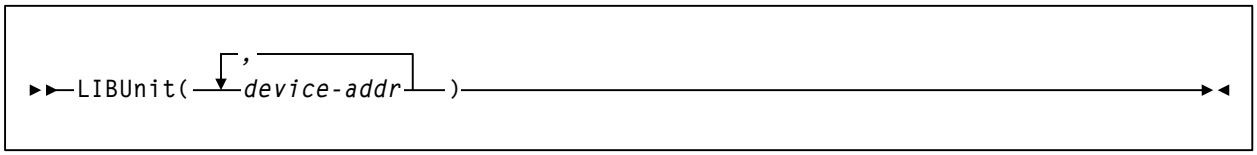
```
LIBD(TACS0,,TACS2)
```

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.

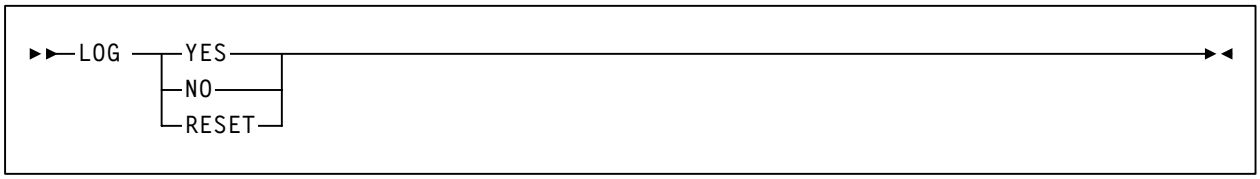
```
LIBU(A001,A002,A003,A004,A005,A006,A007)
```

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

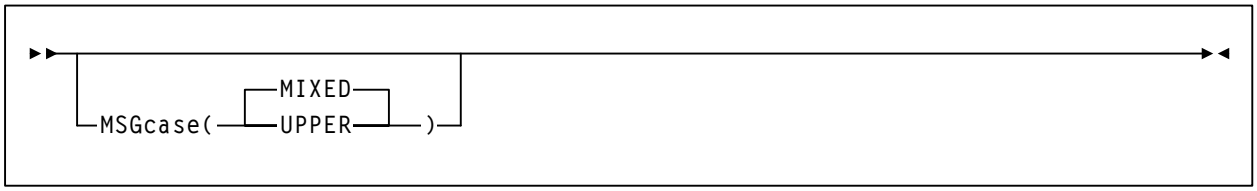
```
LOG(YES)
```

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

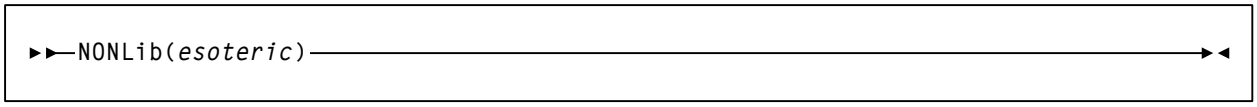
The following shows an example of the MSGcase parameter used to specify that all message output appears in upper-case format.

```
MSG(UPPER)
```

NONLIB Startup Parameter

This parameter specifies the esoteric name for transports outside the MVS/CSC-controlled library. This parameter is optional, but is required if non-library transports are used.

Syntax



Parameter Descriptions

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

esoteric

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

Example

The following shows an example of the NONLib parameter used to specify the esoteric name MANCART for a transport outside the MVS/CSC-controlled library.

```
NONL(MANCART)
```

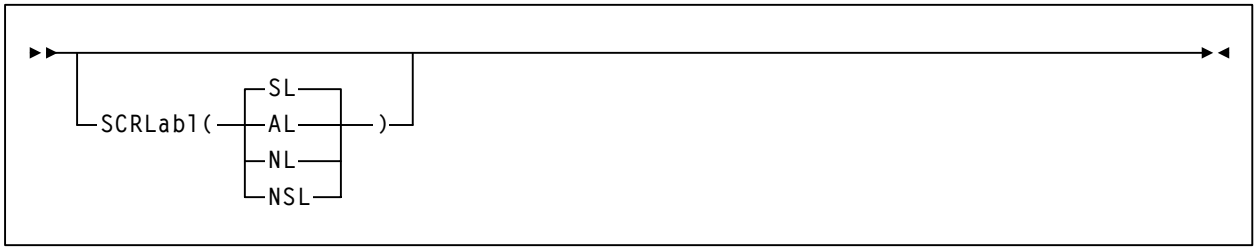
SCRLABL Startup Parameter

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

The MVS/CSC assumes that non-specific requests with other than the SCRLABL label type are outside the library. If a non-specific 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. See the *MVS/CSC System Programmer's Guide* 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 (non-labeled).

NSL
Specifies a non-standard label.

Example

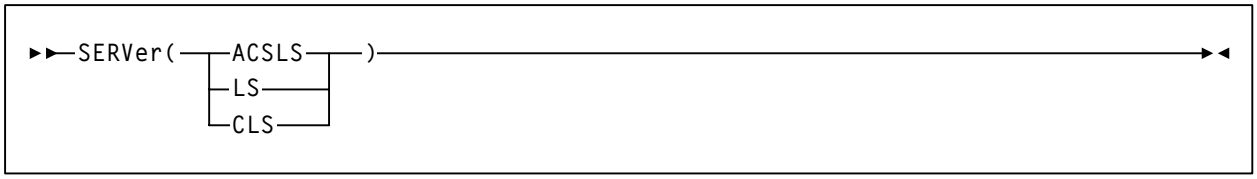
The following shows an example of the SCRLabl parameter used to specify the magnetic label type of a scratch volume that the MVS/CSC automates.

```
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.

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

LS
Specifies the connection to the LibraryStation server using ACSLS RPC connection-less protocol.

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

Example

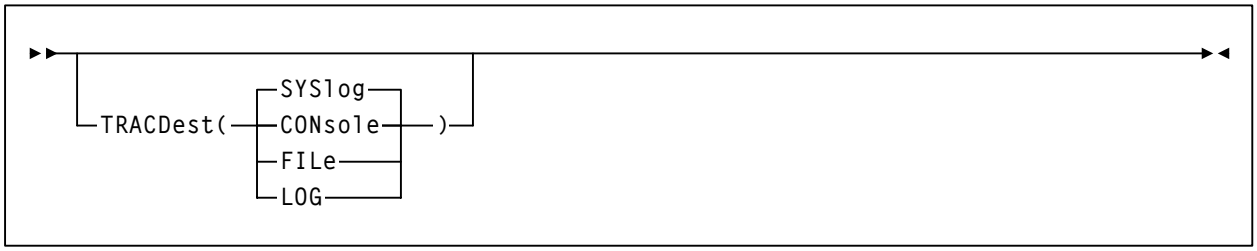
The following shows an example of the SERVer parameter used to specify ACSLS as the server protocol used during LCS communications.

```
SERVER(ACSLS)
```

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)
```

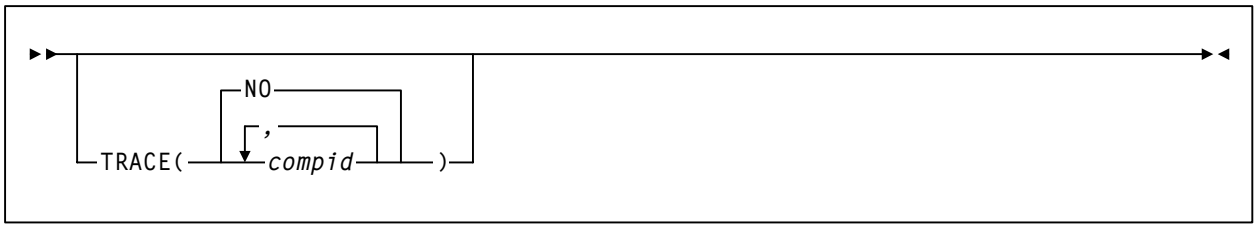
TRACE Startup Parameter

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

Notes:

- 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 a StorageTek System Support Representative (SSR).

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:

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

NO

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

Example

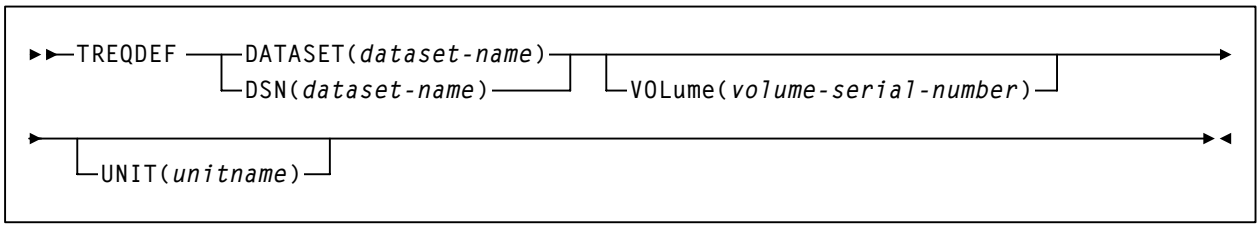
The following shows an example of the TRACE startup parameter used to specify tracing of operator commands.

```
TRACE(OC)
```

TREQDEF Startup Parameter

This parameter specifies the definition data set that contains tape request attributes; optionally, specifies the volume on which the data set resides. This parameter is required if you are loading tape request attributes. (See "TAPE REQUEST (TAPEREQ) Control Statement" on page 93 for information about the tape request attributes you can specify.)

Syntax



Parameter Descriptions

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

dataset-name

Specifies the name of the data set containing TAPEREQ control statements and optionally, the OPTion TITLE statement.

The definition data set can be a fixed length 80-byte sequential data set, or a fixed length 80-byte member of a PDS. If the definition data set is a member of a PDS, you must enclose the PDS and member name within single quotes.

volume-serial-number

Optionally, specifies the volume on which the data set resides. This parameter is required if the data set is not cataloged.

Valid characters for the volume serial number are A-Z, 0-9, #, $, @, and trailing blanks.

Notes:

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

Example

The following shows an example of specifying a cataloged sequential data set using the TREQDEF parameter.

```
TREQDEF(MYTREQ)
```

The following shows an example of specifying a cataloged PDS using the TREQDEF parameter.

```
TREQDEF('MYPDS1(TREQMEM1)')
```

The following shows an example of specifying an uncataloged PDS using the TREQDEF parameter.

```
TREQDEF('MYPDS2(TREQMEM2)',TOBA#1)
```

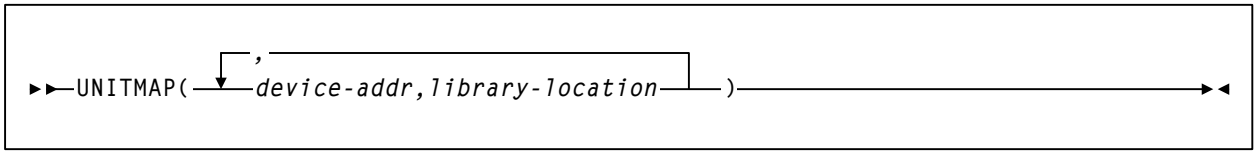
UNITMAP Startup Parameter

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 or LibraryStation (LS). 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 9840 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

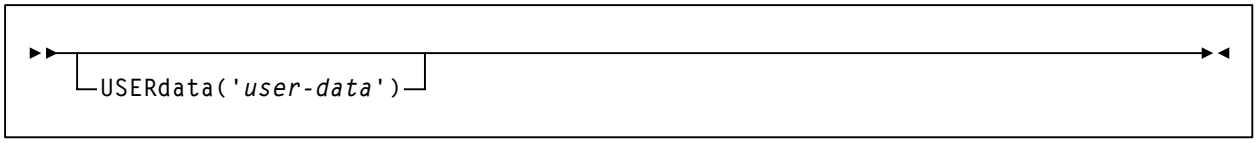
The following example shows the LIBUnit and UNITMAP startup parameters used to define the unit mapping for 9840 transports.

```
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

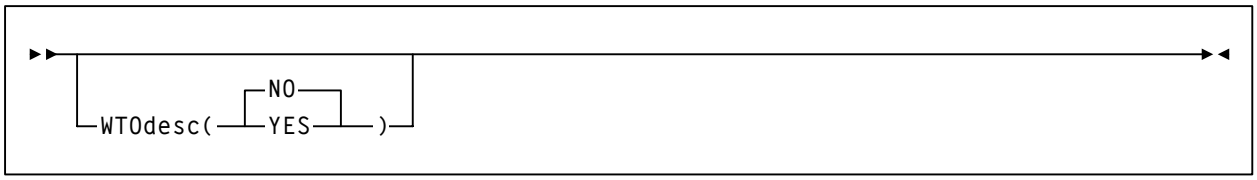
```
USER(‘JOB=CSCØ*,DSN=BKUP*’)
```

WTODESC Startup Parameter

This parameter defines whether mount and dismount messages for MVS/CSC-controlled cartridges remain highlighted on the MVS operator console during an automated mount or dismount. 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 WTOdesc startup parameter.

YES
Specifies that library mount and dismount messages remain highlighted.

NO
Specifies that library mount and dismount messages do not remain highlighted. This is the default value.

Example

The following example shows the WTOdesc parameter used to specify that mount or dismount messages remain highlighted on the MVS operator console.

