
StorageTek[®]

PN 312537201

OS 2200 Client System Component
(CSC)

5R1 Operations Guide

February 2005

Copyright © 2005 by Storage Technology Corporation
All Rights Reserved

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

The information in this document is confidential and proprietary to Storage Technology Corporation, and may be used only under the terms and conditions of a Storage Technology Corporation license agreement. The information in this document, including any associated software program may not be disclosed, disseminated, or distributed in any manner without the written consent of Storage Technology Corporation.

THIS DOCUMENT DOES NOT EXTEND OR CREATE WARRANTIES OF ANY NATURE, EITHER EXPRESSED OR IMPLIED. Storage Technology Corporation cannot accept any responsibility for use of the information in this document or any associated software program. Storage Technology Corporation assumes no responsibility for any data corruption or erasure as a result of the information in this document, including any associated software program. You are responsible for backing up your data. You should be careful to ensure that use of the information complies with all applicable laws, rules, and regulations of the jurisdictions with respect to which it is used.

Unisys and EXEC are registered trademarks, as well as OS 2200 and 1100/2200, of Unisys Corporation. IBM is a registered trademark, as well as VM and MVS, of International Business Machines Corporation. Solaris is a registered trademark of Sun Microsystems, Inc. Ethernet is a registered trademark of Xerox Corporation. WolfCreek is a trademark, as well as StorageTek, of Storage Technology Corporation.

Information in this document is subject to change, and Storage Technology Corporation reserves the right to revise or modify the information in this document without prior notification. In the event of changes, this document will be revised. Contact Storage Technology Corporation to verify that you have the most current version of this document.

Comments concerning the information in this document should be directed to:

Storage Technology Corporation
2270 South 88th Street
Louisville, CO 80028-4232
(800) 678-4430

CONTENTS

PREFACE **XI**

Purpose xi
Audience xi
How to Use This Document..... xi
Command Syntax Notation..... xiii
Related Documentation..... xiii

1. OVERVIEW OF CSC **1-1**

Introduction to CSC..... 1-2
Features of CSC 1-2
What CSC Does 1-4

2. USING CSC **2-1**

Starting and Stopping CSC..... 2-2
 Starting CSC..... 2-2
 Stopping CSC..... 2-2
Monitoring the CSC and CDI Environment 2-3
 Console Interface to CSC..... 2-3
 Demand Terminal Security 2-3
 Rules to Consider When Changing Security Levels 2-5
Monitoring and Controlling the Control Path..... 2-6
 CSC Handling of Control Path Hardware 2-6
 Monitoring CSC 2-6
 Monitoring CDI..... 2-7
Establishing the Cartridge Drive Environment..... 2-9
 CULP Use at Boot Time 2-9
 Compatibility Between CULP and Non-CULP Systems 2-10
 Facility Commands 2-10
 Tape Automatic Volume Recognition (TAVR) 2-10
 Miscellaneous Environmental Concerns 2-11
Modes of Operation 2-11

Normal Operations	2-11
Manual Operations	2-11
Recovery	2-14

3. CDI COMMAND REFERENCE 3-1

BRKPT	3-2
DN.....	3-3
FS	3-4
HELP	3-6
ID	3-7
PING	3-8
STATUS	3-9
① TCP ABORT	3-10
TCP CONNECTIONS	3-11
TCP HELP	3-12
① TCP KILL	3-13
TCP STATUS	3-14
TERM	3-16
UP	3-17

4. CSC COMMAND REFERENCE 4-1

Locations in a Library	4-2
① ABORT	4-3
ACTIVATE	4-4
BRKPT	4-5
CLEAR	4-6
CMD (NCS only).....	4-7
① CYCLE.....	4-8
① DEBUG OFF	4-9
① DEBUG ON	4-10
① DEBUG STATUS	4-11
DISMOUNT	4-12
DOWN (DN).....	4-13
① EABT	4-14
EJECT	4-15
ENTER	4-16
HELP	4-17
LEVEL.....	4-18
MEMORY	4-19
MOUNT.....	4-20
QUERY.....	4-21
QUEUES.....	4-25
REPORT	4-26
STATUS	4-28
TERM	4-31
UP	4-32

VENTER.....	4-33
VERBOSE	4-34
VERBOSE CTL	4-35
VERBOSE EJECT	4-36
VERBOSE STATUS.....	4-38

5. MESSAGE REFERENCE **5-1**

Answerable Messages.....	5-2
Informational Messages	5-3
Subsystem Error Messages	5-17
Exec Messages.....	5-20

6. PROBLEM SOLVING **6-1**

How Failures Affect Automated Operations	6-2
Failure of ACS Requests	6-3
Failure of Control Path Components.....	6-5
Handling Abnormal Conditions	6-7
Data Path Conditions.....	6-7
Control Path Conditions	6-7
Handling Specific Error Conditions	6-10
Initialization Problems	6-10
Mount Problems	6-11
Dismount Problems	6-13
Write Ring Problems	6-14
Software Failures.....	6-15
Reporting CSC Problems.....	6-17
Useful Information for Problem Analysis	6-17
How To Collect Diagnostic Information.....	6-18
Starting CSCLOG and CSCDUMP.....	6-19
Alternatives for Collecting Diagnostic Information.....	6-19
What Diagnostic Information To Collect.....	6-20

APPENDIX A. CSC RUN DEFAULTS AND XQT OPTIONS **A-1**

CSC Run Defaults.....	A-2
CSC XQT Options	A-2

INDEX **A-1**

EFFECTIVE PAGES **A-1**

TABLES

Table 2-1. Demand Terminal Security Default Values.....	2-4
Table 3-1. *CDI FS Response Detail.....	3-5
Table 3-2. TCP Connection States.....	3-15
Table 4-1. *CSC STATUS Response Notes.....	4-29
Table 6-1. CSC Diagnostic Information and How it is Collected	6-19
Table A-1. CSC XQT Options.....	A-2

FIGURES

Figure 1-1. Component Overview	1-4
Figure 1-2. Top-Down Overview of CSC Control Path	1-6
Figure 3-1. *CDI BRKPT Command/Response Example	3-2
Figure 3-2. *CDI DN Command/Response Example	3-3
Figure 3-3. *CDI FS Command/Response Example	3-4
Figure 3-4. *CDI HELP Command/Response Example	3-6
Figure 3-5. *CDI ID Command/Response Example	3-7
Figure 3-6. *CDI PING Command/Response Example	3-8
Figure 3-7. *CDI STATUS Command/Response Example	3-9
Figure 3-8. *CDI TCP ABORT Command/Response Example	3-10
Figure 3-9. *CDI TCP CON Command/Response Example	3-11
Figure 3-10. *CDI TCP HELP Command/Response Example	3-12
Figure 3-11. *CDI TCP KILL Command/Response Example	3-13
Figure 3-12. *CDI TCP STATUS Command/Response Example	3-14
Figure 3-13. *CDI TERM Command/Response Example	3-16
Figure 3-14. *CDI UP Command/Response Example	3-17
Figure 4-1. *CSC ABORT Command/Response Example	4-3
Figure 4-2. *CSC ACTIVATE Command/Response Example	4-4
Figure 4-3. *CSC BRKPT Command/Response Example	4-5
Figure 4-4. *CSC CLEAR Command/Response Example	4-6
Figure 4-5. *CSC CMD Command/Response Example	4-7
Figure 4-6. *CSC CYCLE Command/Response Example	4-8
Figure 4-7. *CSC DEBUG OFF Command/Response Example	4-9
Figure 4-8. *CSC DEBUG ON Command/Response Example	4-10
Figure 4-9. *CSC DEBUG STATUS Command/Response Example	4-11
Figure 4-10. *CSC DISMOUNT Command/Response Example	4-12
Figure 4-11. *CSC DOWN Command/Response Example	4-13
Figure 4-12. *CSC EABT Command/Response Example	4-14
Figure 4-13. *CSC EJECT Command/Response Example	4-15
Figure 4-14. *CSC ENTER Command/Response Example	4-16
Figure 4-15. *CSC HELP Command/Response Example	4-17
Figure 4-16. *CSC LEVEL Command/Response Example	4-18
Figure 4-17. *CSC MEMORY Command/Response Example	4-19
Figure 4-18. *CSC MOUNT Command/Response Example	4-20

Figure 4-19. *CSC QUERY ACS/LSM/CAP/SERVER	
Command/Response Example	4-22
Figure 4-20. *CSC QUERY POOL/SCRATCH/REQUEST	
Command/Response Example	4-22
Figure 4-21. *CSC QUERY DRIVE/VOLUME	
Command/Response Example	4-23
Figure 4-22. *CSC QUERY LOCK-DRIVE/LOCK-VOLUME	
Command/Response Example	4-23
Figure 4-23. *CSC QUEUES Command/Response Example	4-25
Figure 4-24. *CSC REPORT ACS Command/Response Example	4-26
Figure 4-25. *CSC REPORT DRIVES Command/Response Example	4-27
Figure 4-26. *CSC REPORT POOLS Command/Response Example	4-27
Figure 4-27. *CSC REPORT FILES Command/Response Example	4-27
Figure 4-28. *CSC STATUS Command/Response Example	4-28
Figure 4-29. *CSC TERM Command/Response Example	4-31
Figure 4-30. *CSC UP Command/Response Example	4-32
Figure 4-31. *CSC VENTER Command/Response Example	4-33
Figure 4-32. *CSC VERBOSE Keyin Forms	
Command/Response Examples	4-34
Figure 4-33. *CSC VERBOSE Command/Response Example	4-35
Figure 4-34. *CSC VERBOSE EJECT Output Example	4-37

PREFACE

PURPOSE

This is the *OS 2200 Client System Component (CSC) Operations Guide*. This guide provides instructions and reference material for operators using CSC. CSC is the software used by the Unisys 1100/2200 Client System (the “client”) to communicate with the Solaris[®]-based Library Control System or the Nearline Control Solution (the “server”) and the Automated Cartridge System (ACS).

AUDIENCE

This guide is written primarily for *operators*. It assumes that you are familiar with the following hardware and software components:

- ACS
- Unisys Series 1100 and 2200 computers
- Solaris-based Library Control System
- Nearline Control Solution

HOW TO USE THIS DOCUMENT

Chapter 1. Overview of CSC

This chapter provides a “management summary” of CSC. Commands, operations, and interactions with other components are presented in a high-level macro view.

Chapter 2. Using CSC

This chapter describes the most important aspects of operating CSC. Topics include starting and stopping CSC, monitoring CSC and its environment, and modes of operation. Read this chapter to learn the basics of day-to-day CSC operations, then use the reference chapters that follow for more detailed discussions of specific commands and messages.

Chapter 3. CDI Command Reference

This chapter presents reference material on all CDI commands, organized alphabetically by command for quick access.

Chapter 4. CSC Command Reference

This chapter presents reference material on all CSC commands, organized alphabetically by command for quick access.

Chapter 5. Message Reference

This chapter presents reference material for the most common messages you might encounter while using CSC. The discussions explain the messages, and suggest possible responses. In this chapter, messages are grouped by message type (answerable, informational, and 4780) and then alphabetically or numerically within that message type.

Chapter 6. Problem Solving

This chapter describes how to solve problems with CSC and the server. Topics include how failures affect cartridge tape mounts, how to handle abnormal conditions, how to handle specific error conditions, and how to use diagnostic tools.

Back Matter

This guide also includes an appendix listing CSC run defaults and XQT options, and an index.

COMMAND SYNTAX NOTATION

This manual uses the following conventions for representing command syntax notation and message displays:

UPPERCASE	indicates a command or keyword
<i>lowercase italic</i>	indicates a user- or system-supplied variable value. For example, in <i>XX=userid</i> , you enter the actual <i>userid</i> for <i>userid</i> .
abbreviation	indicates a command that can be abbreviated to its minimum acceptable form. For example, ENABLE can be abbreviated to ENA.
vertical bar	separates operand alternatives. For example, A B indicates that you must select either A or B.
brackets []	indicate an option that can be omitted. For example, [A B C] indicates that you can select A, B, C, or nothing.
braces {}	indicate an option that you <i>must</i> choose. For example, {A B} indicates that you must choose either A or B.
<u>underlining</u>	indicates the system default. If you do not enter a parameter or value, the system will supply the underscored value. For example, A B <u>C</u> indicates that if you do not choose an option, the system will default to C.
ellipses ...	indicate that entries can be repeated as often as necessary.
SMALLCAPS	indicate a key, such as XMIT or F1.

RELATED DOCUMENTATION

OS 2200 Client System Component (CSC) Technical Bulletin, Storage Technology Corporation (312537701)

OS 2200 Client System Component (CSC) Installation Guide, Storage Technology Corporation (313471401)

OS 2200 Client System Component (CSC) System Administrator's Guide, Storage Technology Corporation (312537501)

OS 2200 Client System Component User Interface (CSCUI) Programmer's Reference Manual, Unisys Corporation (7844 8677)

OS 2200 Client System Component (CSC) Client Direct Interconnect (CDI) Troubleshooting Guide, Storage Technology Corporation (312537601)

OS 2200 Client System Component (CSC) User Reference Manual, Storage Technology Corporation (312537801)

OS 2200 Client System Component (CSC) UNISYS OS 2200 CSC 5R1 VSM Reference, Storage Technology Corporation (312537901)

Automated Cartridge System Library Software Product Document Set for Solaris, Storage Technology Corporation.

Nearline Control Solution 4.0 Publication Kit, Storage Technology Corporation (313456301)

Exec System Software Operations Reference Manual, Unisys Corporation (7831 0281)

Executive Control Language (ECL) and FURPUR Reference Manual, Unisys Corporation (7830 7949)

COMUS End User Reference Manual, Unisys Corporation (7830 7758)

Software Library Administrator (SOLAR) End User Reference Manual, Unisys Corporation (7831 0604)

Communications Management System (CMS 1100) Operations Reference Manual, Unisys Corporation (7831 5694)

ClearPath HMP IX Series Cooperative Processing Communications Platform (CPCOMM) Configuration and Operations Guide, Unisys Corporation (7844 8438)

1. OVERVIEW OF CSC

This chapter presents an overview of CSC and its relationship with the Solaris-based Library Control System or the Nearline Control Solution (the “server”) and the Automated Cartridge System (ACS). This macro view explains CSC concepts, features, and operations from a high-level perspective.

INTRODUCTION TO CSC

Traditionally, mainframe tape operations have been a manual process. A human operator, responding to requests from user applications, manually mounts tapes, dismounts tapes, and maintains information about the site's tape library (e.g., tape location, scratch status, write-protect status, etc.). The ACS is a tape storage and retrieval device that robotically mounts and dismounts cartridge tapes. The ACS, combined with several software components, automates this formerly manual process.

CSC—the Client System Component—is a collection of software components used by the Unisys client to communicate with the server and the ACS. With CSC, a server, and the ACS, most manual tape operations become unnecessary. This helps reduce operations costs, minimize errors, and provide faster, more reliable tape services to system users.

FEATURES OF CSC

CSC doesn't just automate tape operations. It also provides communication tools, diagnostic utilities, and command-activated functions that give you maximum control with minimum effort. CSC features include:

- **Installation.** Using standard COMUS or SOLAR dialogues, you can install CSC simply by responding to a series of prompts.
- **Recovery.** CSC is aware of the tape processing states maintained by the client and server. When external events disrupt the synchronization of these states, CSC initiates a recovery process to restore automated operations.
- **Manual mount/dismount processing.** CSC allows you to perform tape operations manually at any time by issuing simple commands through the 2200 System Console (the “system console”) or a properly configured demand terminal.
- **Programmatic user interface.** The programmatic user interface to CSC lets user applications, such as a Tape Library Management System (TLMS), exchange information with the server and monitor ACS activities.
- **Operator interface.** CSC provides a command and control interface through the system console. An operator can receive and respond to critical messages, check the status of communications, and issue tape activity commands directly from the system console.

- **Diagnostics.** CSC diagnostics can be used to turn specific debug flags on or off, breakpoint print files, and display a variety of program statuses. These features can help you monitor the CSC environment and determine if CSC is operating efficiently.
- **Communication link.** CSC communicates with the server via a LAN, using either the Client Direct Interconnect (CDI) communications method or the Transport Service Access Method (TSAM) which supports CMS and CPCOMM.
- **Multiple ACS support.** If your site has more than one ACS, CSC configurations help you direct cartridge tape allocations to the correct ACS. CSC allows you to define ACS-names to be optionally used in ECL assignment statements. Other configuration options define defaults.
- **Multiple Media Support.** If you have an ACS with more than one type of tape drive, or more than one type of recording media, CSC can help you direct cartridge tape allocations to the correct drive type and/or media type. CSC allows you to define a media type to be used for specific scratch pools, and to define a scratch pool to be used for specific drive types within an ACS.
- **Multiple Server Support.** You can execute Multiple instances of CSC on the same system in order to communicate with different servers, each controlling it's own ACS. In addition to the existing STK\$ Exec interface, CSC also uses the CARTTAPELIB\$ Exec interface, which supports up to four cartridge tape library servers like CSC.

WHAT CSC DOES

In this discussion, we'll explain how CSC interacts with the various components of the system from a high-level perspective. You should first understand the basic components involved. If you don't, Figure 1-1 presents an overview.

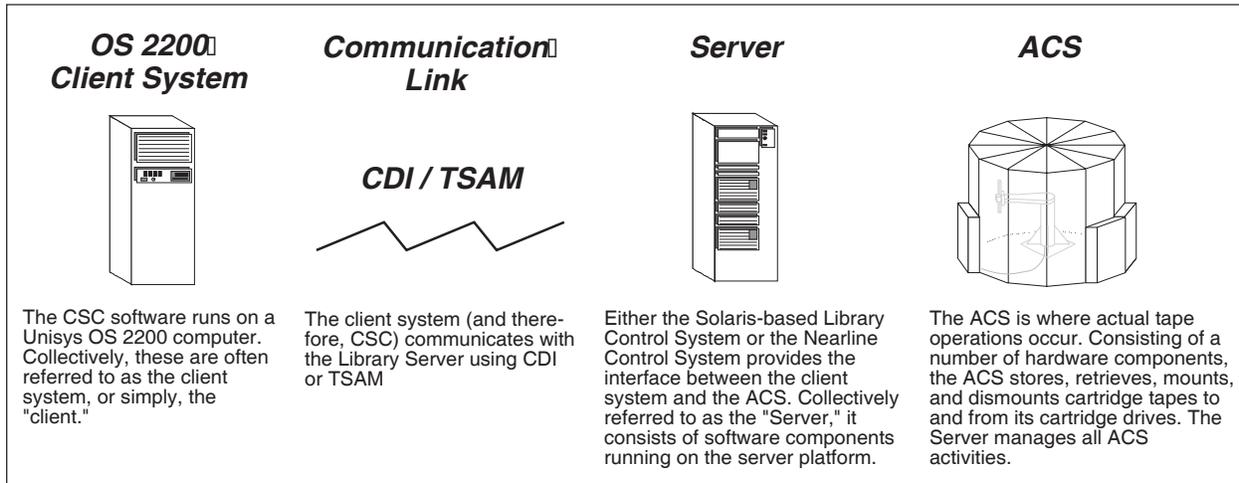


Figure 1-1. Component Overview

How CSC Automates Operations

Because CSC is designed to automate most cartridge tape activities via the ACS, most CSC operations are transparent to the operator. In general, the only time manual intervention is required is when something goes wrong with one or more interrelated components. Otherwise, you'll use CSC to view and respond to messages, check system statuses and operating conditions, and perform diagnostics.

For example, when a user requests a cartridge tape mount, CSC automatically translates and processes the request. If an error occurs—say, the cartridge tape must be entered into a different ACS—CSC will display a message on the system console. Using standard CSC commands, you can diagnose the condition and correct it as needed.

User applications, such as a TLMS, interface with CSC via the Client System Component User Interface (CSCUI). CSCUI allows user applications to make requests for volume scratch status changes, volume information, and system statuses.

If any system component fails, CSC will attempt to recover and restore all components to their pre-failure states. In certain cases, this is not always possible.

How CSC and the ACS Communicate

There are two possible interface types between the client and the server, depending on the software used.

- The Client Direct Interconnect (CDI) provides TCP/IP services and connections to Control Path Adapters (CPAs) that attach to the server.
- The Transport Service Access Method (TSAM) interface provides TCP/IP services via CMS 1100 or CPCOMM and any of the network attachment hardware supported by these programs. CSC supports two CMS interfaces and four CPCOMM interfaces.

The server is a collection of hardware and software components that are used to manage all ACS activities. CSC 5R1 works with either the Solaris-based Library Control System or with the Nearline Control Solution server.

The Solaris-based server software is collectively known as Automated Cartridge System Library Software (ACSLs). Its major components include the Network Interface (NI), the Client System Interface (CSI), and the ACS Library Manager (ACSLM). ACSLS also provides an operator interface to the server: the ACS System Administrator (ACSSA)/Command Processor. The Command Processor allows operators to monitor and control server operations.

The Nearline Control Solution (NCS) is an MVS software package that allows multiple systems to share an ACS complex. It consists of the Host Software Component for MVS (MVS/HSC), the Client System Component for MVS (MVS/CSC), and the Library Station feature. NCS software typically executes on an existing MVS platform within a computing facility. The Library Station feature allows non-MVS clients to share the ACS complex. CSC uses Library Station to communicate with NCS.

The ACS is where all automated cartridge tape activity occurs. The ACS is sometimes referred to as the “silo,” and consists of one or more Library Storage Modules (LSMs). Each LSM contains a robotic arm that moves bar-coded cartridge tapes from storage cells in the LSM to and from cartridge drives. The ACS has its own control hardware called the Library Management Unit (LMU). The LMU provides the connection to the server.

Tape requests travel from CSC on the client, through CDI or TSAM, to the server. The server then passes the requests to the LMU, which in turn instructs the ACS to perform the actual operation(s). Notifications travel back to CSC along the same path, but in the opposite direction (from the ACS to CSC).

Figure 1-2 provides a top-down overview of CSC, server, and ACS components.

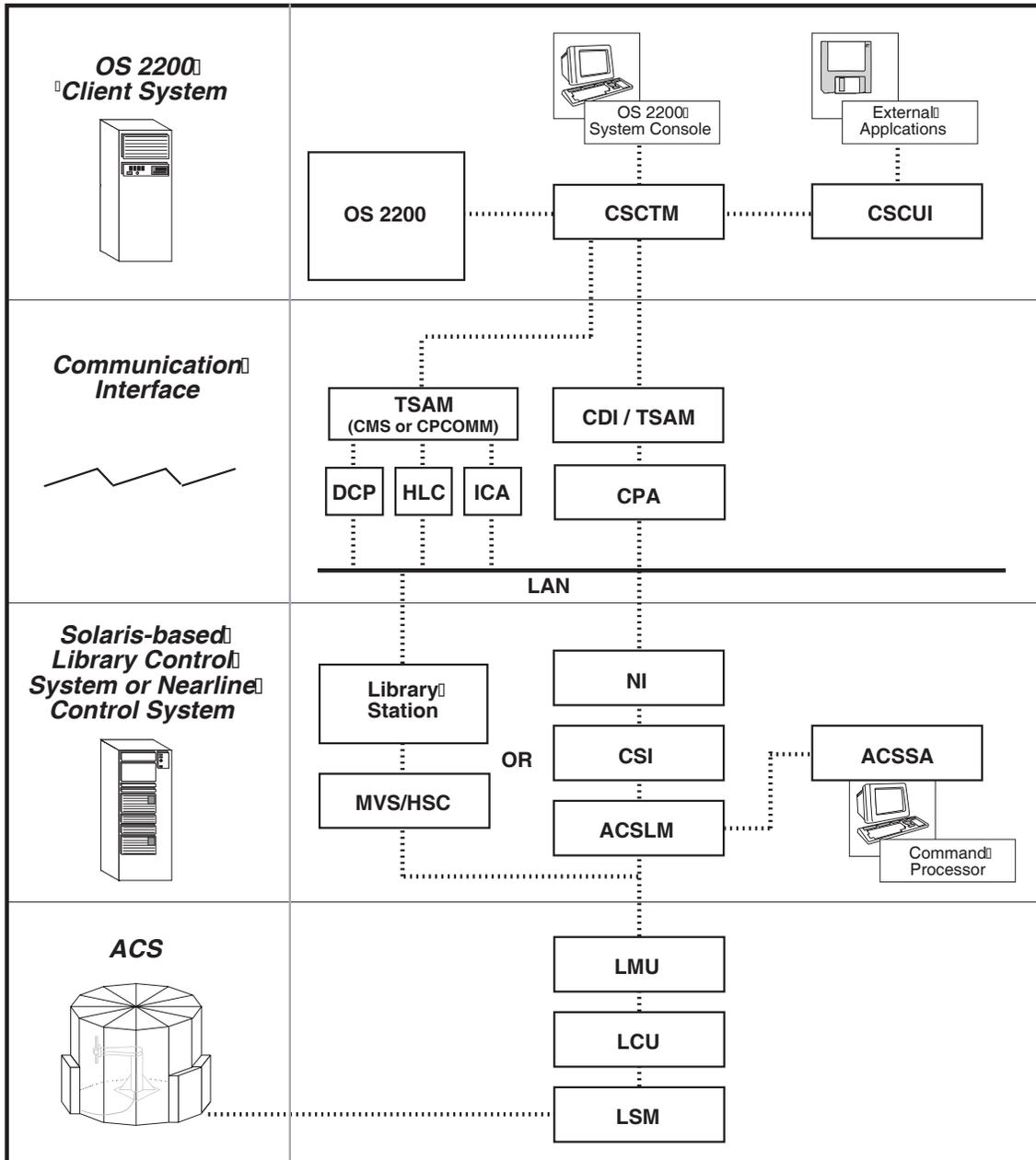


Figure 1-2. Top-Down Overview of CSC Control Path

Multiple CSC Considerations

CSC communicates with OS 2200 using either an Executive Request or the CARTTAPELIB\$ extended mode call. At any given time, OS 2200 can support one CSC via the ER interface and up to four CSCs via the CARTTAPELIB\$ interface. Each of these CSCs communicates with one library server. Using multiple CSCs introduces the following operational concerns:

- All CSCs can share a single instance of CDI. That instance of CDI exists in only of the CSC installations.
- Each CSC using CMS or CPCOMM must have a unique TSAM process.
- Each CSC that uses the CARTTAPELIB\$ interface is identified by a site defined CTL identifier. These CTL ids must be unique among the CSCs. This identifier is used in the CTL keyin to OS 2200.
- Each CSC must have a unique console keyin. This keyin is displayed on the console during CSC initialization. You can also find the keyin by issuing an II csc-runid command.
- When OS 2200 needs to know where to mount a tape, it queries the CARTTAPELIB\$ connected CSCs first and the ER connected CSC last. The CARTTAPELIB\$ CSCs are polled in the order that they appear in the output of the CTL FS console keyin. This order is established by the order in which the CSCs are started.

2. USING CSC

This chapter describes the most important aspects of operating CSC. Topics include starting and stopping CSC, monitoring CSC and its environment, and modes of operation. Read this chapter to learn the basics of day-to-day CSC operations. Then, use the reference chapters that follow for more detailed discussions of specific commands and messages.

Before performing any of the procedures in this chapter, you should already have installed and configured CSC. You should have also verified that all the components of your system are operational. For details on installation and configuration, please refer to the *CSC Installation Guide*.

STARTING AND STOPPING CSC

This section describes the procedures for starting and stopping CSC.

Starting CSC

Follow these steps to start CSC:

1. If CSC is already running, stop it using the procedures described in the next section, "Stopping CSC."
2. If the previous CSC run terminated in error and if during the COMUS BUILD of CSC you opted to use CSCUI, you should reload the following common banks:
 - CSCUICBA
 - CSCUICBB
3. If CDI is used, START the CDI run.
4. If CMS is used, make sure it is up and operational.
5. If CPCOMM is used, make sure it is up and operational.
6. START the CSC run.

Stopping CSC

Follow these steps to stop CSC:

1. First, terminate CSC by entering:
*CSC TERM
2. Next, if CDI is used, terminate CDI by entering:
*CDI TERM

MONITORING THE CSC AND CDI ENVIRONMENT

This section describes how you can configure the CSC operating environment and use CSC commands to monitor its operation.

Console Interface to CSC

Console operators communicate with CSC through the system console, or through a demand terminal in “console mode.”

The system console is your primary interface to CSC. Use it to initiate all CSC commands and utilities described in this manual.

A demand terminal is a secondary interface to CSC. If your demand terminal has console privileges, you can use it to enter *CSC commands.

NOTE

*CSC console commands begin with “*CSC” and CDI console commands begin with “*CDI.” These are the command names and not the runids. These command names can be changed when CSC is installed. If you do not know the command name associated with a specific CSC runid, you can issue II runid from the system console, and CSC will display its command name.*

Demand Terminal Security

Demand terminal security is a feature that can be controlled by your security officer. The execution of CSC and CDI commands from a demand terminal in console mode (@@CONS), will be regulated by global security privileges. These privileges are established during the product installation process, and can be configured any time after the installation. Authorized security levels include:

- | | |
|-------------|--------------------|
| 1 = Basic | 4 = Display |
| 2 = Limited | 5 = Response |
| 3 = Full | 6 = System Console |

Table 2-1 indicates the default values established for each CSC and CDI command during the product installation process.

Table 2-1. Demand Terminal Security Default Values

If your security level is...	You can execute the following commands for...	
	CSC	CDI
1 or higher	HELP, LEVEL, MEMORY, QUEUES, REPORT, STATUS, VERBOSE	BUFFER, FLAGS, FS, HELP, ID, STATUS
3 or higher	N/A	PING
5 or higher	ABORT, ACTIVATE, BRKPT, CLEAR, CMD, CYCLE, DEBUG, DISMOUNT, DOWN, EABT, EJECT, ENTER, MOUNT, QUERY, TERM, UP, VENTER	BRKPT, CLR, DN, DUMP, SET, TCP, TERM, UP

Changing Security Levels

Demand terminal security levels can be overridden by editing the CSC\$PARAM and CDI\$PARAM elements found in the CSC parameter file. This file is specified in the CSCPARAM parameter during the product build process. The default filename is SYS\$LIB\$*CSC-PARM. To configure these elements, you can use the @ED processor or any other editor that creates ASCII elements. For example, to change the security level for the *CSC QUEUES and *CSC STATUS commands from level 1 to level 2, you must first use your site's procedure for entering privileged user mode.