```
WTO(YES)
```

Communications Startup Parameters

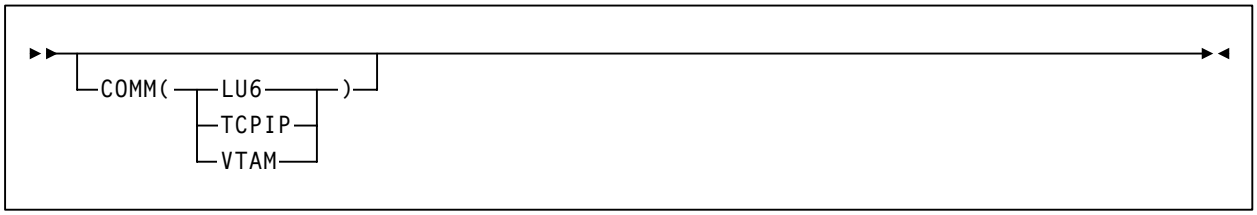
This section lists the MVS/CSC startup parameters that are specific to communications setup and network timeout activities.

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. You must use the SRVRList startup parameter to specify XCF as the communications method, or when specifying multiple MVS-based LCSs for dynamic server switching. Dynamic server switching is used to eliminate an LCS as a single point of failure (SNA LU 6.2 and XCF communications only).

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 an MVS-based LCS or UNIX-based LCS.

Note: If you want to specify multiple MVS-based LCSs to eliminate an LCS as a single point of failure, use the SRVRList startup parameter instead of the COMM startup parameter to define SNA LU 6.2 as the communications method. See "SRVRLIST Startup Parameter" on page 75 for more information.

TCPIP

Specifies that the communications method used is TCP/IP. This is the default value for the UNIX-based and MVS-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

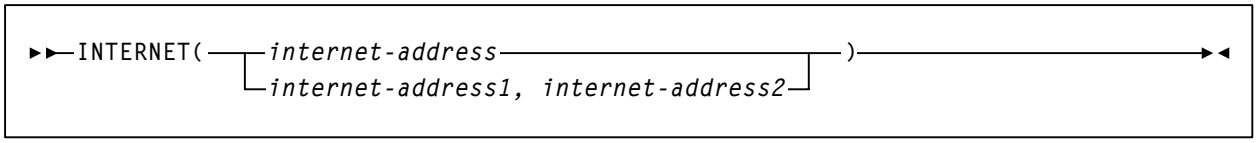
The following example shows how to specify SNA LU 6.2 as the communications method using the COMM parameter.

```
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)
```

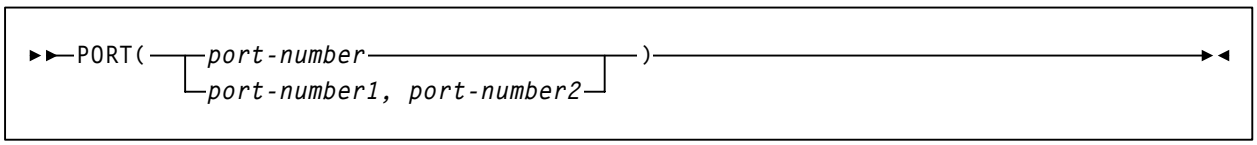
PORT Startup Parameter

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

Notes:

- 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



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.

Notes:

- 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.

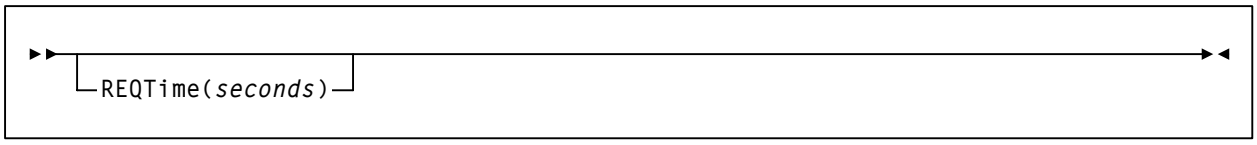
```
PORT(3000)
```

REQTIME Startup Parameter

This parameter defines the number of seconds that the MVS/CSC network interface waits for the ACSLS or LibraryStation 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 or LibraryStation. 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 166 for more information about setting values for this parameter. See "Considerations When Setting ALOCTIME and REQTIME" on page 169 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.

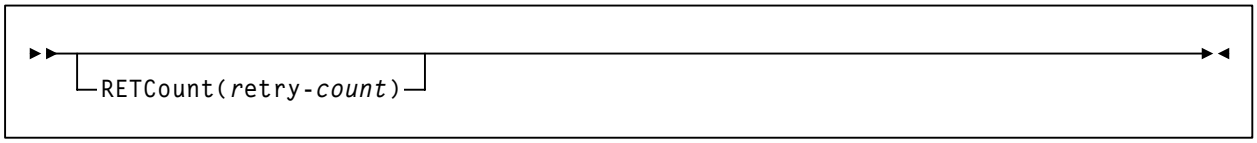
```
REQT(120)
```

RETCOUNT Startup Parameter

This parameter defines the number of times the MVS/CSC attempts to transmit a message before it determines that the ACSLS or LibraryStation server is not available.

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

Syntax



Parameter Descriptions

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

retry-count

Specifies the number of attempts (in decimal digits). Values can range from 0 to 999,999,999. The default value is 5.

This value multiplied by the value specified in the RETTime 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 166.

Example

The following shows an example of the RETCount parameter used to specify the number of times the MVS/CSC attempts to transmit a message.

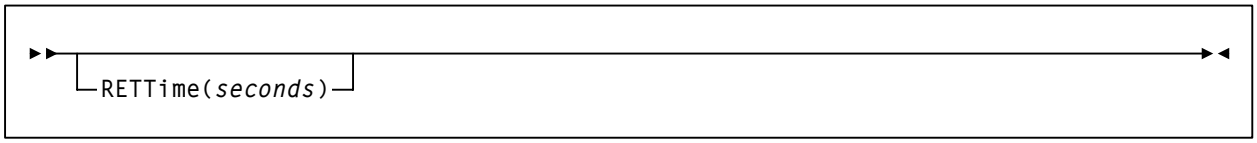
```
RETC(2)
```

RETTIME Startup Parameter

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

Note: This parameter is optional and is valid only when the communications method is TCP/IP and the server type is ACSLS or LibraryStation. 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 166.

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 or LibraryStation server request.

```
RETT(10)
```

SRVRLIST Startup Parameter

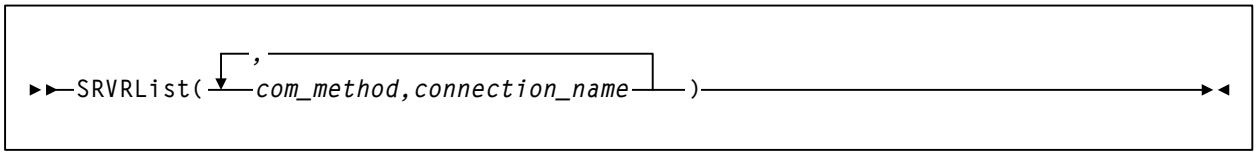
This parameter is used to specify the communications method used in a sysplex or MVS VTAM environment. This parameter is also used to specify an ordered server list to allow dynamic server switching for multiple MVS-based LCSs.

Dynamic server switching allows an MVS/CSC client to dynamically switch to an alternate MVS-based LCS when it detects that the current LCS is unavailable.

The first LCS specified in the SRVRList parameter is considered to be the primary LCS. When the MVS/CSC detects that this LCS is unavailable, the client dynamically switches connection to the next LCS specified in the list. You can specify up to three MVS-based LCSs to eliminate an LCS as a single point of failure.

Dynamic server switching is supported only for the MVS-based LCS when the communications method is XCF or SNA LU 6.2. Dynamic server switching is not supported for the VM-based or UNIX-based LCS, or when the communications method is TCP/IP for an MVS-based LCS.

Syntax



Notes:

- You must specify both the *com_method* and *connection_name* for each MVS-based LCS.
- You can specify a maximum of three MVS-based LCSs (*com_method,connection_name*) on the SRVRList startup parameter. Multiple specifications must be separated by a comma.

Parameter Descriptions

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

com_method

Specifies the communications method. You can specify the following communications methods.

XCF

Specifies that the communications method used is XCF. This value applies only to an MVS-based LCS within a sysplex environment.

You must also specify the XCF member name and the XCF group name used to establish communications between the MVS/CSC and the MVS-based LCS. The XCF member name corresponds to the *connection_name* value of the

SRVRList startup parameter. You use the XCFGROUP startup parameter to specify the XCF group name that is defined for the MVS-based LCS.

See the description for the *connection_name* value for more information about specifying the XCF member name. See "XCFGROUP Startup Parameter" on page 82 for information about specifying the group name.

LU6

Specifies that the communications method used is SNA LU 6.2. This value applies only to an MVS-based LCS.

You must also specify the associated symbolic destination name used to identify the LCS. The symbolic destination name corresponds to the *connection_name* value of the SRVRList startup parameter. See the description for the *connection_name* value for more information about specifying the symbolic destination name.

connection_name

Specifies one of the following values depending on the communications method specified.

xcf_member_name

For XCF communications, specifies the XCF member name for each MVS-based LCS. The XCF member name that you specify must match the XCF member name defined for the MVS-based LCS. The specification of the XCF member name and XCF group name establishes communications between the MVS/CSC and MVS-based LCS.

The XCF member name that you specify can consist of one to sixteen characters, including national characters: $, #, and @. Valid alphanumeric characters are 0-9 and upper case A-Z. See *IBM MVS/ESA Planning: Sysplex Management* for more information about defining XCF member names. See the *LibraryStation Configuration Guide* for the default MVS-based LCS member name.

symdestname

For SNA LU 6.2, specifies the associated symbolic destination name used to identify the LCS. 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 MVS-based LCS.

The symbolic destination 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.

You can also specify both XCF and SNA LU 6.2 as the communications methods to eliminate a communications link as a single point of failure. Multiple specifications must be separated by a comma.

Example

The following example shows how to specify dynamic server switching, and both XCF and SNA LU 6.2 as the communications methods using the SRVRList parameter.

```
SRVRL(XCF,xcf_member_name1,XCF,xcf_member_name2,LU6,symdestname)
```

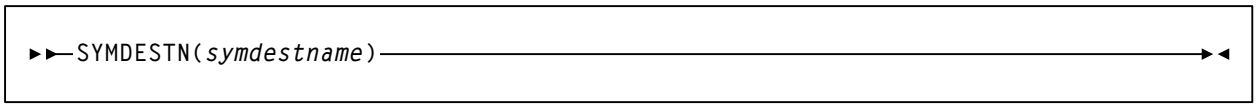
SYMDESTN Startup Parameter

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 (i.e. LibraryStation or ACSLS).

Notes:

- This parameter is required when the communications method is SNA LU 6.2.
- To specify multiple MVS-based LCSs to eliminate an LCS as a single point of failure, use the SRVRLIST startup parameter instead of the SYMDESTN startup parameter to specify the symbolic destination name. See "SRVRLIST Startup Parameter" on page 75 for more information.

Syntax



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)
```

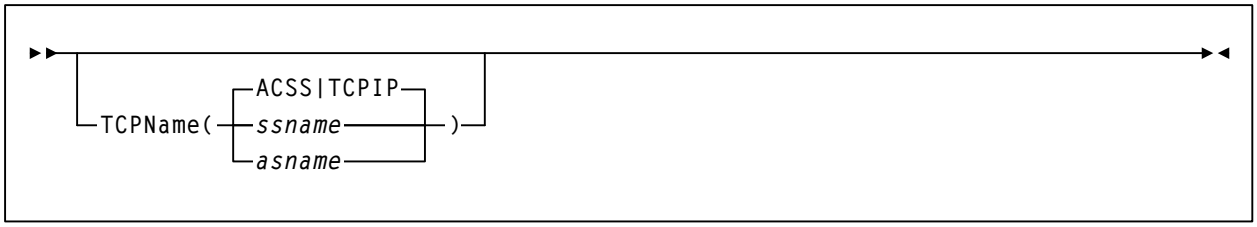
TCPNAME Startup Parameter

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 Interlink TCPaccess, Interlink CISCO/IOS for S/390, and IBM TCP/IP.

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 Interlink TCPaccess or Interlink CISCO/IOS for S/390.

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 Interlink TCPaccess.

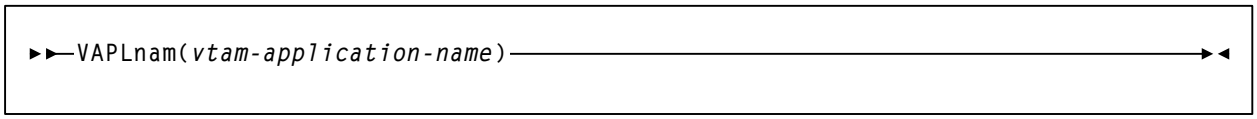
```
TCPN(TCP1)
```

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



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

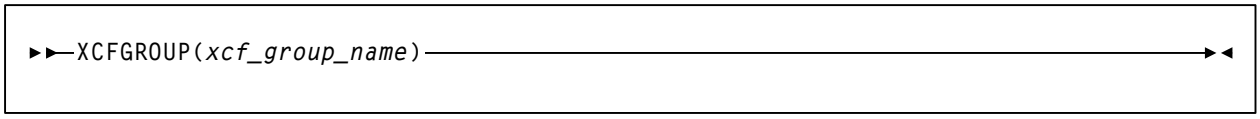
The following shows an example of the VAPLnam parameter used to specify VTAM0 as the VTAM application name.

```
VAPLNAM(VTAMØ)
```

XCFGROUP Startup Parameter

This parameter specifies the XCF group name. The XCF group name that you specify must match the XCF group name defined for the MVS-based LCS. You must specify this parameter when XCF is the communications method. See "SRVRLIST Startup Parameter" on page 75 for information about specifying XCF as the communications method.

Syntax



Parameter Descriptions

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

xcf_group_name
Specifies the XCF group name defined for the MVS-based LCS.

The XCF group name that you specify can consist of one to eight characters, including national characters: $, #, and @. Valid alphanumeric characters are 0-9 and upper case A-Z.