The next step is to call the editor and update the CSC\$PARAM element. For instance, you might enter the following statement:

```
@ED,U SYS$LIB$*CSC-PARM.CSC$PARAM
```

Display the entire element, and insert the following parameter at any line within the element:

```
SEC_LEVEL_2=QUEUES, STATUS
```

This new parameter will override the previously established default security levels for *CSC QUEUES and *CSC STATUS only. You can insert as many parameter lines as you want. When you're finished, exit the editor and save your updates. If CSC was active when the changes were made, you will have to terminate and restart CSC before the new security levels take affect.

To change the security level for a *CDI command, you would repeat the above process, but substitute CDI\$PARAM for CSC\$PARAM. For example, to change the security level for the *CDI HELP and *CDI STATUS commands from level 1 to level 2, call the editor and update the CDI\$PARAM element. For instance, you might enter the following statement:

```
@ED, U  SYS$LIB$*CSC-PARM.CDI$PARAM
```

Display the entire element, and insert the following parameter at any line within the element:

```
SEC_LEVEL_2  HELP, STATUS
```

This new parameter will override the previously established default security levels for *CDI HELP and *CDI STATUS only. You can insert as many parameter lines as you want. When you're finished, exit the editor and save your updates. If CDI was active when the changes were made, you will have to terminate and restart CDI before the new security levels take affect.

Rules to Consider When Changing Security Levels

When editing the CSC\$PARAM and CDI\$PARAM elements to override default security levels, consider the following rules:

- When a default security level is overridden, the change is applied globally to all demand terminal users. There is no option to change a security level for an individual user.
- When editing CDI\$PARAM, use a space to separate the parameter from the commands. When editing CSC\$PARAM, use an equal sign (“=”) to separate the parameter from the commands.
- Spaces are not allowed before or after commas.
- If CSC and CDI are active at the time you edit the CSC\$PARAM and/or CDI\$PARAM elements, it will be necessary to terminate and restart CSC and CDI before the changes take affect.
- If a CSC or CDI command entered on a demand terminal is misspelled or the syntax is incorrect, you will not receive an error or warning.

MONITORING AND CONTROLLING THE CONTROL PATH

The control path includes the CSC interface software components (CSCTM and CDI or CMS or CPCOMM); the hardware that connects the Unisys 2200 to the server; and the software executing on the server platform. Each of the software components allows the operator to monitor and control its operation. This section describes the common operational practices used to accomplish this.

CSC Handling of Control Path Hardware

The CSC software supports one or more physical paths to the server. One feature of CSC 5R1 is code that looks for and tries to circumvent physical path failures. CSC cannot correct path failures. It can only notify the operator when one is detected.

The following sections highlight commands that allow the operator to monitor and control the CSC control path.

Monitoring CSC

Once CSC starts and establishes a connection with the server, you can issue several commands to assist you in monitoring the CSC environment. Remember to preface all CSC commands with “*CSC” (or the site-defined CSC command name) regardless of CSC’s runid.

Obtaining CSC Status Information

To display general CSC status information, enter:

```
*CSC STATUS
```

The STATUS command displays the current state of CSC, the total number of active tasks, the total number of configured tasks, the type of server, the type of interface to Exec, the number of mounts completed since CSC initialized, and the path used for communications with the server.

The state of CSC can be INITIALIZING, ACTIVE, or TERMINATING.

A variation of this command describes the status of each interface or path to the server in addition to the preceding information. The command to display the status of each interface is:

```
*CSC STATUS INTERFACE
```

To display the status of each path to the server:

```
*CSC STATUS PATH
```

Controlling the control path

CSC is aware of the paths to the server. It periodically tests server communication through each available path. If communication through the currently active path is interrupted, CSC tries to switch to another path. Path switching can also be requested using the following CSC command:

```
*CSC ACTIVATE path-name
```

If your site has more than one interface to the server, you can switch to a path through another interface program using the following command:

```
*CSC ACTIVATE interface-name
```

Checking the Status of CSC Drive Queues

CSC maintains a request queue for each active cartridge drive. To display CSC drive queues, enter:

```
*CSC QUEUES
```

The resulting display includes the cartridge drive name and location, the volume serial number (volser) of the cartridge tape in use on the drive, the number of requests queued for that drive, and the current request. Drive requests exist only when a volume mount or dismount is in progress.

Obtaining Information From the Server

CSC allows you to obtain some status information from the server through the system console. You may obtain more server information using a console or terminal attached to the server. Both servers have commands to display status information about library components and cartridge tapes, as well as status information on system settings and specified parameters. Refer to the administrator and operator references for the server for more information about the specific commands.

With the Nearline Control Solution, CSC may be configured to allow arbitrary server requests from the client system console. These server requests are submitted using *CSC CMD.

Monitoring CDI

Several CDI commands help you monitor your CDI environment. (You preface all CDI commands with “*CDI”.)

Displaying CDI Status Information

To display general CDI status information, enter:

```
*CDI STATUS
```

This command indicates the status of CDI, the status of each configured CPA, and the Internet address of each network interface. If a CPA is DOWN to CDI, CSC will not try to use it.

Displaying Network Interface Status

To display the current status of the network interface, enter the name of the CPA in the following command:

```
*CDI FS cpa_name
```

This command displays the status of the specified CPA, the internal option settings, the Internet address assigned to the CPA, the number of packets and bytes transmitted and received on the interface, and the number of errors occurring on the interface. Errors fall into the following categories:

- **FORMAT.** The number of error packets reported by the interface. These include the sum of frames with Frame Check Sequence (FCS) errors, frames that were too short, and other format errors.
- **HEADER.** The number of frames not acceptable because of an invalid LAN packet header.
- **NO BUFFER.** The number of packets discarded by the interface because buffers were not available. This condition occurs when data packets are not read from the interface.
- **TIMEOUT.** The number of times, within a specified period, that data was not received from the LAN. This is not actually an error: There simply may be no data to transfer.
- **COLLISIONS.** The number of times that the interface started sending data at the same time as an incoming data frame was starting.

Controlling the Network Interface State

To bring UP a specific interface, enter the name of the CPA in the following command:

```
*CDI UP cpa_name
```

This command displays the Power-On Confidence (POC) test progression messages, the CPA operating mode, the hexadecimal LAN address, the

equipment name, internal option settings, and the Internet address associated with the interface.

To bring DOWN a specific CPA, enter:

```
*CDI DN cpa_name
```

This command displays the CPA name, internal option settings, and notification that the interface is down. CSC will not try to use an interface that is DOWN.

Transmission Control Protocol (TCP) Commands

To display a list of available TCP commands, enter:

```
*CDI TCP HELP
```

For information about all TCP connections, enter:

```
*CDI TCP CON
```

ESTABLISHING THE CARTRIDGE DRIVE ENVIRONMENT

Before you alter the cartridge drive environment, you should have an understanding of the forces that control the environment.

OS 2200 includes an electronic partitioning feature called Control Unit Level Partitioning (CULP). CULP uses a hardware feature of the control unit to prevent multiple clients from concurrently accessing each cartridge drive. The control unit implements this partitioning in response to a request:

- to obtain exclusive use of a cartridge drive
- to release a cartridge drive from exclusive use

OS 2200 issues these commands automatically when the client is booted, and when you enter certain facility commands. Control units maintain cartridge drive states, and permit or refuse client system access to the cartridge drive.

CULP Use at Boot Time

At boot time, OS 2200 tries to exclusively assign each cartridge tape drive that is not DN. If a cartridge drive can't be exclusively assigned because it was previously assigned to another client, OS 2200 will display an error message and down the drive. (Chapter 5, "Message Reference," lists all error messages created by standard Unisys operating system code that apply to ACS cartridge drives.)

Compatibility Between CULP and Non-CULP Systems

If your site shares cartridge drives among multiple clients, you should consider the capabilities of each of those clients. One possibility is that the cartridge drives are shared by clients that support CULP, and by clients that don't. In this case, when you perform an UP command of a cartridge drive on 2200 system, that drive becomes exclusively assigned to the 2200 system. If the UPed cartridge drive is currently in use on a client without CULP, that original client would no longer be allowed to access that cartridge drive. This will interrupt work in progress on that cartridge drive on the client without CULP.

Facility Commands

Once the client is operational, you use facility commands of OS 2200 to control the assignment of each cartridge drive. These commands and their CULP relationships are:

- **UP/RV.** Assigns the cartridge drive if the command is at the cartridge drive level. If the command is the level of I/O processor (IOP), path, or control unit, CULP assigns all cartridge drives that are configured under the higher-level component. If CULP cannot assign the cartridge drive, the UP or RV command is not performed, and CULP returns the cartridge drive to the DN state.
- **DN.** Releases exclusive use of the cartridge drive for device-level commands.
- **TU.** Allows you to “take” a cartridge drive assigned to another client. Use this command when a cartridge drive assigned to another client can't be released by the owner-client (e.g., when the owner-client is DOWN). The TU command does not function on 4780 cartridge drives with microcode levels prior to 4.1-7.
- **PM.** Used on clients with Tape Automatic Volume Recognition (TAVR) to designate a cartridge drive as a premount-only device. A premount-only cartridge drive can be assigned *only* when the drive is premounted or aftermounted with the requested cartridge tape. An absolute assignment overrides the premount-only designation.

Tape Automatic Volume Recognition (TAVR)

TAVR is an OS 2200 feature that detects the presence of a cartridge tape mounted on a cartridge drive, then processes the cartridge tape label. TAVR lets you select the cartridge drive on which to mount a cartridge tape, either before or after a load message. This flexibility allows you to determine where and when to mount cartridge tapes.

TAVR plays an important role during manual operations. For example, when a cartridge tape mount is required and you're in manual mode on a client with

TAVR, you can mount the cartridge tape on any compatible cartridge drive (instead of the cartridge drive designated by OS 2200).

Miscellaneous Environmental Concerns

See the *CSC System Administrator's Guide* for information on:

- Entering cartridge tapes into the ACS
- Initializing new cartridge tapes
- Ejecting cartridge tapes from the ACS

MODES OF OPERATION

This section describes the different modes of operation that affect CSC and its environment. Included is a discussion of the CSC recovery process.

Normal Operations

Under normal conditions, all cartridge tape activity is automated and transparent. CSC and other components select, mount, and dismount cartridge tapes, and perform read/write operations automatically. These actions occur through two activity paths: the data path and the control path.

The data path transfers data between a user program and a cartridge tape on a cartridge drive. The client system contains the hardware and software components that allow this exchange of information to occur.

The control path manages the physical aspects of cartridge tape handling, including selection, mounting, dismounting, and write protection. The control path begins with the client's operating system and ends with the hardware in the ACS.

CSC is a real-time program that runs under OS 2200 supervision. A real-time program is one that must respond to a request in a specific amount of time, or the request will fail. To avoid failures, OS 2200 gives CSC a very high priority. This lets CSC manage control path functions without noticeable delays.

Manual Operations

When CSC is down or unavailable, or a communications problem prevents CSC and the server from exchanging information, you must revert to manual operations. (In general, the terms *manual* and *non-automated operations* are synonymous.) Manual operations fall into two categories: *console-initiated* and *physical*.

Console-initiated operations occur when you perform MOUNT or DISMOUNT commands using the system console or the ACSLS or Command Processor or the NCS console. Physical operations occur when you must step inside the LSM to

physically load or unload a cartridge tape. For convenience, we refer to both of these manual methods as manual mounts and dismounts.

Usage Guidelines

The server maintains a database containing location and usage information for each cartridge tape. This database resides within the server.

To ensure the integrity of the server database, follow these guidelines

Avoid manual operations unless automated operations are seriously disrupted.

Avoid physically entering the LSM. If you *must* enter the LSM, avoid physically moving cartridge tapes if possible. When unloading cartridge tapes, you should physically remove them from the LSM. After resuming normal operations, use the ENTER command to get the cartridge tapes back into the LSM.

If you don't use the ENTER command and you physically place a cartridge tape in a different storage cell, the server cannot find that cartridge. This will prevent further automated operations involving that cartridge.

Perform a server AUDIT to ensure that the server database reflects the proper physical location of all cartridge tapes. The server takes several seconds to audit each location. Auditing an entire LSM may take several hours. The AUDIT performs a physical inventory on one or more of the following:

- A sub-panel within an LSM
- A panel within an LSM
- An entire LSM
- All LSMs in an ACS

Mounting Cartridge Tapes Using the Command Processor

If any component in the control path isn't available, OS 2200 will continue to display service messages for cartridge tape requests on the system console. If the server and ACS are operational, you can perform manual operations through the server console using the MOUNT and DISMOUNT commands.

If your site is running TAVR, you have the flexibility to specify any available cartridge drive in the ACS containing the requested volume. When you enter the MOUNT command, TAVR lets OS 2200 detect the cartridge tape load and process the cartridge tape.

If a cartridge tape in a multi-ACS environment is in the wrong ACS and your site does not have TAVR, use the Command Processor ENTER and EJECT commands to move the cartridge tape to the correct ACS.

Physically Mounting Cartridge Tapes

If the robotic system within the ACS is down or unavailable or if the server is not operational, you may be able to physically enter the LSM to load and unload cartridge tapes.

Use this mode of operation only when you have no alternative. If you must physically load cartridge tapes, read the following sections for important information about restoring the ACS environment.

Finding Cartridge Tapes Within the ACS

To determine the cell location of a cartridge tape, you can produce a printed volume report using the server VOLRPT utility. You should periodically generate a volume report so that you are better prepared if you need to physically find and load cartridge tapes inside the ACS.

If the server is available, a server command may be used to display the location of a cartridge. This can be done with the *CSC QUERY VOLUME command or the equivalent command from the server.

Tape locations within a library consist of a 4 or 5 part number for volumes that are mounted or in their home cells respectively. The first two parts are the ACS number and LSM number. These are determined by the physical and electrical configuration of the LSMs at your site. The third part is the panel number within the LSM where the volume is located. These can be found on labels inside of the LSM. For tape drives, the fourth number tells the drive number counted from the topmost drive which is number 0. For storage cells, the fourth and fifth numbers are the row and column of the storage cell. These are identified on labels near the panel label mentioned above.

As an example, you would use the following steps to find the tape volume at location 1,2,3,4,5:

- Find LSM number 2 in ACS number 1.
- Enter the LSM and locate panel number 3.
- Count down 4 tape rows.
- Count across 5 cells. You should find the desired tape in this storage cell.

Physically Mounting Cartridge Tapes Within the ACS

To physically mount a cartridge tape within the ACS, open the LSM door and walk inside. For detailed instructions about the safe and proper way to physically enter the LSM, refer to the Operator's Guide for your particular LSM.

Then, using the cell address found above, find the cartridge tape and load it into the cartridge drive specified in the service message.

Alternatively, if your site has TAVR, use any available cartridge drive to enable OS 2200 to detect the cartridge tape load and process it (this does not apply to scratch mount requests).

After the user program finishes with the cartridge tape, OS 2200 rewinds it. You must then physically unload the cartridge tape. *Rather than return the cartridge tape to its location, remove the cartridge tape from the LSM. Once the LSM is fully operational, use the ENTER command to reintroduce the cartridge tape into the LSM.*

Remember that ACS integrity depends upon the ability of the server software to accurately identify the cell location of each cartridge tape through the server database. This is not possible if you physically place a cartridge tape in a different cell location from the position registered in the server database.

If you do physically move a cartridge tape within the ACS, you should then run an AUDIT to reestablish the accuracy of the server database.

Recovery

As mentioned earlier in this chapter, recovery is the process of reestablishing automated operations after a component failure. Recovery can occur after any hardware or software failure that disables CSC, ACSLS or NCS, and/or the control path (communications).