Note: To avoid duplicating IBM XCF group names, do not start group names with A-I or with the character string "SYS". In addition, the group name `UNDESIG` is reserved. See *IBM MVS/ESA Planning: Sysplex Management* for information about defining XCF group names. See the *LibraryStation Configuration Guide* for the default MVS-based LCS group name.

Example

The following shows an example for specifying an XCF group name.

```
XCFGROUP(SLGSTATN)
```

Job Processing Startup Parameters

This section lists the MVS/CSC startup parameters that are specific to SMC message handling in both JES2 and JES3 environments.

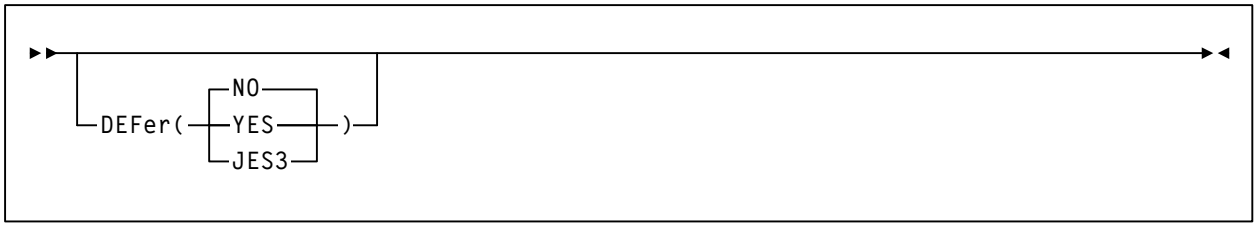
DEFER Startup Parameter

This parameter specifies whether or not to enable deferred mount processing for cartridges. When deferred mount processing is enabled, a cartridge is not mounted until a data set is opened; thus freeing the LSM and the transport to perform other work. However, if the data set is opened, the job waits until the cartridge is mounted. This parameter is optional.

Notes:

- 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 the ALTer command.
- The SMC IATUX09 JES3 user exit must be installed in order to defer mount messages during common allocation in JES3 environments. However, SMC automatically defers all library mounts for dynamic allocation requests regardless of the setting on the DEFer startup parameter during JES3 device allocation.
- If you are using the MVS IEFBR14 utility to uncatalog your data sets, the DEFer startup parameter should be set to YES.
- Values specified in the Defer Mount User Exit (SCSUX09) for MVS mount requests, and Defer Mount User Exit (SCSUX11) for JES3 mount requests override the DEFer startup parameter.
- DEFer startup parameter settings do not apply to virtual tape drives (VTDs). These mounts are automatically deferred.

Syntax



Parameter Descriptions

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

YES

Enables deferred mount processing for cartridges. The setting overrides the user's JCL and defers mounting cartridges until the data set is opened.

NO

Disables deferred mount processing for cartridges. During MVS allocation, the cartridge is mounted when a system initiator begins the step allocation process. During JES3 allocation, some or all of the cartridges required by a job are mounted before job execution.

NO is the default value for JES2 and JES3 environments without TAPE SETUP processing.

JES3

Enables JES3 deferred mount processing. Cartridges are mounted when a system initiator begins the step allocation process.

This parameter applies only to JES3 environments with TAPE SETUP processing. JES3 is the default value for these environments.

Example

The following shows an example of the DEFer parameter used to mount the cartridge when the data set is opened.

```
DEF(YES)
```

JES2 Startup Parameters

This section lists the MVS/CSC startup parameters that are specific to job processing in JES2 environments.

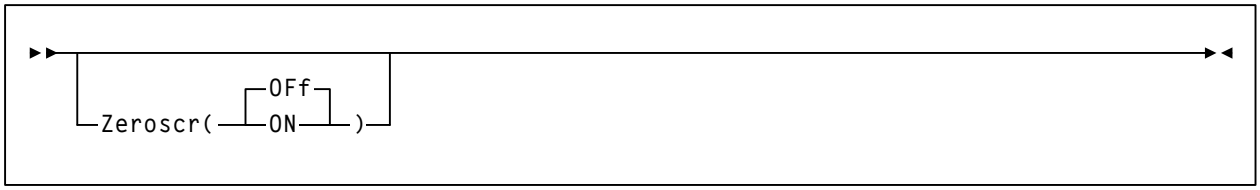
ZEROSCR Startup Parameter

This parameter defines whether devices in ACS(s) that contain valid scratch volumes are marked ineligible during allocation. This parameter is optional. In order for this parameter to take effect, user exit 02 (SCSUX02) must be active and return UX02LIB (return code 4) for the affected mount request.

Notes:

- This parameter is persistent. Once a value is specified, that value overrides any default value and remains in effect until explicitly re-specified by another initialization or ALTer command.
- The Zeroscr parameter is supported for JES2 and JES3 environments. See the *SMC Configuration and Administration Guide* for information on implemeting this parameter for JES3.

Syntax



Parameter Descriptions

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

ON

Specifies that devices in ACS(s) that do not contain valid scratch volumes are marked ineligible during allocation, even if a job requires allocation recovery.

OFf

Specifies that devices in ACS(s) that do not contain valid scratch volumes are not marked ineligible during allocation, even if this requires the ejection of a scratch cartridge from one ACS to be inserted into another ACS. This is the default value.

Example

The following example shows the Zeroscr parameter used to specify that devices in ACS(s) that do not contain valid scratch volumes are marked ineligible during allocation.

```
Z(ON)
```

JES3 Startup Parameters

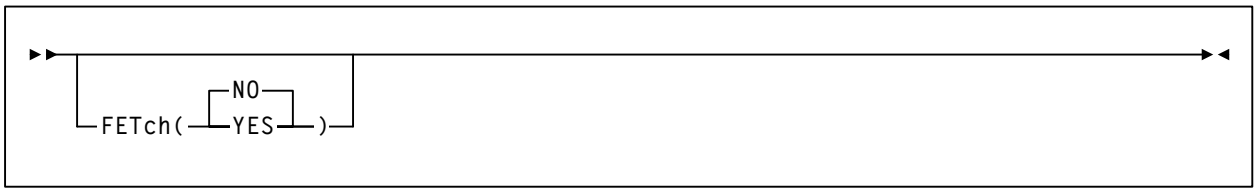
This section lists the MVS/CSC startup parameters that are specific to job processing in JES3 environments.

FETCH Startup Parameter

This parameter specifies whether or not JES3 operator fetch message IAT5210 is issued during library mount processing.

Note: The 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 by the ALTer command.

Syntax



Parameter Descriptions

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

YES
Specifies that JES3 operator fetch messages are issued for library mount processing.

NO
Specifies that JES3 operator fetch messages are suppressed for library mount processing. This is the default value.

Note: The MVS/CSC IATUX09 JES3 user exit **must** be installed in order to suppress operator fetch messages.

Example

The following shows an example of the FETch parameter used to specify the suppression of JES3 operator fetch messages during library mount processing.

```
FET(NO)
```

Chapter 4. Configuring the MVS/CSC License Key

Overview

MVS/CSC 5.1 requires a valid license key for initialization. Product license keys are validated during initialization and immediately after midnight each day. MVS/CSC will **not** initialize without a valid license key.

License keys can be obtained through the StorageTek Customer Resource Center (CRC) at **www.support.storagetek.com**, or by contacting your StorageTek Software Manufacturing Distribution Representative, Marketing Representative, or Systems Engineer. License Keys are generally issued within 48 hours of receipt of the request.

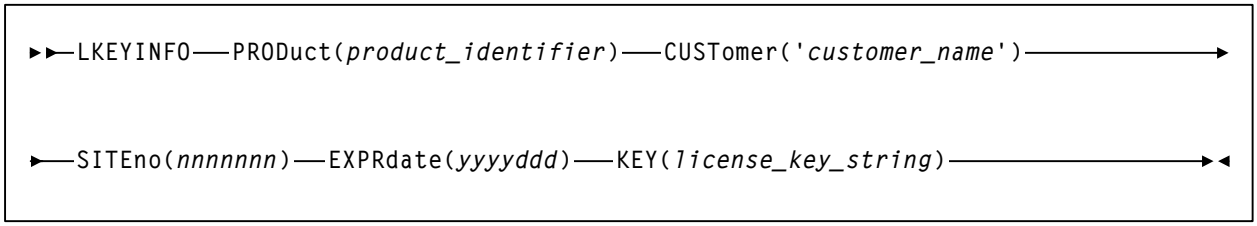
Visit the Customer Resource Center at the address listed above for more information about obtaining a license key.

Once a license key is assigned by StorageTek, you must make the license key information available to the MVS/CSC license key validation service. This is accomplished using the LKEYDEF startup parameter and LKEYINFO control statement described on the following pages.

LKEYINFO Control Statement

The LKEYINFO control statement is used to input license key information for the MVS/CSC. It is placed in a data set or Partitioned Data Set member identified by an LKEYDEF startup parameter, and must be present for the MVS/CSC to initialize.

Syntax



Parameter Descriptions

Note: All parameters must be entered exactly as received from StorageTek.

PRODuct
Specifies the product and release to which the license key applies.

product-identifier
Indicates the product identifier.

The product identifier consists of the MVS/CSC product abbreviation (CSC) followed by a four-character release id. i.e. CSC0500.

CUSTomer
Specifies the customer name as received from StorageTek. A maximum of 20 characters can be entered for the customer name.

customer-name
Indicates the customer name.

SITEno
Specifies the site number as received from StorageTek.

nnnnnnn
Indicates the site number.

EXPRdate
Specifies the expiration date of the license key as received from StorageTek.

yyyyddd
Indicates the expiration date.

KEY
Specifies the license key string as received from StorageTek.

license-key-string
Indicates the license key string.

Example

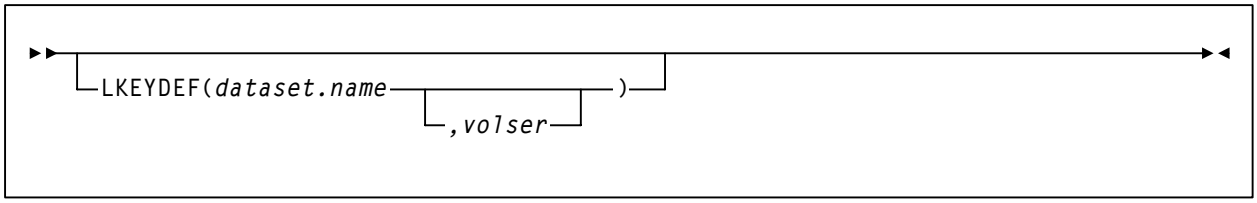
```
LKEYINFO PROD=(CSC0500) -
         CUST=('CUSTOMER NAME') -
         SITE=(111222) -
         EXPRD=(2001001) -
         KEY=(676767640XWV21)
```

LKEYDEF Startup Parameter

This startup parameter retrieves an LKEYINFO control statement containing MVS/CSC license key information, and loads it into an address space where it is available for retrieval by the MVS/CSC license key validation service.

This startup parameter **must** be present in the MVS/CSC startup parameter file prior to initialization.

Syntax



Parameter Descriptions

This section describes the values that can be specified with the LKEYDEF startup parameter.

dataset-name
Indicates the name of the data set containing the LKEYINFO control statement(s).

The definition data set can be a fixed length 80-byte sequential data set, or a fixed length 80-byte member of a PDS. If the definition data set is a member of a PDS, you must enclose the PDS and member name within single quotes.

volser
Indicates the volume serial number for the volume on which the data set resides.

Example

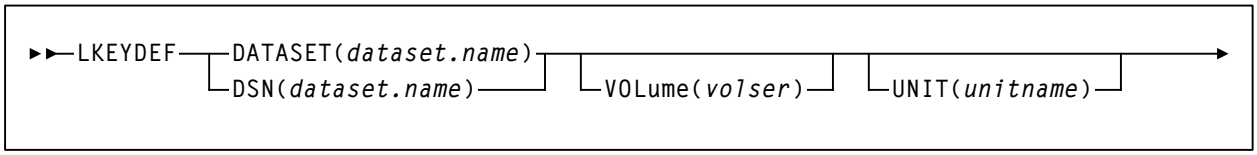
In the following example, the LKEYDEF startup parameter specifies that MVS/CSC license key information is retrieved from the data set MY.LKEYINFO.FILE.

```
LKEYDEF(MY.LKEYINFO.FILE)
```

LKEYDEF Operator Command

The LKEYDEF operator command retrieves LKEYINFO control statements containing license key information, and loads them into an address space where they are available for retrieval by the MVS/CSC. The LKEYDEF operator command overrides the LKEYDEF startup parameter settings.

Syntax



Parameter Descriptions

DATASET or DSN

Specifies the name of the data set containing the LKEYINFO commands.

dataset-name

Indicates the name of the data set.

The definition data set can be a fixed length 80-byte sequential data set, or a fixed length 80-byte member of a PDS. If the definition data set is a member of a PDS, you must enclose the PDS and member name within single quotes.

VOLume

Optionally, specifies the volume on which the data set resides. This parameter is required if the data set is not cataloged.

volser

Indicates the volume serial number.

UNIT

Optionally, specifies the unit where the definition data set resides.

unitname

Indicates the unit name. If the definition data set is not cataloged or this parameter is omitted, a unit name of SYSDA is the default.

HOSTID

Optionally, limits the execution of this command to the specified hosts.

host-id

Specifies the name of one or more hosts from which to execute this command.

Example

In the following example, the LKEYDEF operator command specifies that MVS/CSC license key information is retrieved from the data set MY.LKEYINFO.FILE.

```
LKEYDEF DSN(MY.LKEYINFO.FILE)
```

Chapter 5. Defining Mixed Media and Devices

Overview

The MVS/CSC uses the TAPEREQ and OPTion TITLE control statements to support mixed media and mixed devices. Mixed media and mixed devices are supported for the MVS-based and 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 tape request (TAPEREQ) control statement is used to define the tape request characteristics for your data center to the MVS/CSC. The MVS/CSC 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. The OPTion TITLE control statement, optionally defines an identifying string, which describes the definition data set that contains these definitions.