CSC requires very little operator intervention when a recovery is attempted. When initialized, CSC attempts to reestablish its pre-failure environment. This is done dynamically through a series of requests that are sent to both the server and to OS 2200. These requests assist CSC in determining which cartridge tapes are mounted in which cartridge drives. CSC uses status information returned by the server and OS 2200 to reestablish its internal cartridge drive information.

Conversely, the server maintains a database to facilitate recovery. The server database contains location and usage information for each cartridge tape in an LSM. It uses a combination of checkpointing and journaling functions to accomplish error recovery. The database resides within the server.

For more information on software failures and recovery processing, please refer to Chapter 6, "Problem Solving."

3. CDI COMMAND REFERENCE

This chapter presents reference material on all CDI commands. It is organized alphabetically by command.

Each command begins on a new page, and includes:

- a description of the command, its uses, and its results.
- a syntax description.
- a representation of a console display, showing the command on the top line and its response (if any) on subsequent lines. On multi-line displays, each line is numbered for quick reference.
- any miscellaneous usage notes.

For more comprehensive information about using CDI commands to correct problem situations, please refer to the *CDI Troubleshooting Guide*.

NOTE

*All CDI console commands in this manual begin with “*CDI”. If your site has changed the CDI command name, use your site’s command instead of *CDI .*

A Note About Diagnostic Commands ⓘ

Included in this chapter are a number of diagnostic commands, designated by the ⓘ symbol to the left of the command name. These commands are for debugging purposes only. These commands include:

TCP ABORT
TCP KILL

BRKPT

Requests that OS 2200 breakpoint the PRINT\$ file of the CDI batch run. *This file is used for diagnostic purposes and should be submitted to your support organization when reporting problems.*

Syntax

```
*CDI BRKPT
```

```
1 ► *CDI BRKPT
    CDI PRINT$ FILE HAS BEEN BREAKPOINTED
```

Figure 3-1. *CDI BRKPT Command/Response Example

DN

Brings down the selected network interface. The name of the interface is defined in the CPA SGS specified in CDISPARAM element. (You can use the *CDI STATUS command to list all configured CPAs.)

Syntax

```
*CDI DN cpa_name
```

where *cpa_name* is the name of the CPA to be brought down.

```
1 ► *CDI DN CPA0
2 ► TCP CONNECTIONS TERMINATED
3 ► CDI *INTERFACE DOWN COMMAND ACCEPTED
4 ► CPA0 IS DN/INPROG WITH INTERNET ADDRESS [10.0.0.3]
5 ► CDI *INTERFACE HAS BEEN MARKED DOWN
```

Figure 3-2. *CDI DN Command/Response Example

FS

Displays the current status of the selected CPA.

Syntax

```
*CDI FS cpa_name
```

where *cpa_name* is the name of the CPA whose status you want to display. (You can use the *CDI STATUS command to list all configured CPAs.)

```
1 ► *CDI FS CPA0
2 ►   CPA0 IS UP WITH INTERNET ADDRESS [10.0.0.3]
3 ►   IN: 2744 PKTS, 436172 BYTES.  OUT: 2983 PKTS, 169228 BYTES.
4 ►   ERRORS: 0 FORMAT, 0 HEADER,  0 NO BUFFER, 0 TIMEOUT.
5 ►   0 COLLISIONS.
```

Figure 3-3. *CDI FS Command/Response Example

Table 3-1. *CDI FS Response Detail

Line	Phrase	Explanation
2	CPA0 IS UP	Interface status, indicating that CPA0 is UP.
	INTERNET ADDRESS	Source Internet address assigned to the network interface, in the example display, [10.0.0.3].
3	IN: PKTS, BYTES OUT: PKTS, BYTES	Number of packets (PKTS) and bytes transmitted (IN) or received (OUT) on this interface.
4, 5	ERRORS:	Number of errors of the following types:
	FORMAT	Number of error packets reported by the CPA. This includes the sum of frames with Frame Check Sequence (FCS) errors, frames that were too short, and other format errors (in the example, 0).
	HEADER	Number of frames not acceptable because of an invalid LAN packet header (in the example, 0 frames).
	NO BUFFER	Number of packets discarded by the CPA because buffers were not available, in the example display, 0 packets. This occurs when data packets are not read from the CPA.
	TIMEOUT	Number of timeouts when no data was received from the LAN (in the example, 0). This is not actually an error: There simply may be no data to transfer from the server.
	COLLISIONS	Number of times the CPA started sending data at the same time incoming data was arriving (in the example, 0 times).

HELP

Displays either a list of CDI commands, or help text for a specific CDI command.

Syntax

```
*CDI {HELP | ?} [command]
```

The general form of this command, *CDI HELP (or *CDI ?) displays a list of valid CDI commands, as shown in the figure below.

```
1 ► *CDI HELP
2 ►  COMMANDS: BRKPT DN FS HELP ID PING STATUS TCP TERM UP
```

Figure 3-4. *CDI HELP Command/Response Example

The specific form of this command, *CDI HELP *command*, displays help text for that *command*. For example, if you enter, *CDI HELP BRKPT, the following help text displays:

```
"*CDI BRKPT" BRKPTS THE PRINT FILE
```

ID

Displays current CDI version information. The number enclosed in parentheses is an internal software revision number. Please use this number when reporting CDI problems to your Unisys CSE.

Syntax

*CDI ID

```
1 ► *CDI ID
2 ►   CDI 1R5C (1-5-17)   (12/02/03-17:02:21)
```

Figure 3-5. *CDI ID Command/Response Example

PING

Checks to see if the destination server responds. You can use this command to test any server that is configured in the CDI\$PARAM element. You cannot ping addresses that are not configured in CDI\$PARAM.

Syntax

```
*CDI PING address
```

where *address* is the target host Internet address.

```
1 ► *CDI PING 10.0.0.1
2 ►   IIACT : PING SUC - RESPONSE TOOK .412 SECS, SEQNO 1
```

Figure 3-6. *CDI PING Command/Response Example

STATUS

Displays the start time of the network control program, the status of CDI, and the status and Internet address of each configured CPA.

Syntax

```
*CDI {STATUS | STAT}
```

```
1 ► *CDI STATUS
2 ►   CDI 1R5C (1-5-17)   (12/02/03-17:02:21)  UP SINCE 16:00:00 ON 12/05/03
3 ►   CPA0 IS UP WITH INTERNET ADDRESS [10.0.0.3]
4 ►   CPA1 IS UP WITH INTERNET ADDRESS [10.0.1.3]
```

Figure 3-7. *CDI STATUS Command/Response Example

① TCP ABORT

Aborts the specified TCP connection. *This command is for debugging purposes only.*

Syntax

```
*CDI TCP ABORT con-id
```

where *con-id* represents the name of the connection to be aborted. (To determine a valid *con-id*, use the *CDI TCP CONNECTIONS command.)

```
1 ► *CDI TCP ABORT 4
2 ► 4:143037 CLOSED [10.0.0.3] 7000 -> 0 [10.0.1.3]
3 ► CONNECTION MARKED TO BE ABORTED
```

Figure 3-8. *CDI TCP ABORT Command/Response Example

TCP CONNECTIONS

Lists all of the active connections in CDI. The connection number (the “CON ID” column at the left of the display) is the number you use to issue *CDI TCP KILL or *CDI TCP ABORT commands.

Syntax

```
*CDI TCP {CONNECTIONS | CON}
```

```
1 ► *CDI TCP CON
2 ►   CON ID      STATE      S-ADDRESS  S-PORT    D-PORT    D-ADDRESS
3 ►   1:104821   LISTEN     [0.0.0.0]  7000     -> 0       [0.0.0.0]
4 ►   2:104823   ESTAB      [10.0.0.3] 2125     -> 2111    [10.0.0.1]
```

Figure 3-9. *CDI TCP CON Command/Response Example

The CON ID indicates the connection usage and the time when the session was established. It has one of the following forms:

hhmmss	a connection opened by the server to CSC to return a response.
CChhmm	a session for the *CSC CMD keyin.
Rhmmss	a connection created by CSC to send a request to the server (h is the low digit of the hour).
VRhmm	a session used to retrieve a volume report from the server.

TCP HELP

Displays a list of TCP-related commands.

Syntax

```
*CDI TCP HELP
```

```
1 ► *CDI TCP HELP  
2 ►  COMMANDS: ABORT <con>, CON, HELP, KILL <con>, STATUS <con>
```

Figure 3-10. *CDI TCP HELP Command/Response Example

① TCP KILL

Removes the data structure of the specified connection. *This command is for debugging purposes only.*

Syntax

```
*CDI TCP KILL con-id
```

where *con-id* is the name of the connection to be removed. (To determine a valid *con-id*, use the *CDI TCP CONNECTIONS command.)

```
1 ► *CDI TCP KILL 2
2 ► 1:143547 LISTEN [0.0.0.0] 7000 -> 0 [0.0.0.0]
3 ► 2:143515 ESTAB [10.0.0.3] 7000 -> 2416 [10.0.0.1]
4 ► CONNECTION MARKED TO BE KILLED
```

Figure 3-11. *CDI TCP KILL Command/Response Example

TCP STATUS

Displays information about the specified connection.

Syntax

```
*CDI TCP {STATUS | STAT} con-id
```

where *con-id* is the name of the connection for which you want to display status information. (To determine a valid *con-id*, use the *CDI TCP CONNECTIONS command.)

```
1 ► *CDI TCP STATUS 1
2 ►   CON ID      STATE   S-ADDRESS  S-PORT  D-PORT  D-ADDRESS
3 ►   1:143807  LISTEN   [0.0.0.0]  7000   ⇒   0   [0.0.0.0]
4 ►   RWIN: 2048, SWIN: 7755, SEGWIN 7755, ROOM: 2048, RTD:51/53
5 ►   IVB:0, OVB:0, OPB:0, TOTSND:3199 TOTRX:5563
6 ►   SNDNXT:61690474 SNDUNA:61690474 RTXSEQ:61690474 ACKTMO:15
```

Figure 3-12. *CDI TCP STATUS Command/Response Example

Table 3-2. TCP Connection States

State	Explanation
Closed	There is no connection state.
CloseWt	Waiting for a connection termination request from the local user
Closing	Waiting for a connection termination request acknowledgment from the remote TCP.
Establ	The connection is open, and received data can be delivered to the user.
FinWt1	Waiting for a connection termination request from the remote TCP, or acknowledgment of the connection termination request previously sent.
FinWt2	Waiting for a connection termination request from the remote TCP.
Lastack	Waiting for an acknowledgment of the connection termination request previously sent to the remote TCP, including acknowledgment of the remote TCP's connection termination request.
Listen	Waiting for a connection request from any remote TCP and port.
SynRcv	Waiting for a confirming connection request acknowledgment after receiving and sending a connection request.
SynSent	Waiting for a matching connection request after sending a connection request.
TimeWt	Waiting for enough time to pass to verify that the remote TCP received acknowledgment of its connection termination request.

TERM

Terminates CDI normally.

Syntax

*CDI TERM

```
1 ► *CDI TERM
2 ► TCP CONNECTIONS TERMINATED
3 ► CDI *INTERFACE DOWN COMMAND ACCEPTED
4 ► CDI FIN
```

Figure 3-13. *CDI TERM Command/Response Example

UP

Brings UP the specified CPA.

Syntax

```
*CDI UP cpa_name
```

where *cpa_name* is the name of the CPA you want to bring UP. (You can use the *CDI STATUS command to examine the current statuses of all configured CPAs.)

```
1 ► *CDI UP CPA0
2 ► CDI CPA0 POC TEST COMPLETE
3 ► CDI* CPA ELC (ATTN MODE) CPA0 UP: ADDRESS: 02:E6:D3:07:80:00
4 ► CPA0 IS UP WITH INTERNET ADDRESS [10.0.1.3]
```

Figure 3-14. *CDI UP Command/Response Example

4. CSC COMMAND REFERENCE

This chapter presents reference material on all CSC commands. It is organized alphabetically by command for quick access.

Each command begins on a new page, and includes:

- a description of the command, its uses, and its results.
- a syntax description.
- a representation of a console display, showing the command on the top line and its response (if any) on subsequent lines. On multi-line displays, each line is numbered for quick reference.
- any miscellaneous usage notes.

NOTE

*All CSC console commands begin with “*CSC”. If your site has changed the CSC command name, use your site’s command instead of *CSC.*

A Note About Diagnostic Commands ⓘ

Included in this chapter are a number of diagnostic commands, designated by the ⓘ symbol to the left of the command name. These commands are for debugging purposes only. These commands include:

ABORT
CYCLE
DEBUG OFF
DEBUG ON
DEBUG STATUS
EABT

Locations in a Library

Locations within a library are expressed as a number with 3 to 5 parts. The first two parts are always the ACS number and the LSM number. These are established by the site based on physical and electrical connection between the library components. Following are the types of library locations used in CSC commands (A,L = ACS#, LSM#):

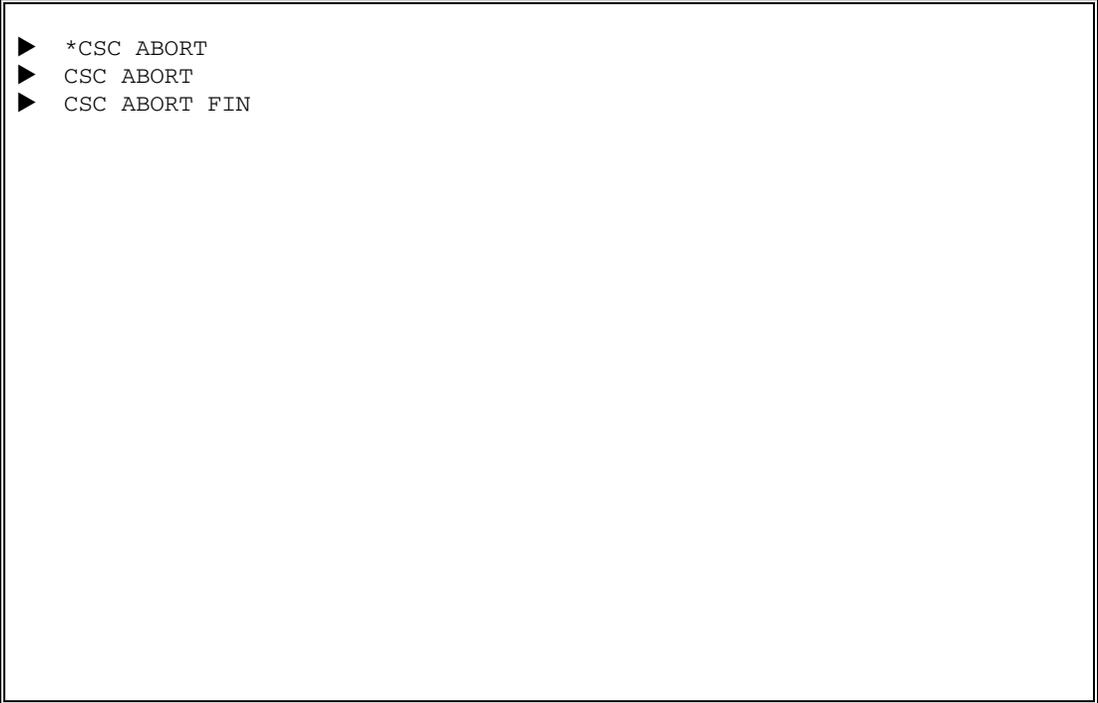
- A,L,# -- CAP # in the designated LSM.
- A,L,#p,#d -- Tape transport on panel #p and position #d of the designated LSM
- A,L,#p,#r,#c -- Storage cell in the designated LSM on panel #p in row #r and column #d

① ABORT

Terminates CSC with an error. This command is equivalent to X'ing off the CSC run. Use this command to terminate CSC only if you suspect an error and CSC will not terminate normally.

Syntax

```
*CSC ABORT
```



```
1 ► *CSC ABORT  
2 ► CSC ABORT  
3 ► CSC ABORT FIN
```

Figure 4-1. *CSC ABORT Command/Response Example

ACTIVATE

Prompts CSC to change the interface or path used to communicate with the server. If an ACTIVATE is requested and there is no other usable path, then the existing path is reactivated.

Syntax

```
*CSC ACTIVATE [INTERFACE|PATH] {interface-name/path-name}
```

where *interface name* or *path name* is the name of the interface or path to be activated.

```
1 ► *CSC ACTIVATE  CDI01
2 ►   CDI01 HAS BEEN ACTIVATED
```

Figure 4-2. *CSC ACTIVATE Command/Response Example

When this command includes an interface name, each path in the interface is considered for activation. If none of these paths can communicate with the server, a failure message is displayed and the request is ignored. If a usable path is available, then it becomes the active path.

When this command includes a path name, then no other path is considered for use. If CSC determines that communication with the server is possible through the specified path, then it becomes the active path. If communication is not possible through the specified path, an error message is displayed and CSC ignores the request.

BRKPT

Closes the current print file and establishes a new one.

Syntax

*CSC BRKPT

```
1 ► *CSC BRKPT
2 ► PRINT$ breakpointed to new cycle of *CSC-PRT-FILE
```

Figure 4-3. *CSC BRKPT Command/Response Example

CLEAR

Sends a clear-lock request to the server. The syntax of the CLEAR command is checked and if invalid, a console message is displayed indicating the error. Otherwise, the request is acknowledged with a console message and the request is sent to the server. A console message indicating the result of the CLEAR command is displayed when the response is received from the server.

NOTE

The form of server locking that is affected by this command is only used by ACSLS. The NCS server will accept this command and return a “success” status without performing any server action.

Syntax

```
*CSC CLEAR LOCK-DRIVE drive_name
```

```
*CSC CLEAR LOCK-VOLUME volume_id
```

```
1 ► *csc clear lock-drive wtap1
2 ► CLEAR Request Has Been Sent to the Server
3 ► CLEAR LCK-DRV WTAP1 1,0,3,1 complete

4 ► *csc clear lock-volume W00021
5 ► CLEAR Request Has Been Sent to the Server
6 ► CLEAR LCK-VOL W00021 complete
```

Figure 4-4. *CSC CLEAR Command/Response Example

CMD (NCS only)

This command is only available with the NCS server. It causes CSC to send an arbitrary HSC or Library Station command to the server and displays the output from that command. The site must properly configure both CSC and the NCS server to allow this command to function.

Syntax

*CSC CMD arbitrary HSC or Library Station command

```
1 ► *CSC CMD .D ACS
2 ► .SLS1000I ACS 00 STATUS: CONNECTED
3 ► COMPATIBILITY LEVELS: HSC=12, LMU=12
4 ► DUAL LMU NOT CONFIGURED
5 ► STATION 0F01 ONLINE
6 ► SCRATCH VOLUMES AVAILABLE..... 121
7 ► FREE CELLS AVAILABLE..... 287
```

Figure 4-5. *CSC CMD Command/Response Example

CSC implements the CMD function using a terminal session to the server in console mode. Each time a *CSC CMD is issued CSC displays whatever console messages occur for the next 60 seconds. This has two side effects:

- Any broadcast messages to the NCS host console during that time will also appear in the *CSC CMD output.
- If the server command takes longer than 60 seconds, you will not see the output.

① CYCLE

Closes the current event log and establishes a new one. *This command is for problem diagnostics only.*

Syntax

```
*CSC CYCLE
```

```
1 ► *CSC CYCLE
2 ► *CSC-LOG-FILE CYCLED - CURRENT CYCLE = 115
```

Figure 4-6. *CSC CYCLE Command/Response Example

① DEBUG OFF

Restores debugging mode to the default level specified in the CSC runstream.
This command is for problem diagnostics only.

Syntax

```
*CSC DEBUG OFF
```

```
1 ► *CSC DEBUG OFF
2 ►   DEBUGGING HAS BEEN DISABLED
```

Figure 4-7. *CSC DEBUG OFF Command/Response Example

① DEBUG ON

Sets full debugging mode. *This command is for problem diagnostics only.*

Syntax

```
*CSC DEBUG ON
```

```
1 ► *CSC DEBUG ON
2 ►   DEBUGGING HAS BEEN ENABLED
```

Figure 4-8. *CSC DEBUG ON Command/Response Example

① DEBUG STATUS

Displays the status of debugging mode. *This command is for problem diagnostics only.*

Syntax

```
*CSC DEBUG STATUS
```

```
1 ► *CSC DEBUG STATUS
2 ►   DEBUGGING IS DISABLED
```

Figure 4-9. *CSC DEBUG STATUS Command/Response Example

DISMOUNT

Sends a request to the server to dismount the volume from a drive. The **FORCE** option instructs the server to force the rewind and unload the volume. *Exercise caution when using the **FORCE** option with this command because CSC cannot first determine if the cartridge tape being dismounted is in use.*

Syntax

```
*CSC DISMOUNT drive_name [FORCE]
```

```
1 ► *CSC DISMOUNT ATAP10
2 ► CSC *R00025 DISMOUNT ATAP10 COMPLETE
```

Figure 4-10. *CSC DISMOUNT Command/Response Example

DOWN (DN)

This command causes CSC to stop using an interface or path.

Syntax

```
*CSC DOWN [INTERFACE|PATH] interface-name/path-name
```

```
1 ► *CSC DOWN PATH PATH2
2 ► DOWN of PATH2 successful
```

Figure 4-11. *CSC DOWN Command/Response Example

The INTERFACE and PATH keywords are not needed if all interface and path names are unique.

If this command includes an interface name, the interface is marked down and that interface is no longer used to communicate with the server. This affects all paths controlled by the interface.

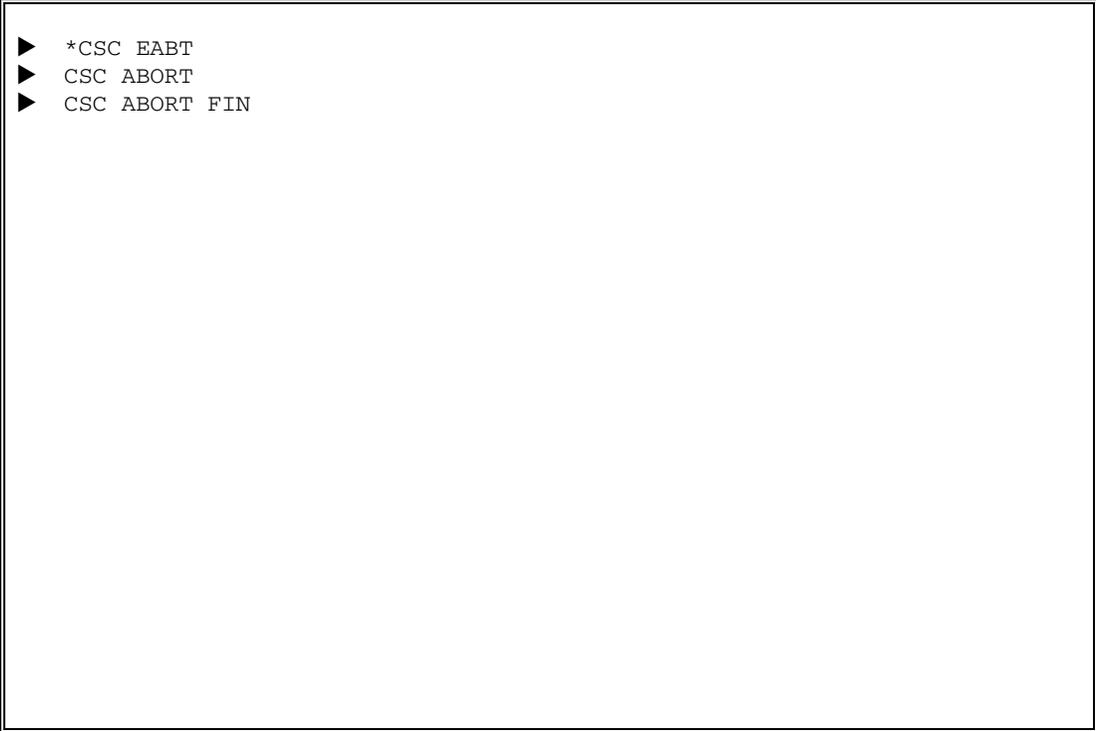
If this command includes a path name, that path to the server is marked down and is no longer used to communicate with the server.

① EABT

Terminates CSC with an error; equivalent to E'ing off the run. Use the EABT command only if the *CSC TERM command fails, or to obtain problem diagnostic information.

Syntax

```
*CSC EABT
```



```
1 ► *CSC EABT  
2 ► CSC ABORT  
3 ► CSC ABORT FIN
```

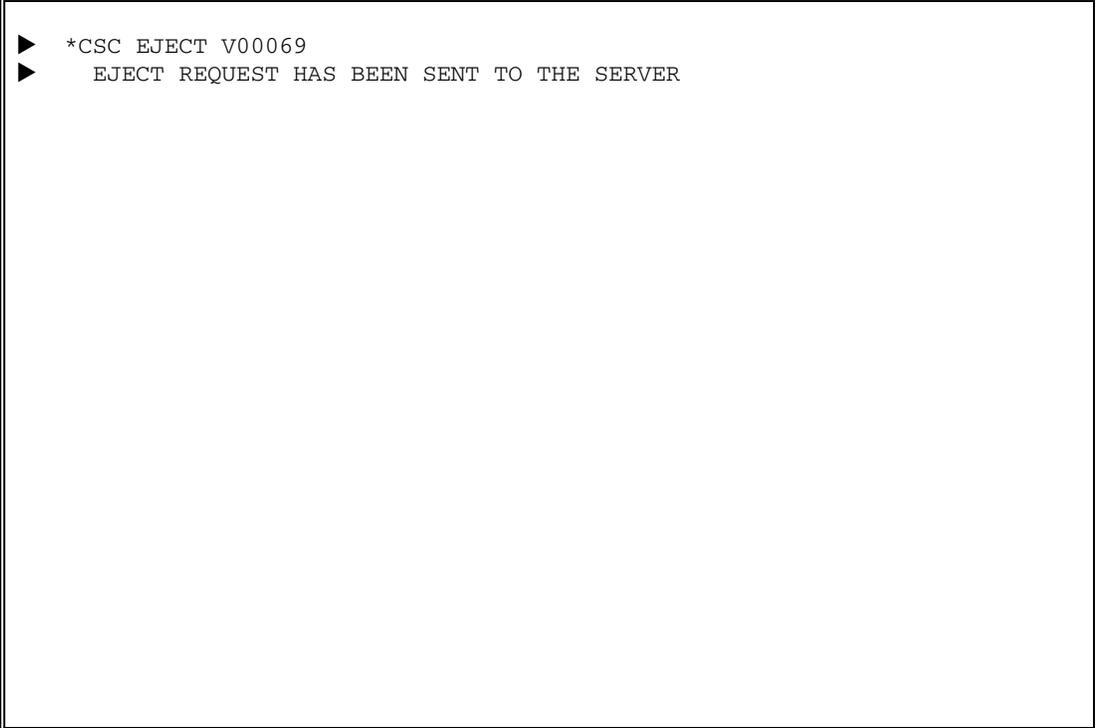
Figure 4-12. *CSC EABT Command/Response Example

EJECT

Ejects a cartridge tape from the ACS. For the eject to work, there must be an available Cartridge Access Port (CAP) with a *non-zero* priority in the ACS containing the volume to be ejected. All eject requests will be acknowledged with a console message stating that the request has been sent to the server. Once a valid request is processed *and* the cartridge tape has been removed from the CAP, CSC will acknowledge the event with a console message that the volume was ejected. If the request fails for whatever reason (e.g., the volume is not in the ACS, all CAPs are busy, there are no CAPs with a non-zero priority), CSC will issue the appropriate console message.

Syntax

```
*CSC EJECT volume_id
```



```
1 ► *CSC EJECT V00069
2 ►   EJECT REQUEST HAS BEEN SENT TO THE SERVER
```

Figure 4-13. *CSC EJECT Command/Response Example

ENTER

Enters a cartridge tape into the ACS. All enter requests will be acknowledged with a console message stating that the request has been sent to the server. Once a valid request is processed and the cartridge tape(s) have been entered into the ACS, CSC will acknowledge the event with a console message for each volume entered.

Syntax

```
*CSC ENTER acs_id, lsm_id, cap_id
```

```
1 ► *CSC ENTER 1,0,0
2 ► ENTER REQUEST HAS BEEN SENT TO THE SERVER
```

Figure 4-14. *CSC ENTER Command/Response Example

HELP

Displays a list of available *CSC commands or the syntax required for a specific *CSC command.

Syntax

*CSC HELP [command]

```
1 ► *CSC HELP
2 ► THE FOLLOWING CSC KEYINS ARE AVAILABLE :
3 ► ABORT ACTIVATE BRKPT CLEAR CMD CYCLE DEBUG DISMOUNT
4 ► DOWN EABT EJECT ENTER HELP LEVEL MEMORY MOUNT QUERY
5 ► QUEUES REPORT STATUS TERM UP VENTER VERBOSE
```

Figure 4-15. *CSC HELP Command/Response Example

LEVEL

Displays the product levels for CSC and CSCUI. The numbers enclosed in parentheses are internal software engineering numbers. Please use these numbers when reporting CSC problems to your Unisys CSE.

Syntax

*CSC LEVEL

```
1 ► *CSC LEVEL
2 ►     CSC 5R1 (5-1-3)
3 ►     UI 2R1E (2-1-16)
4 ►     CDI 1R5C (1-5-17)
```

Figure 4-16. *CSC LEVEL Command/Response Example

MEMORY

Displays the amount of memory currently in use by CSC.