You use the TREQDEF startup parameter to load the definition data set. You can also use the TREQDEF operator command to dynamically load or reload the definition data set. See "TREQDEF Startup Parameter" on page 61 for information about the TREQDEF startup parameter. See the *MVS/CSC Operator's Guide* for information about the TREQDEF operator command.

The following sections describe the TAPEREQ and OPTion TITLE control statements.

TAPE REQUEST (TAPEREQ) Control Statement

Note: Virtual Storage Manager (VSM) support has been added for the TAPEREQ control statement. Refer to the *VTCS Installation, Configuration and Administration Guide* for more information.

Tape request attributes describe the media type and recording technique or model number used for a particular data set, or group of data sets. You use the TAPEREQ control statement to define tape request attributes to the MVS/CSC. TAPEREQ control statements are placed in the definition data set specified by the TREQDEF command. The MVS/CSC uses the information in the definition data set to ensure that the correct media type is used to satisfy a request, and that the cartridge is mounted on the appropriate device.

The definition data set can also contain an identifying string describing its definition data set contents (see “OPTION TITLE Control Statement” on page 112 for information about specifying an identifying string). You can use the TREQDEF startup parameter to load the definition data set, or use the TREQDEF operator command to dynamically load or reload the data set.

As the MVS/CSC processes each allocation request, it searches the TAPEREQ control statements in the order that they appear in the definition data set to determine the media type and recording technique or model number to assign to the request. Recording technique and model number are mutually exclusive.

The first statement that matches the request’s input criteria is used. Therefore, it is recommended that you order the TAPEREQ control statements from most specific to most general.

The TAPEREQ control statement parameters are divided into selection criteria (input) parameters, and media and format requirements (output or result) parameters. The selection criteria are used to locate the first TAPEREQ control statement that matches the request under consideration. The media and format requirements provide information that is used to modify the request. For example:

- For specific volume requests, the MVS/CSC modifies the MVS eligible device list (EDL), or the JES3 intermediate job summary table (IJS) during to exclude inappropriate devices (via recording technique or model number).
- For non-specific (scratch) requests, the MVS/CSC modifies the MVS EDL or JES3 IJS during drive exclusion to exclude inappropriate devices (via recording technique or model number), and excludes cartridges that are the wrong media type.
- During MVS/CSC processing, a media-type code is included in the mount-scratch request; the server uses this code to select the correct media type.

You can use wild card characters to identify collections of data sets with common characteristics in a single TAPEREQ control statement. For example, if all data sets that reside on enhanced capacity media are identified with an L as the first character of the data set name, then you can specify L*.** to represent all data sets whose names start with L in a single TAPEREQ control statement.

The TAPEREQ control statement is in essence an IF-THEN statement: IF the selection criteria parameters (ANDed together) match the current tape request variables, THEN the media and format requirements parameters apply.

If the MVS/CSC is unable to match the request with a media type value, the media type is determined from the RECtech as shown in Table 1 on page 101.

Disabling a TAPEREQ Definition

Use the following procedure to disable a TAPEREQ definition.

1. Edit the definition data set containing the current TAPEREQ control statements.
2. Delete the parameters of the TAPEREQ control statement you want to disable, or delete the TAPEREQ control statement.
3. Issue the TREQDEF operator command to reload the definition data set.

With the modified definition data set in effect, the MVS/CSC searches the remaining TAPEREQ control statements to determine the media type and recording technique or model number to assign to a request.

To disable all TAPEREQ definitions, load a definition data set that contains only one TAPEREQ control statement with no parameters. Refer to Table 1 on page 101 and Table 2 on page 103 for a list of TAPEREQ media and recording technique (or model) default values.

Syntax

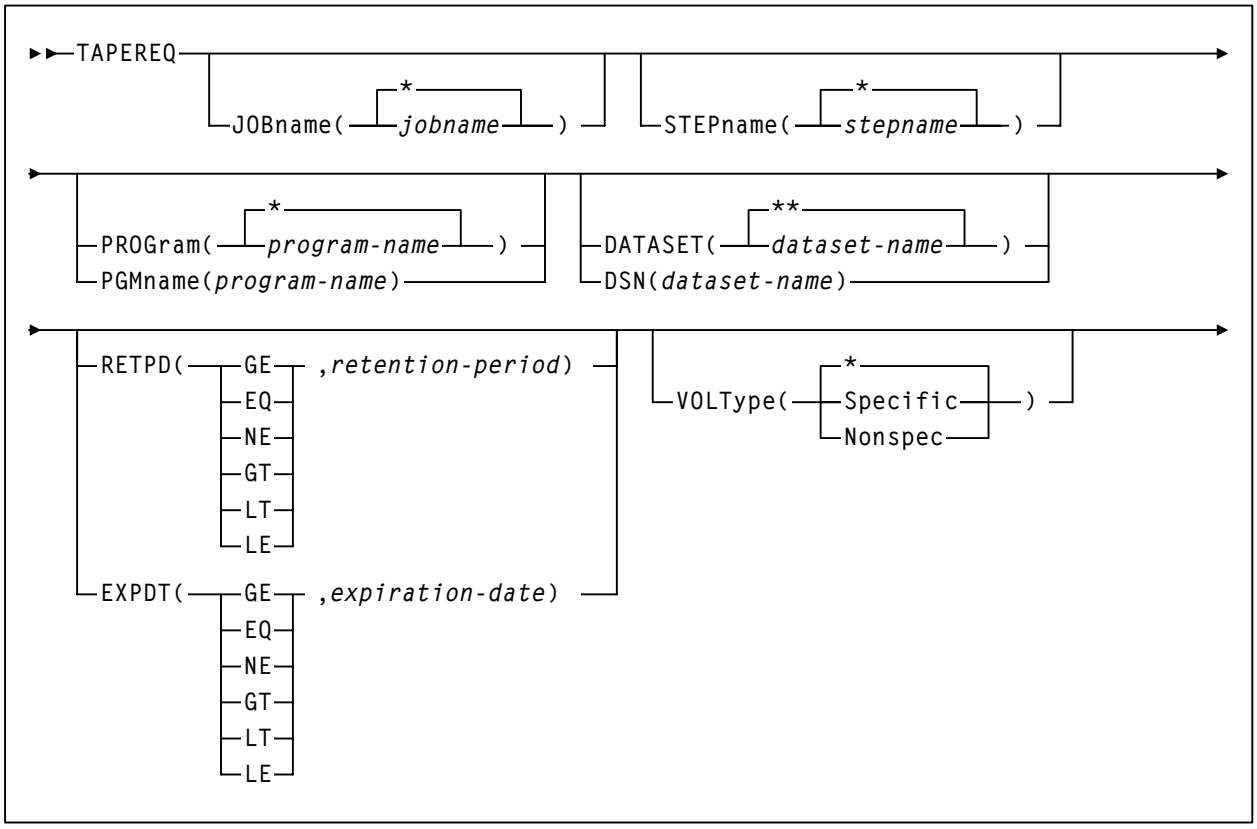


Figure 8. TAPEREQ Control Statement (Part 1 of 2)

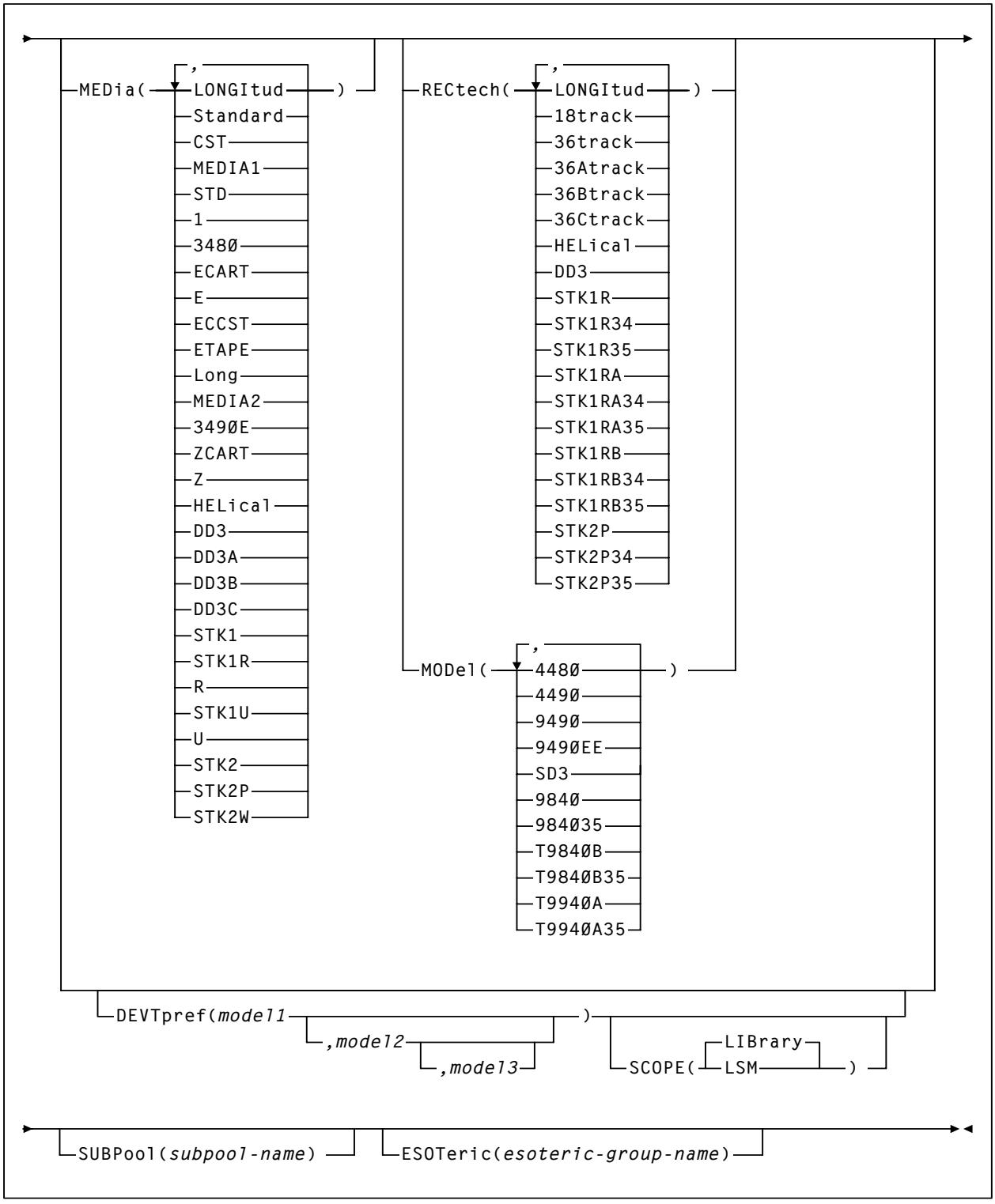


Figure 8. TAPEREQ Control Statement (Part 2 of 2)

Note: See "Syntax Flow Diagrams" on page xx for syntax flow diagramming conventions.

Control Statement Name

TAPEREQ
Initiates the TAPEREQ control statement.

If any of the following conditions exist:

- there is no TAPEREQ control statement specified in the definition data set to match a request
- no matching TAPEREQ statement specifies the MEDia, RECtech, or MODel parameters
- no definition data is loaded

then the MEDia and RECtech or MODel parameters are set to undefined, which matches all requests. Any available device is eligible for allocation.

See Table 1 on page 101 for default values used if the MEDia parameter is omitted.

Parameters

The following sections describe the input and output parameters for the TAPEREQ control statement.

Input Parameters

Note: The JOBname, STEPname, PROGram, and DATASET parameter values can include the following wild card characters:

% or ?
any single non-blank character.

*****
any character string (length zero to 8) not to exceed one qualification level. For example, A.B.* matches A.B and A.B.C, but does not match A.B.C.D.

******
used only in DATASET. Indicates any number of qualifiers (zero or more). Cannot be used with any other characters within a qualifier. For example, A.B.** matches A.B, A.B.C, A.B.C.D, A.B.C.D.E, and so forth.

The TAPEREQ selection criteria (input) parameters include:

JOBname
Optionally, identifies the job name. If JOBname is not specified, the default value is *.

Note: If you are coding a TAPEREQ statement for DFHSM, you must specify JOBname as a selection criteria. Using DATASET (DSN) can cause unpredictable results.

jobname

Specifies the name of the job for which the TAPEREQ control statement is processed.

The TAPEREQ control statement is used only when the specified *jobname* matches the job name in the request.

STEPname

Optionally, specifies the step name. If the STEPname parameter is not specified, the default value is *.

stepname

Specifies the name of the step for which the TAPEREQ control statement is processed. The TAPEREQ control statement is used only when the specified *stepname* matches the step name in the request.

PROGram or PGMname

Specifies the program name.

The program name used is the job step program name. The default value is *.

program-name

Specifies the program name for which the TAPEREQ control statement is processed.

The TAPEREQ control statement is used only when the specified *program-name* matches the program name in the request.

Note: This parameter is not supported in JES3 environments.

DATASET or DSN

Specifies the data set name.

The default value is **.

dataset.name

Specifies the data set name for which the TAPEREQ control statement is processed.

The TAPEREQ control statement is used only when the specified *dataset.name* matches the data set name in the request.

Notes:

- If you are coding a TAPEREQ statement for DFHSM, you must specify JOBname as a selection criterion. Using DATASET (DSN) can cause unpredictable results.
- The name you specify on this parameter might be different from the value coded in the DSN parameter on a DD statement. For example,

  ```
  DSN=&&ABC
  ```

  may be coded on a DD statement if a data set is temporary. However, the actual data set name is not &&ABC. Similarly, if the DSN parameter refers back to a previous DD statement, the data set name is resolved to the referred-to data set. Thus, the data set name referred to must be coded on the DATASET parameter in order for the TAPEREQ control statement to be processed.

RETPD

Specifies the retention period and the relationship that must exist. If you do not specify either RETPD or EXPDT, the default value is RETPD(GE,0) (any expiration date will match this criterion).

EQ equal to.

NE not equal to.

GT greater than.

GE greater than or equal to.

LT less than.

LE less than or equal to.

retention-period

Specifies the retention period in days for the data set. Specify the number of days as a 1- to 4-digit decimal number.

Note: This parameter is not supported in JES3 environments.

EXPDT

Specifies the expiration date and the relationship that must exist.

EQ equal to.

NE not equal to.

GT greater than.

GE greater than or equal to.

LT less than.

LE less than or equal to.

expiration-date

Specifies the expiration date of the data set in *YYDDD* or *YYYY/DDD* format.

Note: This parameter is not supported in JES3 environments.

VOLType

Specifies whether or not a non-specific volume is being requested. If VOLType is omitted, the default value is * (either specific or non-specific matches).

Specific

Specifies that the request is for a specific volume.

Note: The TAPEREQ parameter never overrides the actual media type of the specific volume returned by the LCS.

Nonspec

Specifies that the request is for a non-specific (scratch) volume.

Output Parameters

The TAPEREQ media and format requirements parameters include:

MEDia

Specifies the desired type of media for a data set being created. You can enter a list of media types, but they must be separated by commas.

Note: A list specifies a generic pool from which a selection is made. There is no implied priority.

This parameter is ignored for an existing data set if the media characteristics are determined by the volume information contained in the LCS's control data set (CDS).

If this parameter is not specified, a default is chosen based on the value of the RECtech parameter. The following table shows default values used if MEDia is omitted:

Table 1. TAPEREQ MEDia Default Values

RECtech Entered:	MEDia Default
18track	Standard
36track, 36Atrack, 36Btrack	LONGItud
36Ctrack	ZCART
LONGItud	LONGItud
DD3, Helical	DD3
STK1R, STK1R34, STK1R35	STK1R
STK2P, STK2P34, STK2P35	STK2

To avoid problems, StorageTek recommends that all TAPEREQ statements specify MEDia and RECtech consistently. Be sure to include one or both parameters on each statement. If both media type and recording technique are omitted, all available devices are eligible for allocation.

Valid media types are:

LONGItud
Indicates standard, enhanced (ECART), or extended-enhanced (ZCART) capacity cartridges.

Standard
Indicates a standard capacity cartridge.

Synonyms for this type of cartridge include:

- CST
- MEDIA1
- STD
- 1
- 3480

A standard capacity cartridge can be used on any longitudinal transport (i.e. 4480, 4490, 9490, or 9490EE). However, if the data is written on the tape in 36-track, the data cannot be read by an 18-track 4480 transport.

ECART

Indicates an enhanced capacity cartridge.

An ECART cartridge can be used only on 36-track transports (i.e. 4490, 9490, and 9490EE), and is identified by a two tone colored case.

Synonyms for this type of cartridge include:

- E
- ECCST
- ETAPE
- Long
- MEDIA2
- 3490E

ZCART

Indicates an extended-enhanced capacity cartridge.

A ZCART cartridge can be used only on TimberLine 9490EE 36-track transports.

ZCART can be abbreviated as Z.

HELical

Indicates a helical cartridge.

A helical cartridge can be used only on RedWood transports. The following subtypes and abbreviations specify a helical cartridge:

- **DD3** indicates any DD3A, DD3B, or DD3C helical cartridge.
- **DD3A or A** indicates a helical cartridge with a 10GB media capacity.
- **DD3B or B** indicates a helical cartridge with a 25GB media capacity.
- **DD3C or C** indicates a helical cartridge with a 50GB media capacity.

Note: The seventh position in the external label is encoded with the cartridge type (i.e. A, B, or C).

STK1

Indicates any T9840 cartridge.

STK1R

Indicates a 20GB T9840 data cartridge. STK1R can be abbreviated as R.

STK1U

Indicates a T9840 cleaning cartridge. STK1U can be abbreviated as U.

STK2

Indicates any STK2P cartridge. This parameter is the default for all T9940A data cartridge types.

STK2P

Indicates a T9940A data cartridge.

STK2W

Indicates a T9940A cleaning cartridge.

RECtech

Optionally, specifies the method used to record data tracks on the tape surface for the desired data set. You can enter a list of recording techniques, but they must be separated by commas.

Note: A list specifies a generic pool from which a selection is made. There is no implied priority.

RECtech and MODel are mutually exclusive.

If this parameter is not specified on any matching TAPEREQ, a default is chosen based on the value of the MEDia parameter. The following table shows default values used if RECtech is omitted.

Table 2. TAPEREQ RECtech Default Values

MEDia Entered:	RECtech Default
LONGItud	LONGItud
Standard	LONGItud
ECART	36track
ZCART	36Ctrack
DD3, DD3A, DD3B, DD3C	DD3
STK1, STK1R	STK1R
STK2, STK2P	STK2P

To avoid problems, StorageTek recommends that all TAPEREQ statements specify MEDia and RECtech consistently. Be sure to include one or both parameters on each statement. If both media type and recording technique are omitted, all available devices are eligible for allocation.

Valid recording techniques are:

LONGItud
Indicates the request is to be allocated to a device that records data tracks in a linear format along the length of the tape surface. These devices include 4480, 4490, 9490, and 9490EE transports.

18track
Indicates the request is to be allocated to a 4480 transport.

36track
Indicates the request is to be allocated to a 4490, 9490, or 9490EE transport (any device that records in 36-track mode).

36Atrack
Indicates the request is to be allocated to a 4490 transport.

36Btrack
Indicates the request is to be allocated to a 9490 transport.

The 9490 transport is similar to the 4490 transport, except it supports a higher data transfer rate and ESCON attachment. The 9490 has the same media requirements and restrictions as a 4490 transport.

36Ctrack
Indicates the request is to be allocated to a 9490EE transport.

The 9490EE transport supports an extended-enhanced capacity 36-track recording technique cartridge (ZCART). The ZCART cartridges uses a thinner media to hold twice the capacity of the ECART cartridge.

HELical
Indicates the request is to be allocated to a device that records data tracks as diagonal stripes across the tape surface.

DD3
Indicates the request is to be allocated to a SD-3 transport.

STK1R
Indicates the request is to be allocated to any 9840 or T9840B transport.

STK1R34
Indicates the request is to be allocated to any 3490E-image 9840 or T9840B transport.

STK1R35
Indicates the request is to be allocated to any 3590-image 9840 or T9840B transport.

STK1RA
Indicates the request is to be allocated to a 3490E or 3590-image 9840 transport.

STK1RA34
Indicates the request is to be allocated to a 3490E-image 9840 transport.

STK1RA35
Indicates the request is to be allocated to a 3590-image 9840 transport.

STK1RB
Indicates the request is to be allocated to a 3490E or 3590-image T9840B transport.

STK1RB34
Indicates the request is to be allocated to a 3490E-image T9840B transport.

STK1RB35
Indicates the request is to be allocated to a 3590-image T9840B transport.

STK2P
Indicates the request is to be allocated to a T9940A transport.

STK2P34
Indicates the request is to be allocated to a 3490E-image T9940A transport.

STK2P35
Indicates the request is to be allocated to a 3590-image T9940A transport.

MODel
Optionally, specifies the model number of a transport. You can enter a list of models, but they must be separated by commas.

Note: A list specifies a generic pool from which a selection is made. There is no implied priority.

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

MODel and RECtech are mutually exclusive.

4480
Indicates a 4480 (18-track) transport.

4490
Indicates a 4490 (36-track Silverton) transport.

9490
Indicates a 9490 (36-track Timberline) transport.

9490EE
Indicates a 9490EE (36-track Timberline EE) transport.

SD3
Indicates an SD-3 (RedWood) transport.

9840
Indicates a 3490E-image 9840 transport.

984035
Indicates a 3590-image 9840 transport.

T9940A
Indicates a 3490E-image T9940A transport.

T9940A35
Indicates a 3590-image T9940A transport.

T9840B
Indicates a 3490E-image T9840B transport.

T9840B35
Indicates a 3590-image T9840B transport.

Note: You can specify multiple values for this parameter; separate each value with a comma.

DEVTpref
Requests device preferencing for a particular allocation request. The use of device preferencing is restricted to StorageTek's 36-track class transport models: 4490, 9490, and 9490EE.

Note: The following rules apply when using the MEDia, MODel, and RECtech parameters in conjunction with DEVTpref:

- The MODel parameter must include the models specified in the DEVTpref parameter.
- The RECtech parameter must include valid values for each model specified in the DEVTpref parameter.
- The MEDia parameter must include valid values for each model specified in the DEVTpref parameter.

Note: Device preferencing is managed by the SMC. If DEVTpref is specified and the list of devices includes 36-track and other devices (e.g., 18-track) the SMC preferences 36-track drives first, followed by other types in the list. Refer to the *SMC Configuration and Administration Guide* for more information.

model1

Specifies the transport model number of the preferred device. These transports are given a higher selection probability than the alternate transport model.

model2

Optionally, specifies the transport model number of the first alternate device. It must be separated from the *model1* by either a comma or a blank.

model3

Optionally, specifies the transport model number of the second alternate device. It must be separated from *model2* by either a comma or a blank.

If *model3* is omitted, the transport model number not specified by *model1* or *model2* is assumed to be the second alternate device.

The following table shows the alternate device order if only *model1* is entered:

Table 3. Alternate Device Default Value

Preferred Device **(***model1***):**	***First Alternate Device*** **(***model2***):**	***Second Alternate Device*** **(***model3***):**
4490	9490	9490EE
9490	4490	9490EE
9490EE	4490	9490

Entering DEVTpref(9490,4490) is functionally equivalent to DEVTpref(9490,4490,9490EE).

SCOPE

Specifies whether device preferencing should be performed across the entire library complex (single or multiple-ACS), or at the individual LSM level. The SCOPE parameter may be used in an attempt to reduce pass-thru events. SCOPE is valid only when the DEVTpref parameter is specified on the same TAPEREQ statement.

LIBrary

Specifies that all of the preferred models in the ACS are to be given a higher selection probability than all of the alternate models, while maintaining the ability to automate tape mounts. LIBrary is the default value.

In a multiple-ACS configuration, the way that the MVS/CSC processes SCOPE(LIBrary) depends on the following:

- The ability to automate the mount
- The type of request; specific versus scratch

A specific volume request is confined to the ACS where the volume resides so that the mount can be automated. The transports in all other ACSs, and all non-library transports are marked ineligible for selection. The MVS/CSC then

applies SCOPE(LIBrary) to one ACS, giving all of the preferred models in that ACS a higher selection probability.

A scratch request can be directed to any ACS, whether or not it contains scratch volumes. In this case, the MVS/CSC applies SCOPE(LIBrary) to all ACSs, giving all of the preferred models in all ACSs a higher selection probability.

Note: To prevent a scratch request from being directed to an ACS that does not contain scratch volumes, use the ALTer Zeroscr(ON) command. Refer to the *MVS/CSC Operator's Guide* for a complete description of the ALTer command.

LSM

Specifies that all of the preferred models within an LSM are to be given a higher selection probability than all of the alternate models within that same LSM, while maintaining the ability to automate tape mounts.

Each LSM is processed individually without regard to the models attached to other LSMs in the ACS. This means that if the preferred model is not available in the LSM where the volume resides, an alternate transport in the same LSM is allocated preventing a pass-thru of the volume to a different LSM.

SCOPE(LSM) should be requested if reducing pass-thru events is more important than allocating a preferred transport in a different LSM.

SCOPE applies only when DEVTpref is specified. LIBrary is the default value for the SCOPE parameter.

SUBPool

optionally, specifies the subpool used to satisfy a scratch request. If specified, it takes precedence over a SUBPool specification in user exits 01, 02, or 04.

subpool-name

Specifies the subpool name.

Notes:

- The SUBPOOL parameter cannot be used to specify a subpool name for ACSLS. User Exit SCSUX01 or SCSUX02 MUST be used to specify subpools for ACSLS.
- In order to apply subpool definition changes, all active MVS/CSCs attached to the server must be stopped and restarted.

ESOTeric

optionally, specifies the esoteric defining the list of eligible transports to be used to satisfy a tape request. If specified, it generally takes precedence over an esoteric specification in user exits 02, 04, 08, 10, 12, or 13. See the *MVS/CSC System Programmer's Guide* for affinity separation and location policy precedence guidelines.

esoteric-group-name

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

Defining TAPEREQ Control Statements

The following entries show how to define TAPEREQ control statements to specify tape request attributes for particular data sets.

Note: The TAPEREQ control statements are numbered for purposes of this example.

```
(1)  TAPEREQ DSN(BILLING.**) REC(36C) MED(Z)

(2)  TAPEREQ DSN(PAYROLL.**) RECtech(HEL) MEDIA(DD3A)

(3)  TAPEREQ DSN(SERVICE.**) MODEL(9840) MEDIA(STK1R)

(4)  TAPEREQ JOB(YOURJOB1.**) DSN(PREF.**) REC(36) DEVTP(9490,9490EE)
     SCOPE(LIB)

(5)  TAPEREQ MEDIA(S) RECTECH(36)
```

The following list describes each statement:

- **Statement 1** specifies that if a request is made for a data set with BILLING as the high-level qualifier, then mount an extended-enhanced capacity cartridge on a TimberLine 9490EE transport.
- **Statement 2** specifies that if a request is made for a data set with PAYROLL as the high-level qualifier, then mount a helical cartridge with a capacity of 10GB on a RedWood transport.
- **Statement 3** specifies that if a request is made for a data set with SERVICE as the high-level qualifier, then mount a 9840 cartridge on a 9840 transport.
- **Statement 4** specifies device preferencing at the library level for all data sets with PREF as the high-level qualifier in the job named YOURJOB1. If a request is made for a data set with PREF as the high-level qualifier, allocate a 9490 transport. If a 9490 transport is not available, then mount the volume on a 9490EE or 4490 transport.
- **Statement 5** specifies a user-defined default; for all data sets that do not match one of the first three statements, mount a standard capacity tape on a 4490, 9490, or 9490EE transport.