Syntax

```
*CSC MEMORY
```

```
1 ►
2 ► *CSC MEMORY
3 ► MEMORY USAGE AS OF MON DEC 8 15:59:28 2003
4 ► PACKET POOL    MAXIMUM    SIZE    FREE
5 ►     STK         50         38     49
6 ►     CSCUI      50        161    48
7 ►     CONS       50         21    50
```

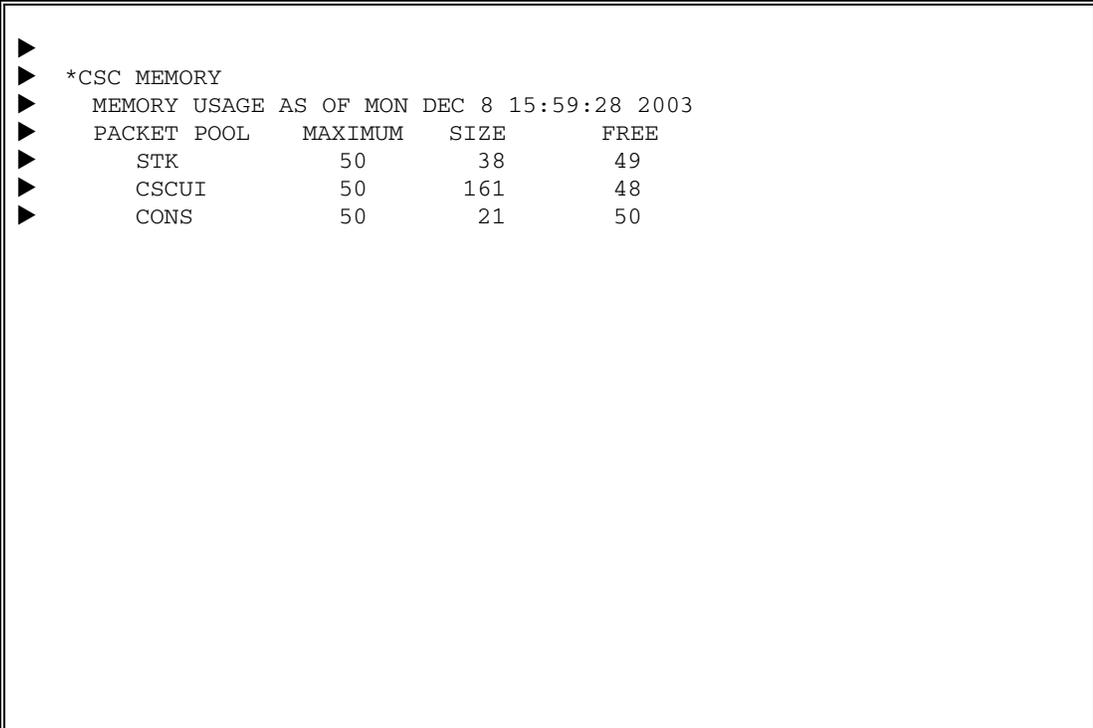


Figure 4-17. *CSC MEMORY Command/Response Example

The resulting display shows information about three types of buffers. The “MAXIMUM” column shows the number of configured memory blocks. The “SIZE” column shows the different memory block sizes used by the buffer types. The “FREE” column shows the number of available memory blocks.

MOUNT

Sends a mount request to the server.

Syntax

```
*CSC MOUNT volume_id drive_name
```

```
1 ► *CSC MOUNT U00053 ATAP10  
2 ► CSC *R00010 MOUNT U00053 ON ATAP10 COMPLETE
```

Figure 4-18. *CSC MOUNT Command/Response Example

QUERY

Sends a query request to the server. The syntax of the QUERY command is checked and if invalid, a console message is displayed indicating the error. Otherwise, the request will be acknowledged with a console message stating that the request has been sent to the server.

Messages containing the result of the QUERY command are displayed when the response is received from the server. If the QUERY command succeeds, a header line is displayed followed by a console message for each entry in the response. If the response contains multiple entries, the first 10 entries are displayed, followed by a read-and-reply console message querying the user if more display is wanted. If a 'Y' is entered, the next 10 entries are displayed. This continues until all entries are displayed, the user enters an 'N' response, or the number of messages exceeds 252, which is the maximum number of messages that CSC will display. If the request fails, a message indicating the request and status returned by the server is displayed.

Syntax

```
*CSC QUERY type id  
*CSC QUERY type ALL
```

where *type id* is:

- ACS *acs*
- CAP *acs, lsm, cap*
- DRIVE *drive_name*
- LOCK-DRIVE *drive-id*
- LOCK-VOLUME *volume_id*
- LSM *acs, lsm*
- POOL *pool_nbr*
- REQUEST *request_nbr*
- SCRATCH *pool_nbr*
- SERVER
- VOLUME *volume_id*

```

1 ► *csc query acs all
2 ►   QUERY Request Has Been Sent to the Server
3 ►   QUERY  ACS   STATE   FREE-CELL-CNT
4 ►   ACS    0    ONLINE   1697
5 ►   ACS    1    ONLINE   503

6 ► *csc query lsm 1,0
7 ►   QUERY Request Has Been Sent to the Server
8 ►   QUERY  LSM   STATE   FREE-CELL-CNT
9 ►   LSM    1,0  ONLINE   503

10 ► *csc query cap all
11 ►   QUERY Request Has Been Sent to the Server
12 ►   QUERY  CAP   PRI   SIZE  STATE   MODE  STATUS
13 ►   CAP    0,0,0  1   21  ONLINE   Man
14 ►   CAP    1,0,0  1   20  ONLINE   Man
15 ►   CAP    1,0,1  0   30  ONLINE   Man  Eject
16 ►   CAP    1,0,2  0   1   ONLINE   Auto

17 ► *csc query server
18 ►   QUERY Request Has Been Sent to the Server
19 ►   QUERY  SERVER STATE   FREE-CELL-CNT
20 ►   SERVER          RUN      2200

```

Figure 4-19. *CSC QUERY ACS/LSM/CAP/SERVER Command/Response Example

```

1 ► *csc query pool all
2 ►   QUERY Request Has Been Sent to the Server
3 ►   QUERY  POOL  VOL-CNT  ATTRIBUTE
4 ►   POOL    0     20
5 ►   POOL    5     486
6 ►   POOL    6     42
7 ►   POOL   1001    5
8 ►   POOL   1002    0      Overflow

9 ► *csc query scratch 1001
10 ►   QUERY Request Has Been Sent to the Server
11 ►   QUERY  POOL  VOLUME  HOME-ADDR  STATUS  TYPE
12 ►   SCRATCH 1001  RA0000  1,0,3,15,0  Home   DD3A
13 ►   SCRATCH 1001  RA0360  1,0,2,1,3   Home   3490E
14 ►   SCRATCH 1001  RB0030  1,0,3,15,1  Home   DD3B
15 ►   SCRATCH 1001  RC0022  1,0,2,8,2   Home   DD3C
16 ►   SCRATCH 1001  W00088  1,0,2,4,2   In-Drive 3480

17 ► *csc query request all
18 ►   QUERY Request Has Been Sent to the Server
19 ►   QUERY  NUMBER  REQUEST  STATUS
20 ►   REQUEST 360    QUERY    PENDING
21 ►   REQUEST 359    DISMOUNT ACTIVE
22 ►   REQUEST 358    MOUNT    ACTIVE

```

Figure 4-20. *CSC QUERY POOL/SCRATCH/REQUEST Command/Response Example

```

1 ► *csc query drive all
2 ►   QUERY Request Has Been Sent to the Server
3 ►   QUERY  NAME  ADDRESS  STATE  STATUS  VOLUME  TYPE
4 ►   DRIVE   0,0,9,2  ONLINE  Available  4480
5 ►   DRIVE   0,0,9,3  ONLINE  Available  4480
6 ►   DRIVE   0,0,10,0  ONLINE  Available  4480
7 ►   DRIVE   0,0,10,1  ONLINE  Available  4480
8 ►   DRIVE   0,0,10,2  ONLINE  Available  4480
9 ►   DRIVE   0,0,10,3  ONLINE  Available  4480
10 ►  DRIVE  MTAP0  1,0,1,0  ONLINE  Available  4490
11 ►  DRIVE   1,0,1,1  OFFLINE  Available  unknown
12 ►  DRIVE  MTAP2  1,0,1,2  ONLINE  Available  44900
13 ►  DRIVE   1,0,1,3  OFFLINE  Available  unknown
14 ► 0-QUERY DRIVE ALL -- 10 of 14 lines, continue?
15 ► 0 Y
16 ►   QUERY  NAME  ADDRESS  STATE  STATUS  VOLUME  TYPE
17 ►   DRIVE  WTAP0  1,0,3,0  ONLINE  W00020  4480
18 ►   DRIVE  WTAP1  1,0,3,1  ONLINE  Available  4480
19 ►   DRIVE  WTAP2  1,0,3,2  ONLINE  Available  4480
20 ►   DRIVE  WTAP3  1,0,3,3  ONLINE  Available  4480

21 ► *csc query volume W00020
22 ►   QUERY Request Has Been Sent to the Server
23 ►   QUERY  VOLUME  STATUS  NAME  LOCATION  TYPE
24 ►   VOLUME  W00020  Home  1,0,2,8,4  3480

25 ► *csc query volume W00021
26 ►   QUERY Request Has Been Sent to the Server
27 ►   QUERY  VOLUME  STATUS  NAME  LOCATION  TYPE
28 ►   VOLUME  W00021  In-Drive  WTAP1  1,0,3,1  3480

```

Figure 4-21. *CSC QUERY DRIVE/VOLUME Command/Response Example

```

1 ► *csc query lock-drive all
2 ►   QUERY Request Has Been Sent to the Server
3 ►   QUERY  NAME  ADDR  LOCK-ID  STATUS  USER ID
4 ►   LCK-DRV  WTAP0  1,0,3,0  25634  In_Use  FC14 Unisys 2200/3800
5 ►   LCK-DRV  WTAP1  1,0,3,1  25634  In_Use  FC14 Unisys 2200/3800

6 ► *csc query lock-volume all
7 ►   QUERY Request Has Been Sent to the Server
8 ►   QUERY  VOLUME  LOCK-ID  STATUS  USER ID
9 ►   LCK-VOL  W00021  24280  In_Use  acsss

```

Figure 4-22. *CSC QUERY LOCK-DRIVE/LOCK-VOLUME Command/Response Example

If the syntax of the QUERY command is invalid, the following error message is displayed:

```
Request Denied - status
```

where *status* indicates the invalid syntax.

If the QUERY command is sent to the server and the server returns an error status, an error message is displayed containing the server status. The message has the following format:

```
QUERY type failed - status
```

For example:

```
QUERY POOL 999 failed - POOL_NOT_FOUND
```

The syntax of the keyin and the format of the console message display is similar to the input and output documented in the "Command Reference" of the *ACSL Administrator's Guide*.

Main differences include:

- For all messages containing drive references, both the OS 2200 drive-name and the server drive address are displayed.
- The number of current and pending requests is not displayed for ACS or LSM requests.
- Low and high water marks are not displayed for POOL requests.
- The lock duration and number of pending locks is not displayed for LOCK-DRIVE and LOCK-VOLUME requests.

QUEUES

Displays a list of all active cartridge drives and the number of requests that CSC has queued to each drive. These requests are either in progress or waiting for an in-progress request to complete.

Syntax

```
*CSC QUEUES
```

1	▶	*CSC QUEUES					
2	▶	DRIVE	ACS	LSM	VOLUME	ENTRIES	CURRENT REQUEST
3	▶	-----	---	---	-----	-----	-----
4	▶	ATAP5	0	0	U00032	0	DISMOUNT
5	▶	ATAP9	0	0		2	MOUNT U00044
6	▶	ATAP10	0	0		0	MOUNT U00033

Figure 4-23. *CSC QUEUES Command/Response Example

The *CSC QUEUES command displays information about active drives. The “DRIVE”, “ACS”, and “LSM” columns specify the cartridge drive information. The “VOLUME” column specifies the volser of the cartridge tape on the cartridge drive. The “ENTRIES” column indicates the number of requests for that cartridge drive. The “CURRENT REQUEST” column specifies the request in progress.

REPORT

Displays information about entities configured within CSC.

Syntax

```
*CSC REPORT type [print queue]
```

Where:

type may be one of the following:

ACS	reports configured ACS names and corresponding ACS numbers (see Fig. 4-24 below).
DRIVES	reports configured Drives with server address and last known status (see Fig. 4-25 below).
POOLS	reports configured default and CTL-pools with current volume counts (see Fig. 4-26 below).
FILES	reports the qualifier, filename, and cycle of files in use by CSC (see Fig. 4-27 below).

print queue If a *print queue* is not specified, the result is displayed on the console. If a *print queue* is specified, the result is queued to the configured device, siteid, or userid.

```
1 ► *CSC REPORT ACS
2 ►
3 ►   ACS NAME  NUMBER
4 ►   - - - - -  - - - - -
5 ►   ACS00      0
6 ►   VTAPE     126
```

Figure 4-24. *CSC REPORT ACS Command/Response Example

```

1 ► *CSC REPORT DRIVES
2 ►
3 ► DRIVE      TYPE      ACS  LSM  PNL  DRV   STATUS  VOLUME
4 ► -----
5 ► ATAP8      4480      0   0   2   0   AVAIL
6 ► ATAP9      4480      0   0   2   1   AVAIL
7 ► ATAP10     4480      0   0   2   2   AVAIL
8 ► ATAP11     4480      0   0   2   3   AVAIL
9 ► ATAP12     4480      0   0   3   0   AVAIL
10 ► ATAP13     4480      0   0   3   1   AVAIL
11 ► ATAP14     4480      0   0   3   2   AVAIL
12 ► ATAP15     4480      0   0   3   3   LOADED  L12773
13 ► VTAP8              126   0   3   0   LOADED  V00045
14 ► VTAP9              126   0   3   1   LOADED  V00047
15 ► VTAP10             126   0   3   2   AVAIL
16 ► VTAP11             126   0   3   3   AVAIL
17 ► VTAP12             126   0   4   0   AVAIL
18 ► VTAP13             126   0   4   1   AVAIL
19 ► VTAP14             126   0   4   2   AVAIL
20 ► VTAP15             126   0   4   3   AVAIL

```

Figure 4-25. *CSC REPORT DRIVES Command/Response Example

```

1 ► *CSC REPORT POOLS
2 ►
3 ► CTL-POOL    MEDIA      TYPE  POOL#  COUNT  DEFAULT FOR
4 ► -----
5 ► VIRTUAL                    SL     7     977  VTAPE  ALL
6 ► VIRTUAL                    NL     8     992  VTAPE  ALL
7 ► Default                    SL     5     491
8 ► Default                    NL     6      89

```

Figure 4-26. *CSC REPORT POOLS Command/Response Example

```

1 ► *CSC REPORT FILES
2 ►
3 ► CSC =      SYS$LIB$*CSC(67)
4 ► Diag$ =   CSC-5R1*CSC-DIAG$(202)
5 ► Print =   CSC-5R1*CSC-PRT-FILE(108)
6 ► Log =     CSC-5R1*CSC-LOG-FILE(354)

```

Figure 4-27. *CSC REPORT FILES Command/Response Example

STATUS

Displays status information about CSC. A command option is available that also describes the status of each possible communication path to the server. Information displayed by the STATUS command includes the:

- CSC level
- time that CSC initialized
- status – INITIALIZING, ACTIVE, or TERMINATING
- total number of active tasks
- total number of configured tasks
- lock id
- total number of mounts completed since initialization
- path used for server communications
- operational status of each (if the PATH option is specified)

Syntax

```
*CSC STATUS [ [PATH|INTERFACE] [path-name/interface-name] ]
```

```
1 ► *CSC STATUS
2 ►   CSC 5R1 (5-1-3)
3 ►   CSC STATUS AS OF MON DEC  8 15:04:13 2003
4 ►     ACTIVE SINCE THU DEC  4 09:12:59 2003
5 ►     CSC IS ACTIVE
6 ►     TASKS TOTAL/ACTIVE/QUEUED: 14/10/0
7 ►     REQUESTS/ACTIVE/QUEUED: 4/1
8 ►       Mnt=2/1  UI=1/0  Ent-Ej=0/0  Qry=0/0  Supp=1/0
9 ►     USING ACSLS SERVER VIA ER STK$
10 ►    USING NCS SERVER VIA CARTTAPELIB$ WITH CTL ID CSC
11 ►    TOTAL MOUNTS COMPLETED = 178
12 ►    CSC CMD IS AVAILABLE
13 ►    CONTROL PATH IS CMSP VIA ICMS
14 ►    Interface ICDI (CDI) is Up using port 7003
15 ►    Interface ICMS (CMSA) is Up using port 7000
16 ►    Interface ICMST (CMSTEST) is Unavailable 10000
17 ►    PATH CDIP (VIA ICDI:UP) IS AVAILABLE AS OF 15:04:12
18 ►    PATH CMSP (VIA ICMS:UP) IS AVAILABLE AS OF 15:04:13
19 ►    Path CMSTP (via ICMST:Unavail) is Untested
```

Figure 4-28. *CSC STATUS Command/Response Example

This example includes output from all three variations of the STATUS command. Lines 14 through 16 are only present with the INTERFACE keyword. Lines 17 through 19 are only present with the PATH keyword.