Processing of TAPEREQ Control Statements

The examples in this section show how the MVS/CSC processes TAPEREQ parameters.

TAPEREQ Processing - Example 1

A user submits the following job:

```
//MYJOB  JOB (1234),MYNAME,CLASS=A
//MYSTEP EXEC PGM=MYPGM
//MYDD   DD DSN=THIS.DATA.SET,RETPD=90,UNIT=CART,DISP=(,CATLG)
```

The definition data set specified by the TREQDEF command contains the following TAPEREQ control statements. The TAPEREQ control statements are numbered for purposes of this example.

(1) `TAPEREQ JOB(MY**) STEP(MY*) DSN(MY*.**) EXPDT(LT,99365) MED(S)`

(2) `TAPEREQ JOB(*JOB) STEP(*STEP) PGM(MY*) REC(36) DEVTP(4490) SCOPE(LSM)`

(3) `TAPEREQ DSN(**.DATA.**) REC(36)`

The MVS/CSC searches the TAPEREQ control statements in the order specified.

Statement 1
JOB, STEP, and EXPDT match but DSN does not, thus the statement does not apply.

Statement 2
JOB, STEP, and PGM match, thus the statement applies. The recording technique is set to 36track with 4490, 9490, and 9490EE transports included in the MVS EDL or JES3 IJS, however, 4490 transports are preferred over 9490 and 9490EE transports in the LSM where the volume resides.

Statement 3
This statement also matches, however, it does not apply since the search ends when a match is found.

Note: In this example media is set to Standard which is the default when the MEDia parameter is not specified.

TAPEREQ Processing - Example 2

The definition data set specified by the TREQDEF command contains the following TAPEREQ control statements. The TAPEREQ control statements are numbered for purposes of this example.

```
(1)  TAPEREQ JOB(TSTHEL*) MED(DD3A)
(2)  TAPEREQ JOB(*HEL*)   MOD(SD3)
(3)  TAPEREQ JOB(TESTR*)  MED(STK1R)
(4)  TAPEREQ JOB(TEST36*) MED(STANDARD) REC(36TRACK) DEVTP(9490)
(5)  TAPEREQ JOB(TESTZ*)  MED(ZCART)    REC(36C)
(6)  TAPEREQ JOB(PROD*)   MED(ECART)
(7)  TAPEREQ JOB(*)       MED(STANDARD) REC(18TRACK)
```

A user submits six jobs: TSTHELIC, TESTRMED, TESTZMED, TEST36T, PRODJOB, and ANYJOB. The matching of the TAPEREQ parameters occurs as follows:

- The first and second TAPEREQ control statement parameters are used for job TSTHELIC. A 10 GB capacity helical cartridge is mounted on a RedWood transport; media type is taken from statement 1 and model number is taken from the first statement that matches the input criteria.
- The third TAPEREQ control statement parameters are used for job TESTRMED; a 9840 cartridge is mounted on a 9840 transport.
- The fourth TAPEREQ control statement parameters are used for job TEST36T; a standard capacity cartridge is mounted on a 9490 transport (if one is available at the library level), or on a 4490 or 9490EE transport (if a 9490 is not available).
- The fifth TAPEREQ control statement parameters are used for job TESTZMED; an ZCART cartridge is mounted on a TimberLine 9490EE transport.
- The sixth TAPEREQ control statement parameters are used for job PRODJOB. An enhanced capacity cartridge is mounted on a 4490, 9490, or 9490EE transport; MEDIA(LONG) implies a 36-track recording technique.
- The last TAPEREQ control statement parameters are used for job ANYJOB. A standard capacity cartridge is mounted on a 4480 transport.

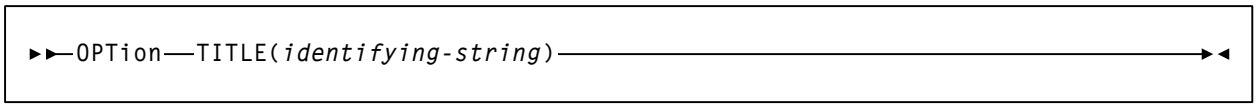
OPTION TITLE Control Statement

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.

You can display the identifying string with the MVS/CSC Display TREQDEF operator command. See the *MVS/CSC Operator's Guide* for descriptions of command syntax and parameters for the Display operator command.

Syntax



Note: See "Syntax Flow Diagrams" on page xx 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')
```

Part 3. MVS/CSC Communications in a Non-Sysplex Environment

Chapter 6. Configuring Communications with a Unix-Based LCS

Overview

This chapter 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. See ACSLS documentation 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 in "Specifying the INTERNET Startup Parameter". 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:

- IBM TCP/IP
- Interlink TCPaccess
- Interlink CISCO IOS for S/390

See the *NCS Installation Guide* 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 69 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 79 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. See 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.

MVS/CSC and Interlink TCPaccess

If you are using Interlink TCPaccess 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 117 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 120 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 120). 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".

The following sections describe the procedures for setting up VTAM and APPC/MVS on the MVS client system for SNA LU 6.2 communications. (See 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.

See 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" 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. See *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

This section describes the steps for setting up APPC/MVS, including:

- Defining a system base LU to APPC/MVS
- Creating a side information file and adding an entry that identifies the LCS

Defining a System Base LU to APPC/MVS

To define a system base LU that APPC/MVS uses to establish sessions between the MVS/CSC and the LCS:

- Create an APPCPM*xx* 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 120 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 APPCPMYY of the SAMPLIB library.

For more information about the APPCPM*yy* member, see *IBM MVS/ESA Planning: APPC Management*.

Creating the side information file

You must 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 Ø)                        -
      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 APPCPM*xx* 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).

See *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 APPCPM*xx* member name on the START command. For example:

```
START APPC,SUB=MSTR,APPC=xx
```

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

Chapter 7. Setting Up Communications With a VM-Based LCS

Overview

This chapter 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. See CLS documentation 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 in "Specifying the INTERNET Startup Parameter". 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:

- Interlink TCPaccess
- Interlink CISCO IOS for S/390

See the *NCS Installation Guide* 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 69 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 79 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 70 for more information about specifying the PORT startup parameter.

MVS/CSC and Interlink TCPaccess

If you are using Interlink TCPaccess 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 81 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.

Notes:

- 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.

Part 4. MVS/CSC Communications in a Sysplex Environment

Chapter 8. Configuring Communications In a Base or Parallel Sysplex Environment

Overview

This chapter describes the communications setup procedures required for the MVS/CSC to communicate with an MVS-based LCS in a base or parallel sysplex environment. TCP/IP, XCF, and SNA LU 6.2 are the communications methods supported for the MVS-based LCS in a base or parallel sysplex environment.

This chapter also describes guidelines you should follow when setting up communications in a parallel sysplex to receive optimum performance during tape automations.

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 in "Specifying the INTERNET Startup Parameter". 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:

- IBM TCP/IP
- Interlink TCPaccess
- Interlink CISCO IOS for S/390

See the *NCS Installation Guide* 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 69 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 79 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. See 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.

MVS/CSC and Interlink TCPaccess

If you are using Interlink TCPaccess 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 XCF as the Communications Method

Cross-system coupling facility (XCF) is supported as a communications method for MVS sysplex environments. XCF services must be available on all MVS systems that are part of the sysplex.

XCF is selected as the communications method using the SRVRList startup parameter. The XCFGROUP startup parameter must also be specified for XCF communications. The specification of the XCF member name and XCF group name establishes communications between the MVS/CSC and MVS-based LCS.

Specifying the SRVRLIST Startup Parameter

Use the SRVRList startup parameter to specify XCF as the communications method (SRVRLIST (XCF, *xcf_member_name*)), where XCF identifies XCF as the communications method, and *xcf_member_name* is the XCF member name defined for the MVS-based LCS. You can also use the SRVRList startup parameter to specify up to three MVS-based LCSs to eliminate the LCS as a single point of failure, or multiple communications methods (XCF and SNA LU 6.2 only) to eliminate the communications link as a single point of failure.

The following example shows how to specify XCF as the communications method and dual MVS-based LCSs to eliminate the LCS as a single point of failure.

```
SRVRL(XCF,xcf_member_name1,XCF,xcf_member_name2)
```

The following example shows how to specify both XCF and SNA LU 6.2 as the communications methods to eliminate the communications link as a single point of failure.

```
SRVRL(XCF,xcf_member_name,LU6,symdestname)
```

The XCF member name you specify must match the XCF member name defined for the MVS-based LCS. See "SRVRLIST Startup Parameter" on page 75 for more information. See the *LibraryStation Configuration Guide* for the default MVS-based LCS member name.

Specifying the XCFGROUP Startup Parameter

Use the XCFGROUP startup parameter to specify the XCF group name. The XCF group name you specify must match the XCF group name defined for the MVS-based LCS. The following example shows how to specify the XCFGROUP startup parameter, where *xcf_group_name* is the XCF group name.

```
XCFGROUP(xcf_group_name)
```

The XCF group name you specify can consist of one to eight characters, including national characters: $, #, and @. Valid alphanumeric characters are 0-9 and upper case A-Z.

To avoid duplicating IBM XCF group names, do not start group names with A-I or with the character string "SYS". In addition, the group name UNDESIG is reserved.

- See "XCFGROUP Startup Parameter" on page 82 for more information about specifying the XCFGROUP startup parameter.
- See *IBM MVS/ESA Planning: Sysplex Management* for information about defining XCF group names and member names.
- See the *LibraryStation Configuration Guide* for the default MVS-based LCS group name.

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 137 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 140 for information about the APPC/MVS side information file).

Note: Use the SRVRList startup parameter instead of the SYMDESTN startup parameter to specify the symbolic destination name for multiple MVS-based LCSs. See "Specifying the SRVRLIST Startup Parameter" on page 135 for more information.

Specifying the SRVRLIST Startup Parameter

Use the SRVRList startup parameter to specify SNA LU 6.2 as the communications method, along with dynamic server switching (SRVRLIST (LU6, *symdestname*)), where LU6 identifies SNA LU 6.2 as the communications method, and *symdestname* is the symbolic destination name. You can also use the SRVRList startup parameter to specify up to three MVS-based LCSs to eliminate the LCS as a single point of failure, or multiple communications methods (SNA LU 6.2 and XCF only) to eliminate the communications link as a single point of failure.

The following example shows how to specify SNA LU 6.2 as the communications method and dual MVS-based LCSs to eliminate the LCS as a single point of failure.

```
SRVRL(LU6,symdestname,LU6,symdestname)
```

The following example shows how to specify both SNA LU 6.2 and XCF as the communications methods to eliminate the communications link as a single point of failure.

```
SRVRL(LU6,symdestname,XCF,xcf_member_name))
```

The symbolic destination name you specify must match the name specified on the DESTNAME keyword in the active APPC/MVS side information file (see "Creating the side information file" on page 140.

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 140). 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".

The following sections describe the procedures for setting up VTAM and APPC/MVS on the MVS client system for SNA LU 6.2 communications. (See 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.

See 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” 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. See *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

This section describes the steps for setting up APPC/MVS, including:

- Defining a system base LU to APPC/MVS
- Creating a side information file and adding an entry that identifies the LCS

Defining a System Base LU to APPC/MVS

To define a system base LU that APPC/MVS uses to establish sessions between the MVS/CSC and the LCS:

- Create an APPCPM*xx* member (*xx* is the two-character suffix used on the MVS START APPC command) in SYS1.PARMLIB, and add the following statements:

  ```
  LUADD ABCNAME(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" 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 APPCPMYY of the SAMPLIB library.

For more information about the APPCPM*yy* member, see *IBM MVS/ESA Planning: APPC Management*.

Creating the side information file

You must 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 Ø)                        -
        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 APPCPM*xx* 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).

See *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 APPCPM*xx* member name on the START command. For example:

```
START APPC,SUB=MSTR,APPC=xx
```

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

Sample MVS/CSC and MVS-based LCS SNA LU 6.2 Parameter Mappings

MVS/CSC startup parameters	LibraryStation startup parameters
• COMM(LU6) • SYMDESTN(LIBSTAT)	• COMMTYPE(LU6) • SYMDESTN(LIBSTAT)
VTAM environment specifications ***Local/remote LU names*** • Local LU = CSCLU1 • Remote LU = LSLU ***Local LU VTAM application definition*** • CSCLU1 APPL ACBNAME = CSCLU1 . . .	**VTAM environment specifications** ***Local/remote LU names*** • Local LU = LSLU • Remote LU = CSCLU1 ***Local LU VTAM application definition*** • LSLU APPL ACBNAME = LSLU . . .

APPC/MVS environment specifications
System base LU definition
`LUADD ACBNAME(CSCLU1) BASE NOSCHED`
`TPDATA(side_info_file) SIDEINFO`
`DATASET(side_info_file)`
Side information file entries
• DESTNAME(LIBSTAT)
• TPNAME = CSCI
• PARTNER_LU = LSLU
• MODENAME = APPCHOST
	APPC/MVS environment specifications
	System base LU definition
	`LUADD ACBNAME(LSLU) BASE NOSCHED`
	`TPDATA(side_info_file) SIDEINFO`
	`DATASET(side_info_file)`
	Side information file entries
	• DESTNAME(LIBSTAT)
	• TPNAME = CSCI
	• PARTNER_LU = LSLU
	• MODENAME = APPCHOST

Parallel Sysplex Performance Tuning

Use the following guidelines to receive optimum performance during tape automations in a parallel sysplex.

- Define either a combination of channel-to-channel (CTC) communication connections and coupling facility list structures, or define CTC communication connections exclusively to establish the signalling paths required for XCF group members to communicate.
- Define transport classes to segregate message traffic.
- If you are using a product such as MIM to serialize the use of shared resources, specify GRSRNL=EXCLUDE in the IEASYSxx parmlib member.
- If you are not using another product to serialize the use of shared resources:
  - Verify that the ACCELSYS option in the GRSCNFxx parmlib member specifies ring acceleration and 2 as the number of systems that must see a resource request before the request is granted.

    Specifying a low ACCELSYS value improves performance by reducing the amount of time that tasks must wait for access to global resources.

- Verify that the RESMIL option in the GRSCNFxx parmlib member specifies 2 milliseconds as the residency time value.

  Specifying a low RESMIL value improves ring performance by increasing ring capacity and decreasing response time for processing requests for global resources.

- Eliminate any unnecessary serialization defined in the GRSRNLxx parmlib member.

- If you are not running the coupling facility on a stand-alone processor (i.e. 9674 Coupling Facility), verify that sufficient CPU resources are available to the coupling facility LPAR.

  If possible, dedicate one or more CPUs to the coupling facility LPAR. If this is not possible, verify that there are enough shared CPUs available with adequate weighting factors. These weighting factors should allow at least a 50% chance for the coupling facility LPAR to be dispatched at any given time.

- Allocate XCF couple data sets on volumes that are free from any non-sysplex ENQUEUE or RESERVE activity.

- If your XCF couple data sets are allocated on a DASD device, review the sysplex failure management (SFM) policies; verify that there is an adequate amount of time specified on the SFM DEACTTIME, RESETTIME, and ISOLATETIME parameters to allow a warm boot to complete before the coupling facility starts recovery processing.

See the following IBM manuals for additional information.

- *MVS/ESA Planning: Global Resource Serialization*
- *MVS/ESA Planning: Sysplex Management*
- *MVS/ESA Initialization and Tuning Reference*
- *MVS/ESA Setting Up a Sysplex*
- *System/390 MVS Parallel Sysplex Test Report*

Part 5. MVS/CSC Configuration Verification and Startup Procedures

Chapter 9. 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:

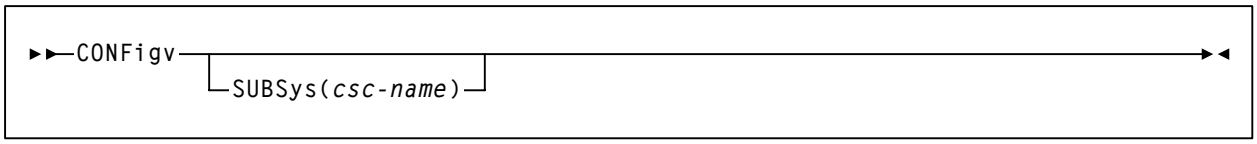
- User-selected startup parameters
- Device esoteric definitions
- LCS configuration compatibility

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

Figure 9 shows sample JCL used to invoke the Configuration Verification utility.

```
//CSCØCFGV JOB ACCT,PGMR
//STEPØ    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-parm-member)
//SCSPRINT DD SYSOUT=*
//SCSIN    DD *
  CONFIGV  [SUBS(CSC1)]
/*
//
```

Figure 9. 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.

Notes:

- If you are running MVS/CSC and MVS/HSC on the same MVS system, the STEPLIB DD statement that identifies the MVS/CSC load libraries is required (see Figure 9). If you omit the STEPLIB DD statement, unpredictable MVS/CSC operations may occur.
- If you installed the MVS/CSC into the HSC and LibraryStation CSI, all references to SACLINK must point to the LibraryStation's SACLINK data set.

Sample Output

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

- A listing of input control statements with appropriate messages when syntax errors occur. Sample output is shown in Figure 10.
- A formatted printout of MVS/CSC startup parameters verified (when all parameters are valid). Sample output is shown in Figure 11 on page 150.
- A formatted printout of MVS/CSC startup parameters verified including error conditions for invalid parameters. Sample output is shown in Figure 12 on page 151.

```
SCUADMIN (5.1.0)    StorageTek MVS Client/Server System Utility     PAGE 001
TIME 10:51:02           Control Card Image Listing        DATE 2002-06-07


 CONF SUBS(CSC1)
```

Figure 10. Configuration Verification Utility Sample Output-Input Control Statement

```
SCUADMIN (5.1.0)  StorageTek MVS Client/Server System Utility          PAGE 002
TIME  10:51:02     Configuration Verify                          DATE  2002-06-07

 ****************************************************************
 ****************************************************************
  SERVER(LS)                                  * SERVER TYPE
  LIBDEV(LIB0,LIB1)                           * LIBRARY ESOTERICS
  NONLIB(NLIB)                                * NON-LIB ESOTERIC
  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
  DELDISP(NOSCRTCH)                           * DELETE DISPOSITION
  SCRLABL(SL)                                 * SCRATCH LABEL TYPE
  DEFER(NO)                                   * DEFERRED MOUNTS
  TCPNAME(TCPIP)                              * IBM's TCP/IP ADDRESS SPACE NAME
  INTERNET(129.80.41.126)                     * INTERNET ADDRESS
  TRACDEST(LOG)                               * TRACE DESTINATION
  Trace(YES)                                  * TRACE ACTIVITY
  TREQDEF(MY.TAPEREQ.FILE)                    * TAPEREQ DEFINITIONS
  LOG(RESET)                                  * LOGGING OPTION

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

Figure 11. CONFIGV Sample Output - All Parameters Valid (MVS-Based LCS with TCP/IP)

```
SCUADMIN (5.1.Ø)  StorageTek MVS Client/Server System Utility          PAGE Ø02
TIME  1Ø:51:Ø2     Configuration Verify                          DATE  2ØØ2-Ø6-Ø7

 ****************************************************************
 ****************************************************************
  SERVER(LS)                                    * SERVER TYPE
  LIBDEV(LIBØ,LIB1)                             * LIBRARY ESOTERICS
  NONLIB(NLIB)                                  * NON-LIB ESOTERIC
  LIBUNIT(1AØØ,1Ø1,1Ø2,1Ø3, -                   * LIBRARY DEVICES
          1Ø4,1Ø5,1Ø6,1Ø7, -                    * LIBRARY DEVICES
          1ØB,1Ø1,1Ø2,1Ø3, -                    * LIBRARY DEVICES
          1Ø4,1Ø5,1Ø6,1Ø7)                      * LIBRARY DEVICES
  UNITMAP(1ØA,ØØ:ØØ:8:6,1Ø1,ØØ:ØØ:8:5, -       * UNIT MAPPINGS
          1Ø2,ØØ:ØØ:8:8,1Ø3,ØØ:ØØ:8:7, -       * UNIT MAPPINGS
          1Ø4,ØØ:ØØ:9:6,1Ø5,ØØ:ØØ:9:5, -       * UNIT MAPPINGS
          1Ø6,ØØ:ØØ:9:8,1Ø7,ØØ:ØØ:9:7, -       * UNIT MAPPINGS
          1ØB,Ø1:ØØ:8:6,1Ø1,Ø1:ØØ:8:5, -       * UNIT MAPPINGS
          1Ø2,Ø1:ØØ:8:8,1Ø3,Ø1:ØØ:8:7, -       * UNIT MAPPINGS
          1Ø4,Ø1:ØØ:9:6,1Ø5,Ø1:ØØ:9:5, -       * UNIT MAPPINGS
          1Ø6,Ø1:ØØ:9:8,1Ø7,Ø1:ØØ:9:7)         * UNIT MAPPINGS
*
  COMM(TCPIP)                                   * TCPIP COMMUNICATIONS
  COMPRFX('/')                                  * MVS/CSC COMMAND PREFIX
  MSGCASE(MIXED)                                * OUTPUT CASE
  DELDISP(NOSCRTCH)                             * DELETE DISPOSITION
  SCRLABL(SL)                                   * SCRATCH LABEL TYPE
  DEFER(YES)                                    * DEFERRED MOUNTS
  TCPNAME(TCPIP)                                * IBM's TCP/IP ADDRESS SPACE NAME
  INTERNET(129.8Ø.41.126)                       * INTERNET ADDRESS
  TRACDEST(LOG)                                 * TRACE DESTINATION
  Trace(YES)                                    * TRACE ACTIVITY
  TREQDEF(MY.TAPEREQ.FILE)                      * TAPEREQ DEFINITIONS
  LOG(RESET)                                    * LOGGING OPTION

SCSØ836E SCSPARM parameter DEFER value invalid; must be YES, NO, or JES3
SCSØ723E Library device 1AØØ supplied in the SCSPARM LIBUNIT parameter not
 contained in any ACS esoteric
SCSØ155I Condition code for utility function is 8
```

Figure 12. CONFIGV Sample Output - Some Invalid Parameters (MVS-Based LCS with TCPIP)

Chapter 10. 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 operating the MVS/CSC.

Notes:

- An SMC subsystem must be active for the MVS/CSC to initialize. Refer to the *SMC Configuration and Administration Guide* for more information.
- MVS/CSC 5.1 requires a valid license key for initialization. See Chapter 4, "Configuring the MVS/CSC License Key" on page 87 for more information.

MVS/CSC Operations

The following sections describe normal MVS/CSC operations, including:

- Pre-initializing the MVS/CSC
- Starting the MVS/CSC
- Stopping the MVS/CSC

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|[AMPND|NOAMPND]]
```

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.

AMPND
Instructs the MVS/CSC to automate pending mounts for configured drives during initialization. This setting is the default and need not be entered.

Notes:

- Configured drives are defined as those listed in the MVS/CSC LIBUNIT startup parameter.
- SCRPOOL and TAPEREQ requests are not honored for pending mounts processed during initialization.
- Automation of pending mounts for Virtual Tape Drives (VTDs) is not supported.

NOAMPND

Instructs the MVS/CSC to bypass automation of pending mounts during initialization.

Note: AMPND and NOAMPND are mutually exclusive. Specifying both results in messages SCS0004I and SCS0511I.

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 MVS/CSC 5.1 LICENSED/SECRET/UNPUBLISHED
          WORK/COPYRIGHT (1992 - 2002) STORAGETEK
...
...
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.

Orderly Shutdown

During an orderly shutdown, the MVS/CSC waits for processing of all activities in progress to be completed before completing shutdown.

Immediate 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 communications access method software:

- IBM TCP/IP, Interlink TCPaccess, or Interlink CISCO IOS for TCP/IP communications
- VTAM for "3270 BISYNC" communications
- APPC/MVS and VTAM for SNA LU 6.2 communications
- Cross-system coupling facility (XCF) for XCF communications

should be operational before starting the MVS/CSC subsystem. 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.

Part 6. Appendices

Appendix A. MVS/CSC Samples, Source Code Modules, and Macros

The following tables list the MVS/CSC samples, source code modules, and macros contained on the NCS installation tape:

Table 4. MVS/CSC Samples

Member Name	Description
APPCPM*yy*	Sample definition of a system base LU for APPC/MVS
CSCPARM0	This is a sample MVS/CSC startup parameter file that is an example of an attachment to a VM-based (CLS) server using VTAM communications
CSCPARM1	This is a sample MVS/CSC startup parameter file that is an example of an attachment to VM-based (CLS) dual servers using TCP/IP communications
CSCPARM2	This is a sample MVS/CSC startup parameter file that is an example of an attachment to a UNIX-based (ACSLS) server using TCP/IP communications
CSCPARM3	This is a sample MVS/CSC startup parameter file that is an example of an attachment to an MVS-based (LibraryStation) server using TCP/IP communications
CSCPARM4	This is a sample MVS/CSC startup parameter file that is an example of an attachment to an MVS-based (LibraryStation) server using LU 6.2 communications
CSCPARM5	This is a sample MVS/CSC startup parameter file that is an example of an attachment to a UNIX-based (ACSLS) server using LU 6.2 communications
CSCPARM6	This is a sample MVS/CSC startup parameter file that is an example of an attachment to an MVS-based (LibraryStation) server using XCF communications
CSCPROC	Sample startup JCL
DEFAPPC	Sample JCL to create APPC Side Information File
IEAAPF*yy*	Sample MVS/CSC APF list entries
IEFSSN*yy*	Sample MVS/CSC subsystem name table entry
JCLCFGV1	Sample JCL for Configuration Verification Utility to verify startup parameters and MVS system definitions only
JCLCFGV2	Sample JCL for Configuration Verification Utility to verify startup parameters, MVS system definitions, and the server configuration for compatibility
JCLCONDB	Sample JCL for Scratch Conversion Utility

Table 4. MVS/CSC Samples (Continued)

Member Name	Description
JCLLOGR	Sample JCL for Event Log Report Utility
JCLSCRUP	Sample JCL for Scratch Update Utility
LU6APPL	Sample definition of a local LU for VTAM
PGMISAM1	Sample QVOLUME request issued within a single MVS/CSC subsystem environment
PGMISAM2	Sample QCSC and QVOLUME request issued within a multiple MVS/CSC subsystem environment
PROG*yy*	Sample MVS/CSC APF list entries
SCHED*yy*	Sample MVS Program Properties Table (PPT) entry for MVS/CSC
TREQSAM1	Sample TAPEREQ control statements
TREQSAM2	Sample TAPEREQ control statements
UX01CSC1	Sample MVS/CSC User Exit 01 to process mount messages
UX02CSC1	Sample MVS/CSC User Exit 02 to influence scratch allocation (JES2)
UX04CSC1	Sample MVS/CSC User Exit 04 to influence scratch allocation (JES3)
UX05CSC1	Sample MVS/CSC User Exit 05 which returns a non-operational return code
UX08CSC1	Sample MVS/CSC User Exit 08 to influence specific allocation (JES2)
UX09CSC1	Sample MVS/CSC User Exit 09 to influence MVS defer of mounts
UX10CSC1	Sample MVS/CSC User Exit 10 to influence unit affinity separation (JES2)
UX11CSC1	Sample MVS/CSC User Exit 11 to influence JES3 defer of mounts
UX12CSC1	Sample MVS/CSC User Exit 12 to influence unit affinity separation (JES3)
UX13CSC1	Sample MVS/CSC User Exit 13 to influence specific allocation (JES3)

Table 5. MVS/CSC Source Code Modules

Member Name	Description
SCUCONDB	Source code for Scratch Conversion Utility.
SCUDRCA1	Source code for Scratch Conversion Utility CA-1 (TMS) database READ routine.
SCUDRTLM	Source code for Scratch Conversion Utility CA-TLMS (TMLS) database READ routine.
SCUDRRMM	Source code for Scratch Conversion Utility DFSMSrmm database READ routine.
SCUDRZAR	Source code for Scratch Conversion Utility Zara database READ routine.