Table 4-1. *CSC STATUS Response Notes

Line	Explanation
2	This line tells the release level of CSC and the internal level of the CSCTM program.
3	This tells when this status output was produced.
4	This tells when CSCTM began execution.
5	The last word tells the status of CSC. Following are the possible values: INITIALIZING – CSC is initializing its internal state. INITIALIZING drive-name – CSC is communicating with OS 2200 and the server to establish the state of the indicated drive. ACTIVE – CSC is active. It is accepting requests for automated operations. TERMINATING – CSC is in the process of terminating. It is not accepting new requests for automated operations.
6	This tells the total number of tasks that CSCTM has created, how many of these tasks are actively doing work for CSC, and how many tasks have been requested but not yet activated.
7	This tells the total number of server requests that CSC is processing. The first number (4) are requests that are being sent to the server or awaiting server responses. The second number (1) are requests queued within CSC to serialize requests for a tape drive, CAP, or other server resource.
8	Only present with VERBOSE STATUS ON. This gives active and queued request counts by type. Types are as follows: Mnt - Mount or dismount requests UI - Scratch and volume reports from the CSC User Interface Ent-Ej - Enter or eject requests Qry - Query requests from the console or volume information requests from CSCUI Supp - Supporting requests for mount and dismount. This includes status requests and requests to manage server resource locks.
9,10	Only one of these lines occurs. It identifies the type of library server (ACSL\$ or NCS) and the type interface to OS 2200 (ER STK\$ or CARTTAPELIB\$). For the CARTTAPELIB\$ interface, the CTL identifier is also shown.
11	This tells the number of tape mounts that have been completed since CSC began execution.

Line Explanation

12 This line is only present with the NCS server. It tells the status of the session CSC uses to process the *CSC CMD keyin. The text is either "CSC CMD is available" or a description of why the command session is not available.

You should notify your site administrator if the text on this line is "CSC CMD is disabled: config error."

13 This line tells which path and interface is being used to send requests to the server. If communication with the server cannot be established, then this line displays: "No functional path to the server".

14-16 These lines give the status of the interfaces which are configured. This information is displayed only when the INTERFACE option is specified. If an interface name is specified, only the specified interface is displayed.

The interface status can be either "Up," "Down," or "Unavailable." If the interface is Up, additional text indicates the port number used for server responses via that interface. An interface that has never been used will be in the Unavailable state. If an attempt to initialize an interface fails, the interface is also placed in the Unavailable state and one of the following codes is appended to indicate the failure.

Code	Meaning
010	CDI is not up and initialized
010000	CMS/CPCOMM is not up and initialized
010003	TSAM TSU already in use by another run
010005	No usable process for TSAM TSU
010006	Incorrect password on the INTERFACE statement
010007	Inocrrrect TSU name on the INTERFACE statement

In Figure 4-28, interface ICMST is unavailable because CMSTEST is not available.

17-19 These lines give the status of the paths which are configured. This information is displayed only when the PATH option is specified. If a pathname is specified, only the specified path is displayed.

The interface name that the path uses and status of that interface is included. The interface status can be: Up, Down, or Unavail.

The time is included to show the last time that CSC verified the operational status. This could be up to 1 minute before the *CSC STATUS PATH keyin. The line also indicates the number of active requests on the path.

Following are the possible operational states:

DOWN – The path is down. CSC will not try to use this path.

UNTESTED – No communication with the server has been attempted using this path.

UNUSABLE – Communication with the server was tried and failed through this path.

AVAILABLE – This path is up and communication with the server has been verified. The path is available for use.

BEING TESTED – Communication with the server is being attempted using this path.

TERM

Terminates CSC. This command allows CSC to perform routine “housekeeping” tasks and terminate normally.

Syntax

```
*CSC TERM
```

```
1 ► *CSC3 TERM
2 ► *Operator requested termination using *CSC3 TERM from T1234A
3 ► *CSC3 IS TERMINATING
4 ► CSC3 *R00027 DISMOUNT ATAP8 FAILED WITH STATUS IDLE_PENDING
5 ► CSC3 FIN
```

Figure 4-29. *CSC TERM Command/Response Example

This example shows the termination of CSC3 using the keyin *CSC3.

Line 2 indicates why CSC is terminating (e.g., “CSC3 TERM”). The last phrase of line 2, “from T1234A,” appears if the TERM keyin was issued from a terminal in console mode rather than from the system console.

If this CSC is using the CARTTAPELIB\$ interface, **line 3** indicates the CTL identifier used by this CSC. When ER STK\$ is used, this line has the runid.

UP

This command makes a path or interface available for use.

Syntax

```
*CSC UP [INTERFACE|PATH] interface-name/path-name
```

```
1 ► *CSC UP PATH CPA2  
2 ► UP of CPA2 successful
```

Figure 4-30. *CSC UP Command/Response Example

The INTERFACE and PATH keywords are not needed if all interface and path names are unique.

If this command contains an interface name, the interface is tested to make sure that it is operational and the status of the interface is set and displayed accordingly.

If the command contains a path name, the associated interface must be up. The path is then tested to make sure it is operational and the status of the path is set and displayed accordingly.

VENTER

Enters a cartridge tape with a missing or unreadable label into the ACS. The volume-id in the keyin is then associated with the cartridge tape that was entered. All ENTER requests will be acknowledged with a console message stating that the request has been sent to the server. Once a valid request is processed and the cartridge tape has been entered into the ACS, CSC will acknowledge the event with a console message indicating the volume was entered.

Syntax

```
*CSC VENTER acs_id,lsm_id,cap_id volume_id
```

```
1 ► *CSC VENTER 1,0,0 TESTTP
2 ►   VENTER REQUEST HAS BEEN SENT TO THE SERVER
```

Figure 4-31. *CSC VENTER Command/Response Example

VERBOSE

Controls the display of additional information about CSC processing. Verbose information is available in the following areas:

- CARTTAPELIB\$ Exec interface (CTL)
- Volume eject processing (EJECT)
- Current request status (STATUS)

Syntax

```
*CSC VERBOSE
```

```
*CSC VERBOSE [CTL | EJECT | STATUS] [OFF | ON | PRINT]
```

The original `*CSC VERBOSE n` syntax is replaced by `*CSC VERBOSE CTL state`.

The responses to the various `*CSC VERBOSE` keyin forms are shown in the following figure. The additional verbose output is described in subsequent sections.

```
1 ▶ *CSC VERBOSE
2 ▶ VERBOSE CTL=OFF, EJECT=OFF, STATUS=OFF

3 ▶ *CSC VERBOSE EJECT OFF
4 ▶ EJECT VERBOSE is OFF
5 ▶ VERBOSE CTL=OFF, EJECT=OFF, STATUS=OFF

6 ▶ *CSC VERBOSE EJECT ON
7 ▶ EJECT VERBOSE changed from OFF to ON
8 ▶ VERBOSE CTL=OFF, EJECT=ON, STATUS=OFF
```

Figure 4-32. *CSC VERBOSE Keyin Forms Command/Response Examples

The command on line 1 displays the current verbose settings shown on line 2. Each of the other verbose commands ends with a display of the verbose settings as shown in lines 5 and 8.

The command on line 3 turns off a verbose setting, in this case EJECT. Line 4 is the response to a verbose change when the setting already has the specified value.

The command in line 6 changes a verbose setting. The response in line 7 tells the original and new verbose settings.

These examples use the EJECT setting. Using CTL or STATUS produces similar responses.

VERBOSE CTL

The CTL settings of verbose control the display of information pertaining to tape operations processed through the CARTTAPELIB\$ Exec interface.

Syntax

```
*CSC VERBOSE CTL [OFF | ON | PRINT]
```

```
*CSC VERBOSE 0 is equivalent to *CSC VERBOSE CTL OFF.
```

```
*CSC VERBOSE 1 is equivalent to *CSC VERBOSE CTL ON.
```

```
*CSC VERBOSE 2 is equivalent to *CSC VERBOSE CTL PRINT.
```

The following figure gives examples of the verbose output for CARTTAPELIB\$.

```
1 ▶ CSC*RUN00 TAPE 15:00:21 QSCR SL ACS00 global default pool #5
2 ▶ CSC*RESP ATAP15,ATAP8,ATAP9,ATAP10,ATAP11,ATAP12,ATAP13,ATAP14
3 ▶ CSC*RUN00 TAPE 15:00:25 MOUNT SL ATAP15 global default pool #5

4 ▶ CSC*RUN00 MYVTAPE 15:07:27 QSCR SL ACS00 user specified VIRTUAL
5 ▶ CSC*RESP VTAP8,VTAP9,VTAP10,VTAP11,VTAP12,VTAP13,VTAP14,VTAP15
6 ▶ CSC*RUN00 MYVTAPE 15:07:34 MOUNT SL VTAP8 user specified VIRTUAL

7 ▶ CSC*RUN00 MYVTAPE2 15:12:18 QSCR SL VTAPE* acs default VIRTUAL
8 ▶ CSC*RESP VTAP9,VTAP10,VTAP11,VTAP12,VTAP13,VTAP14,VTAP15,VTAP8
9 ▶ CSC*RUN00 MYVTAPE2 15:12:25 MOUNT SL VTAP9 acs default VIRTUAL
```

Figure 4-33. *CSC VERBOSE Command/Response Example

*CSC VERBOSE Response Notes

The line numbers in the following descriptions refer to the example above.

Assume CSC is configured as shown by the REPORT examples above, and RUN00 has issued the following assignment:

```
@ASG, T TAPE. , U47L
```

- 1 CSC is polled by the Exec and issues a query scratch for a labeled tape using a default ACS and the global scratch pool.
- 2 Is the drive list returned by CSC to the Exec.
- 3 CSC receives a mount request from the Exec and forwards it to the server using the global default scratch pool.

Assume RUN00 issued the following assignment:

```
@ASG, T MYVTAPE, HICM, , , , VIRTUAL
```

- 4 CSC is polled by the Exec and issues a query scratch for a labeled tape using a default ACS and a user specified scratch pool.
- 5 Is the drive list returned by CSC to the Exec.
- 6 CSC receives a mount request from the Exec forwards it to the server using the user specified scratch pool.

Assume RUN00 issued the following assignment:

```
@ASG, T MYVTAPE2, VTAPE
```

- 7 CSC is polled by the Exec and issues a query scratch for a labeled tape using a user specific ACS and a scratch pool configured as a default for that ACS.
- 8 Is the drive list returned by CSC to the Exec.
- 9 CSC receives a mount request from the Exec and forwards it to the server using the ACS default scratch pool.

VERBOSE EJECT

The EJECT settings of verbose control the display of additional information on tape ejects processing.

Syntax

```
*CSC VERBOSE EJECT [OFF | ON ]
```

```

1 ▶ *CSC EJECT AV0001
2 ▶ CSC*EJECT 2247 received from console
3 ▶ CSC*EJECT 2253 received from CSCUI
4 ▶ CSC*EJECT 2253 queued waiting for resources
5 ▶ CSC*R00031 VOLUME AV0001 EJECTED
6 ▶ CSC*EJECT 2247 volumes=AV0001
7 ▶ CSC*EJECT 2247 completed with 1 volumes ejected
8 ▶ CSC*EJECT 2253 volumes=AV0005 AV0006 AV0007 AV0008 AV0009 AV0010
9 ▶ CSC*EJECT 2253 volumes=AV0011 AV0012
10 ▶ CSC*EJECT 2253 completed with 8 volumes ejected

```

Figure 4-34. *CSC VERBOSE EJECT Output Example

*CSC VERBOSE EJECT Output Notes

The line numbers in the following description refer to the example in Figure 4-34.

Each eject request in CSC has a unique identifier. Each output produced by the verbose eject setting contains a 4-digit version of this unique identifier. All output for an eject request contains the same identifier.

- 1 An eject keyin is issued from the console.
- 2 CSC acknowledges the receipt of the eject request from the console.
- 3 CSC acknowledges the receipt of an eject request via the CSC User Interface
- 4 Eject 2253 from CSCUI cannot be processed at this time and has been queued. The reason in this case is that no CAP is available for this eject.
- 5 This is the existing message telling that a volume was ejected as a result of an eject console keyin.
- 6 This is verbose eject output telling the volumes ejected by request 2247.
- 7, 10 Each of these is the last line of verbose eject output. It tells how many volumes were ejected. An eject that fails may show 0 volumes ejected.
- 8,9 These are the verbose output from a multiple volume eject.

VERBOSE STATUS

The STATUS settings of verbose control the display of additional information in response to a *CSC STATUS keyin. The output produced with this setting enabled is shown in the *CSC STATUS command.

Syntax

```
*CSC VERBOSE STATUS [OFF | ON ]
```

5. MESSAGE REFERENCE

This chapter presents reference material for the most common messages you might encounter while using CSC. The discussions explain the messages, and suggest possible responses. In this chapter, messages are grouped by message type (answerable, informational, and subsystem) and then alphabetically or numerically within that message type.

ANSWERABLE MESSAGES

▷ 0-WRITE RING REQUIRED IN *volser* ON UNIT *transport - A,E*

This OS 2200 message indicates a write ring problem. Please refer to the procedures listed under the heading “Write Ring Problems” (page 6-14) in Chapter 6, “Problem Solving.”

INFORMATIONAL MESSAGES

▷ Active path *path-name* has failed

The path being used for communications to the server has failed.

▷ Answer *outstanding messages*

CSC is terminating and it has outstanding messages awaiting an operator reply. Respond to those CSC messages.

▷ CLEAR LCK-DRV failed - *status*
▷ CLEAR LCK-VOL failed - *status*
▷ QUERY *type* failed - *status*

The CLEAR or QUERY request failed. Status indicates the reason for the failure.

▷ CLEAR Request Has Been Rejected by Server
▷ QUERY Request Has Been Rejected by Server

The CLEAR or QUERY request was not accepted by the server.

▷ *xxx* Command is not available

The *CSC *xxx* keyin that you issued has been disabled by your site's configuration.

▷ Command request *request_nbr* cannot be cancelled

At termination, CSC tries to cancel all outstanding requests. The indicated request could not be cancelled on the server.

▷ CONS_input (*keyin*) DETECTED ANOTHER ACTIVE CSC

During initialization the CSC function CONS_input received an error status trying to register the CSC keyin. The “(*keyin*)” portion of the message is only present if the site configuration has changed the CSC keyin to something other than *CSC. After producing this message, CSC terminates.

▷ Contingency detected: *reason*
▷ CSC will restart in 2 minutes

CSC has detected a problem that prevents further execution. *Reason* indicates the nature of the problem. CSC restarts itself with a 2-minute delay and then terminates. This restart is of the CSC runstream in the CSC product file.