Table 6. MVS/CSC Macros

Member Name	Description
SCSUX01P	MVS/CSC User Exit 01 parameter list
SCSUX02P	MVS/CSC User Exit 02 parameter list
SCSUX04P	MVS/CSC User Exit 04 parameter list
SCSUX05P	MVS/CSC User Exit 05 parameter list
SCSUX08P	MVS/CSC User Exit 08 parameter list
SCSUX09P	MVS/CSC User Exit 09 parameter list
SCSUX10P	MVS/CSC User Exit 10 parameter list
SCSUX11P	MVS/CSC User Exit 11 parameter list
SCSUX12P	MVS/CSC User Exit 12 parameter list
SCSUX13P	MVS/CSC User Exit 13 parameter list
SCSXREQ	Programmatic Interface request
SCSXREQM	Programmatic Interface mapping macro

Appendix B. Updating the JES3 Initialization Deck

For MVS systems running JES3 with TAPE SETUP processing, you must define the cartridge transports that JES3 uses to satisfy job requests. You must also define the user-assigned esoteric names for each device type in each ACS and for non-library device types. TAPE SETUP processing occurs for cartridge transports identified on SETNAME statements when JOB, high watermark setup (HWS), or tape high watermark setup (THWS) is specified on the SETUP parameter of the JES3 STANDARDS initialization statement.

Refer to the *SMC Configuration and Administration Guide* for information about defining JES3 initialization statements.

Appendix C. Considerations When Setting Network Timeout Parameters

This appendix contains guidelines for setting the network timeout parameters for the MVS/CSC (RETTime, RETCount, and REQTime) and the equivalent parameters on the ACSLS and LibraryStation servers. 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, ACSLS, and LibraryStation. However, the ACSLS timeout parameters are defined with unique names, which do not match those used by MVS/CSC and LibraryStation. These names are summarized in Table 7.

Table 7. Timeout Parameter Names Cross-Reference Table

LibraryStation and MVS/CSC Name	ACSLS Name	Description
RETTime	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 document, MVS/CSC parameter names are used to refer to all network timeout parameters.

In addition to the timeout parameters listed in Table 7, MVS/CSC provides the ALOCTime timeout parameter, which is specific to query requests issued by the MVS/CSC allocation enhancement component during device allocation. See "Considerations When Setting ALOCTIME and REQTIME" on page 169 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 169 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 allocation enhancement component 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 device allocation enhancement component 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 device allocation enhancement component 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 MVS/CSC 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 non-library 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 MVS/CSC 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 and MVS-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. See the *MVS/CSC Messages and Codes Guide* 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:

- Non-library 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 MVS-based and 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 an allocation timeout has occurred. Processing of query requests for volume location and volume attribute information occurs as described in “Considerations When Setting ALOCTIME” on page 167.

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 an allocation timeout has occurred. Processing of query requests for volume location and volume attribute information occurs as described in “Considerations When Setting ALOCTIME” on page 167.

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 167.

Considerations When Setting ALOCTIME and REQTIME in a Multiple-LCS Environment

The following considerations should be taken into account when setting both the ALOCTime and REQTime timeout startup parameters in a multiple-LCS environment where dynamic server switching is specified.

Note: See "SRVRLIST Startup Parameter" on page 75 for information about dynamic server switching.

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 an allocation timeout has occurred. Processing of query requests for volume location and volume attribute information occurs as described in "Considerations When Setting ALOCTIME" on page 167.

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 an allocation timeout has occurred. Processing of query requests for volume location and volume attribute information occurs as described in "Considerations When Setting ALOCTIME" on page 167.

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, MVS/CSC attempts to connect to an alternate LCS. If an alternate LCS is available, query requests for volume location or volume attribute information are redirected to the LCS for processing.

If the time set on ALOCTime has not expired prior to connecting to an alternate server, normal processing occurs for these query requests. If the time set on ALOCTime has expired prior to connecting to an alternate server, processing of query requests for volume location and volume attribute information occurs as defined in "Considerations When Setting ALOCTIME" on page 167.

If an alternate server is not available, a message is issued indicating that the server is unavailable and 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 167.

Note: You can specify a higher timeout value on the ALOCTime timeout startup parameter to allow the MVS/CSC more time to connect to an alternate server for processing of query requests for volume location and volume attribute information. The time that you specify depends on the server and network performance.

Appendix D. Third-Party Software Restrictions

MIM

Multi-image Manager (MIM) 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, the following procedures must be followed when using MIM with MVS/CSC:

- The MVS/CSC can be started *before* or *after* MIM has been initialized.
- Devices within the MVS/CSC-controlled library cannot be in the MIM device-preference list. However, manual transports can remain in the device-preference list.
- MIM should not be used to control the ENQ in the MVS/CSC startup parameters.

Appendix E. 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.).

See the *Requesting Help from Software Support* guide for more information.

Appendix F. Migration and Coexistence

This appendix provides guidelines for migration from previous releases of the MVS/CSC to MVS/CSC Release 5.1. 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 5.1

Perform the following steps:

- Make the MVS/CSC 5.1 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.
- Remove any down-level (non-5.1) LibraryStation servers from the ordered server list used in dynamic server switching.
- Start MVS/CSC 5.1.

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 5.1 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 LibraryStation

MVS/CSC 5.1 is not compatible with LibraryStation releases prior to 5.1.

Appendix G. 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
STK	StorageTek
Sysplex	System complex
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
XCF	Cross-system coupling facility

Glossary

Terms are defined as they are used in the text. If you cannot find a term here, check the index.

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 4480, 4490, 9490, 9490EE, SD3, or 9840 cartridge drives, and cartridges residing in the ACSs.

ACSSA— The ACS System Administrator console provides access to the LCS and the library for the UNIX-based LCS.

ACSLS— *See* Automated Cartridge System Library Software.

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).

authorization— The granting of VM userids access to the CLS system.

Automated Cartridge System (ACS)— The library subsystem consisting of one or two Library Management Units (LMUs) and from one to 16 Library Storage Modules (LSMs) attached to the LMUs.

Automated Cartridge System Library Software (ACSLS)— 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 the StorageTek CARTLIB product.)

cartridge transport— *See* transport.

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

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 computing system (CCS)— A computer and an operating system.

client-initiated utilities (CIU)— VM/HSC utilities that can be executed from a CLS or client operator console.

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).

Client System Interface— Software that provides a transport and translation mechanism between the Library Control System (LCS) and the Client System Component (CSC).

CLS— *See* Common Library Services.

CLSCM— *See* Common Library Services Manager.

CLSCOMM— *See* Common Library Services Communication.

CLSM— *See* Common Library Services Manager.

CLSLP— *See* Common Library Services Logical Port.

CLSOC— *See* Common Library Services Operator Console.

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.

Common Library Services (CLS)— A Storage Technology software system that allows single or multiple non-IBM systems (client systems) to use the ACS.

Common Library Services Communication (CLSCOMM)— 3270 communication interface that connects CLS logical port to the client system.

Common Library Services Configuration Management (CLSCM)— *See* Configuration Management.

Common Library Services Logical Port (CLSLP)— The CLS software that resides on the CLS, and 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.

Common Library Services Manager (CLSM)— The CLS administrator virtual machine from where all CLS functions are controlled. This virtual machine controls the CLS Operator Consoles, routes commands and responses, and keeps logs of what the CLS has done.

Common Library Services Operator Console (CLSOC)— A VM-attached console that is used by CLS operators to monitor CLS events and from which CLS-related commands are issued.

communication parameters— Keywords that need to be specified for a client's mode of access to CLS (VM/Pass-Through facility or TCP/IP).

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

configuration data base (CDB)— Data used by CLS to maintain the CLS configuration.

Configuration Management (CM)— A CLS program that provides a menu-driven facility for users to define and maintain CLS configurations.

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 program (CP)— The piece of the VM operating system that controls the real hardware, provides services to virtual machines so that they appear to be real machines, and provides the timesharing services on the processor.

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.

conversational monitor system (CMS)— A virtual machine operating system that provides a general interactive environment and operates only under the control of VM.

coupling facility— A special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex.(I)

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

coupling services— In a sysplex, the functions of XCF that transfer data and status between members of a group residing on one or more MVS systems in the sysplex.(I)

cross-system coupling facility (XCF)— XCF is a component of MVS that provides functions to support cooperation between authorized programs running within a sysplex.(I)

CTC— Channel-to-channel.

D

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

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

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

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

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

device separation— *See* drive exclusion.

DFSMS— Data Facility Storage Management Subsystem.

direct access storage device (DASD)— 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).

DMS/OS— DASD Management System/Operating System.

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. See the *SMC Configuration and Administration Guide* for more information.

drive panel— An LSM wall containing tape transports. The drive panel for a 9840 transport can contain either 10 or 20 transports. The drive panel for a non-9840 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. See the *SMC Configuration and Administration Guide* for more information.

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

Dual LMU VM/HSC— VM/HSC release 1.1.0 or later that automates a switchover to the standby LMU in a dual LMU configuration.

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

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

E

ECART— Enhanced Capacity Cartridge.

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

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

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

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

ESCON— Enterprise Systems Connection.

esoteric name— The name assigned to transports that have the same device type.

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

event control block (ECB)— Provides an area for a completion code to be stored when an operation has completed.

EXEC— VM CMS command.

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.

heartbeat interval— Specifies how often CLS checks the communications link to a client to make sure it is still "up."

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.

heterogeneous systems— Systems of dissimilar processor or system type.

homogeneous— Of the same or a similar kind or nature.

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.

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.

Intelligent Management Unit (IMU)— Hardware similar to a Control Unit. This term is reserved for future products.

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.

Inter-User Communication Vehicle (IUCV)— A CP communications facility that allows users to pass information between properly authorized virtual machines.

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— An installation of one or more ACSs, attached 4480, 4490, 9490, 9490EE, SD3, or 9840 cartridge drives (also known as transports), volumes (cartridges) placed into the ACSs, host software that controls and manages the ACSs and associated volumes, and the library control data set that describes the state of the ACSs.

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 16 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.

LibraryStation— Software that allows MVS hosts to share ACS facilities with client systems.

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 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 0 (hexadecimal) the second LSM listed acquires a number of 1, and so forth, until all LSMs are identified (up to a maximum of sixteen or hexadecimal F).

M

manual mode— Operation of a 4480, 4490, 9490, 9490EE, SD3, or 9840 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.

N

Nearline Storage Server— The hardware and software necessary to use ACS libraries by client computing systems.

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 document, 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.

pipe— VM Inter-User Communications Vehicle (IUCV) path.

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.

privilege class— Applicable to both the VM and CLS environments, userids are granted access to either system based on assigned rights to execute various commands.

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

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

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.

request status record (RSR)— An in-memory record, maintained by CLS, that tracks the status and disposition of each client request to the VM/HSC.

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 volsers with similar physical characteristics (type of volume—reel or cartridge, reel size, length, physical location, and so on). Some installations may also subdivide their scratch pools by other characteristics such as label type.

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

shadow recording— A technique for recovery involving maintaining both a control data set and a copy (shadow) of the data set.

signon script— A series of statements used by CLS to initiate or verify VM Pass Through communications with the CSC. Signon scripts are defined by the CLSCM EXEC.

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 OS/390 and z/OS operating systems and StorageTek real and virtual tape hardware. SMC performs the allocation processing, message handling, and SMS processing for the NCS solution. It resides on the MVS host system with HSC and/or MVS/CSC, and communicates with these products to determine policies, volume locations, and drive ownership.

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

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

synchronous— *See* BISYNC.

synchronous LAN— Local area network built on synchronous communications.

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

System Control Program (SCP)— A control program that provides the required environment in a virtual machine to run VM/HSC. The SCP is a component of the VM/HSC.

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 document, refers to any tape library management system, not to CA-1.

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

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

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

Transmission Control Protocol (TCP)— 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

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/Pass-Through Facility— IBM's software utility for implementing synchronous communications between CLS and client.

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 location record (VLR)— A record, maintained by the CLS system, that tracks the status of each volume from the time it is mounted until it is dismounted.

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.

X

XCF— Cross-system coupling facility.

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 from one to two hundred and fifty-six LMUs with each LMU connected to from one to sixteen 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— 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.

9840—The StorageTek access-centric cartridge transport capable of reading and writing 9840 cartridges.

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

Index

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X

Z

Printed in U.S.A.