▷ Control path is now using path *path-name*

This message indicates that the named path is being used to communicate to the server.

▷ CSC 5R1 (5-1-3)
Copyright 2004 Storage Technology Corporation
All Rights Reserved

These messages are displayed at CSC initialization.

▷ CSC 5R1 initialization complete

Indicates that CSC 5R1 has successfully initialized and is beginning automated operation.

▷ CSC and server settings do not match for *CSC CMD
▷ CSC and server settings do not match for volrpt

The signon parameters in CSC for creating the CMD or volrpt session do not match those on the server.

▷ CSC cannot respond to the following message
- text of the Exec tape message

CSC using the CARTTAPELIB\$ interface received an answerable message request from Exec but CSC does not have the information needed for a response. CSC displays the message and, when the operator responds, returns the response to Exec.

The most likely cause of this message sequence occurs when an unlabeled volume is mounted by hand in manual mode. The server and CSC do not know that a volume has been and, therefore, do not have a reel number for a response.

▷ CSC CMD is disabled: config error

The site-defined configuration for *CSC CMD is not correct. CSC displays this message and disables *CSC CMD.

▷ *CSC CMD is not available now

The session used by CSC to pass console commands to the NCS server is not functioning at this time. CSC will try to reestablish the session within two minutes. You can use the *CSC STATUS keyin to find why the session is not available.

▷ CSC configured drive *drive* is not usable - *status*

While initializing a drive, CSC encountered a condition that prevents automated tape operations on that drive. Status indicates the reason for the failure. It is one of the following:

Status	Meaning
CTL Controlled	The drive is already registered to another CSC using CARTTAPELIB\$.
UNKNOWN DRIVE	The drive name is CSC\$DRIVES is not known to Exec.
ACS_NOT_IN_LIBRARY DRIVE_NOT_IN_LIBRARY LSM_NOT_IN_LIBRARY	The drive location in CSC\$DRIVES is not valid.

▷ CSC Drive count: ACS nn (UP), ACS nn (DN), non-ACS nn

Displays the number of ACS drives, from the CSC drive configuration, that are UP, DOWN, or not in the ACS at CSC initialization.

▷ CSC is waiting for event

If a condition exists that prevents CSC from communicating with the server, the preceding message is produced. "Event" is replaced with a description of the cause of the problem.

▷ CSC to server communications restored

A condition which prevented CSC from communicating with the server has been resolved, thus communications have been restored.

▷ *CSC xxx is disabled

You issued *CSC HELP xxx for a keyin that is disabled by your site's configuration.

- ▷ CTL_input DETECTED ANOTHER ACTIVE CSC
- ▷ CSCIM_input DETECTED ANOTHER ACTIVE CSC
- ▷ CSCUI_input DETECTED ANOTHER ACTIVE CSC

During initialization, the function CSCIM_input, CTL_input, or CSCUI-input received an error status trying to register with either CSCUI or OS 2200 to provide communication with the server. After producing this message, CSC terminates.

- ▷ DEFAULT LABELED POOL USED INSTEAD OF *ctl-pool* ON drive
- ▷ DEFAULT UNLBLD POOL USED INSTEAD OF *ctl-pool* ON drive

The default scratch pool is used instead of the *ctl-pool*, because there is no corresponding ACS pool in the CSC configuration, and the UNDEFINED_POOL MOUNT ACTION IS DEFAULT statement is present in the configuration.

▷ DISMOUNT *drive* ALREADY REQUESTED

This is the response to a console-initiated dismount that is identical to an existing dismount for the indicated drive. This console-initiated dismount will not be performed.

▷ DISMOUNT *drive* COMPLETE

The ACS dismount of a volume has completed successfully.

▷ DISMOUNT *drive* FAILED WITH STATUS *status*

An ACS dismount of the volume from the indicated drive has just failed. Status indicates the reason for the failure as reported by the server.

▷ DISMOUNT *drive* TIMED OUT

A dismount request for the indicated drive did not complete in the allotted time of two minutes.

▷ DISMOUNT *drive* WILL NOT BE DONE

A condition has been detected that prevents the dismount of a volume from the indicated drive. This is the second of two messages. The preceding message explains the condition that caused this message.

▷ DOWN OF *name* SUCCESSFUL

▷ DOWN OF *name* FAILED

A DOWN request was made for an interface or a path. This message indicates the result of that request.

▷ DRIVE *drive* CAN NOT BE REGISTERED WITH THE EXEC

The Exec CARTTAPELIB\$ interface rejected registration of the indicated drive; CSC initialization continues with the drive marked as unusable.

▷ DRIVE *drive* IS NOT AN ACS UNIT

An operation was requested on a cartridge drive that is in the CSC\$DRIVE configuration element, but is not in the ACS. Another message will follow this one to report the actual request failure.

▷ drive-types element does not exist

The element named “drive-types” does not exist in the CSC product file. Without this element, CSC cannot verify that its names and types for library drives are equivalent to those on the server. CSC continues to operate using the names and types from the server.

```
EJECT ##### received from source  
EJECT from source is a duplicate of #####  
EJECT ##### queued waiting for resources  
EJECT ##### volumes=vvvvv1 vvvvv2 vvvvv3 vvvvv4 vvvvv5 vvvvv6  
EJECT ##### completed with ## volumes ejected  
EJECT ##### failed with status stat
```

These messages are produced when VERBOSE EJECT is set to ON. They tell the progress of an eject request and one or more of these will appear in the order shown for each eject request. Following are the fields substitutions in these messages.

is a 4 digit number that identifies a particular eject request.

source is the source of the eject request. It is either "CSCUI" or "console."

vvvvv1 - vvvvv6 are volumes that were ejected.

When an eject is received, either a "received from" or a "duplicate" message is produced.

If the eject request cannot be immediately processed, a "queued" message is produced.

After volumes are ejected, one or more "volumes" messages are produced to tell which volumes were successfully ejected.

If the eject request itself fails, a "failed" message is produced with *stat* telling the reason for the failure.

If the eject does not fail, a "completed" message is produced telling how many (##) volumes were ejected. It is possible for an eject to complete normally and eject 0 volumes.

▷ EJECT FAILED WITH STATUS *status*

An eject failure resulted from the use of the *CSC EJECT command. Status indicates the reason for the failure as reported by the server.

ENTER FAILED WITH STATUS *status*

An enter failure resulted from the use of the *CSC ENTER command. Status indicates the reason for the failure as reported by the server.

- ▷ Equipment check[error] - Unit recovering
- ▷ CPAA(*cpa-name*) Equipment check[error].
- ▷ CPA Reset due to Ethernet Controller Error
- ▷ CPA Reset Successful

These messages indicate that an equipment check or error has been detected by CDI. CDI resets the CPA to attempt to resolve this problem.

- ▷ Fatal errors detected in configuration
- ▷ Check print file

These messages are displayed at initialization if fatal errors have been encountered in the CSC\$PARAM file. The error messages are printed in the PRINT\$ file. CSC will FIN after these messages are displayed.

- > Interface *I/F-name* comm program is not active
- > Interface *I/F-name* has bad process name
- > Interface *I/F-name* has bad password
- > Interface *I/F-name* process is in use
- > Interface *I/F-name* attach failure status: ##

One of these messages displays if CSC cannot complete its attachment to CMS or CPCComm. The portion of the message following *I/F-name* tells the cause of the failure. When the failure cause is "comm program is not active", CSC retries the attachment every 15 seconds. Following all other messages, CSC returns the interface to the uninitialized state.

Failure causes other than "comm program is not active" tell of conditions that may require a change to your site's CSC configuration. The failure-cause describes the TSAM attach status. The following table shows the relationship between attach status and displayed failure cause.

TSAM Status	Displayed failure-cause
10000	comm program is not active
10002 or 10003	Process is in use
10005	has bad process name
10006	has bad password
all others	Attach failure status: ## ## is the TSAM status

- ▷ LABELED CTL-POOL *ctl-pool (ACS-pool)* IS LOW
- ▷ UNLABELED CTL-POOL *ctl-pool (ACS-pool)* IS LOW
- ▷ LABELED CTL-POOL *ctl-pool (ACS-pool)* IS EMPTY
- ▷ UNLABELED CTL-POOL *ctl-pool (ACS-pool)* IS EMPTY
- ▷ LABELED CTL-POOL *ctl-pool (ACS-pool)* DOES NOT EXIST
- ▷ UNLABELED CTL-POOL *ctl-pool (ACS-pool)* DOES NOT EXIST

These messages are displayed if a CTL-pool name was passed and an error status of pool empty, pool low, or pool not found is returned from the mount request. The ACS-pool is the decimal ACS pool number from the translation table entry corresponding to the CTL-pool name.

- ▷ LABELED SCRATCH POOL *number* DOES NOT EXIST
- ▷ UNLABELED SCRATCH POOL *number* DOES NOT EXIST

A mount request for a scratch tape of the indicated type has just failed because it requested a scratch pool that was not configured on the server. Because CSC maps label types to server scratch pool numbers, there may be an error in the mapping definition. Another possibility is that a server command was used to delete the indicated scratch pool.

- ▷ LABELED SCRATCH POOL *number* IS EMPTY
- ▷ UNLABELED SCRATCH POOL *number* IS EMPTY

A mount request for a scratch tape of the indicated type has just failed because the server had no tapes in the indicated scratch pool. Some action should be taken to replenish the indicated pool.

- ▷ LABELED SCRATCH POOL *number* IS LOW
- ▷ UNLABELED SCRATCH POOL *number* is LOW

A mount request for a scratch tape of the indicated type has just completed. The number of tapes remaining in the server scratch pool after this mount is below the low-water threshold defined on the server. Some action should be taken to replenish the indicated pool.

- ▷ *media-types* element does not exist

The element named “media-types” does not exist in the CSC product file. Without this element, CSC cannot verify that its names and types for library media are equivalent to those on the server. CSC continues to operate using the names and types from the server.

- ▷ MOUNT *volser* ON *drive* ALREADY REQUESTED

This is a response to a console-initiated mount that is identical to a mount that has not yet completed. The console-initiated mount is not done.

- ▷ MOUNT *volser* ON *drive* COMPLETE

An ACS mount request has completed normally.

- ▷ MOUNT *volser* ON *drive* FAILED WITH STATUS *status*

An ACS mount failed. Status indicates the reason for the failure as reported by the server. Another more descriptive message precedes this one for several of the more common types of failures.

- ▷ MOUNT *volser* ON *drive* TIMED OUT

The indicated ACS mount did not complete within two minutes of its submission to the server. The OS 2200 service message will retry the mount if necessary.

▷ MOUNT *volser* ON *drive* WILL NOT BE DONE

A condition reported by a prior message will prevent the completion of an ACS mount of the indicated volume. This is always the second of two messages.

▷ MOUNT *volser* ON *drive* CANCELED BY DISMOUNT

This is the response to a console-initiated mount request that will not be done because a subsequent dismount request was received for the indicated transport before processing of the mount could start. Rather than mount the volume and then immediately dismount it, both this mount request and the subsequent dismount request are discarded.

▷ No drives configured

▷ No drives to work with

These messages both indicate that CSC has discovered it has not drives to control. The first form occurs if either the CSC configuration element contains no drives or if every drive in the configuration element is not known to both Exec and the server. The second form occurs with CARTTAPELIB\$ if, after configuration processing, Exec cannot allow the remaining drives to be CTL-controlled. This typically occurs if another CSC already controls those drives.

▷ NO TAPE FOUND ON *dddddd* TO DISMOUNT

A *CSC DISMOUNT command was issued on the system console, but no tape was found on the specified drive name.

▷ Path *path-name* is not usable

The indicated path is not available for use in communicating with the server.

- ▷ POOL *ctl-pool* FOR LABELED SCRATCH ON *drive* IS UNDEFINED
- ▷ POOL *ctl-pool* FOR UNLBLD SCRATCH ON *drive* IS UNDEFINED

An @ASG statement for a scratch tape requested a *ctl-pool* for which there is no corresponding ACS pool in the CSC configuration.

- ▷ PRINT\$ has been breakpointed

A *CSC BRKPT command has been successfully completed. This response occurs when the PRINT\$ file of the CSC run is not breakpointed when CSCTM is executed.

- ▷ PRINT\$ breakpointed to new cycle of *CSC-PRT-FILE

CSC created a +1 cycle of the CSC-PRT-FILE and breakpointed PRINT\$ into that file. This response form is used when the PRINT\$ file of the CSC run is breakpointed to a file when CSCTM is executed. This breakpoint is initiated by either a *CSC BRKPT command or automatically when the CSC-PRT-FILE to which CSC is breakpointed reaches 90% of its maximum size.

- ▷ Request Denied - CSC is initializing

The keyin that was entered is not allowed while CSC is initializing.

- ▷ Requested path/interface *name* is down
- ▷ Requested path/interface *name* is unavailable

This is displayed when a *CSC ACTIVATE keyin is entered for a path or interface that is down or not operational.

- ▷ Run does not have SSCARTLIB privilege

SB5R4 places a restriction on the interface used by CSC to communicate with OS 2200. To overcome this restriction, CSC must have a userid with the SSCARTLIB privilege *or* CSC must run under the Security Officer's userid. If this requirement is not satisfied, CSC produces the preceding message and terminates.

- ▷ The CSC keyin is *site-keyin*
- ▷ The CDI keyin is *site-keyin*

This message displays during CDI or CSC initialization if the console keyin was changed by site configuration to other than *CDI or *CSC. This message tells the site-defined keyin name.

- ▷ UNABLE TO TERMINATE CSC USER INTERFACE ACTIVITY NORMALLY
- ▷ CSC WILL ABORT THE ACTIVITY TO ALLOW THE PROGRAM TO FIN

At CSC termination, the CSCUI input activity could not be terminated.

- ▷ UNEXPECTED CARTTAPELIB\$ STATUS: *status*

This message indicates CSC received a CARTTAPELIB\$ status that it was not expecting, resulting in CSC termination.

- ▷ UP OF *name* SUCCESSFUL
- ▷ UP OF *name* FAILED

AN UP request was made for an interface or a path. This message indicates the result of that request.

- ▷ VOLUME *volser* EJECTED

The indicated volume was successfully ejected from the ACS.

- ▷ VOLUME *volser* ENTERED

The indicated volume was successfully entered into the ACS.

- ▷ VOLUME *volser* HAS AN UNREADABLE LABEL

The external label on the indicated volume could not be ready by the vision system in the ACS. Another message will follow this one to report the request failure.

▷ VOLUME *volser* IS IN ACS *acs-name*

A mount request for *volser* has just failed because the tape was in a different ACS from the requested drive. This message tells which ACS contains the requested volume. The mount failure message will follow this one.

▷ VOLUME *volser* IS IN USE

The volume in a request is either in use in the ACS or reserved by the server for another client. Another message will follow this one to report the request failure.

▷ VOLUME *volser* IS MISPLACED

The volume in a request is known to the server, but could not be found by the server. Another message will follow this one to report the request failure.

▷ VOLUME *volser* IS NOT IN THE ACS

The volume in the request is not known to the server. Another message will follow this one to report the request failure.

▷ VOLUME *volser* NOT EJECTED WITH STATUS *status*

The indicated volume was not ejected from the ACS. Status indicates the reason for the failure as reported by the server.

▷ VOLUME *volser* NOT ENTERED WITH STATUS *status*

The indicated volume was not entered into the ACS. Status indicates the reason for the failure as reported by the server.

▷ *volser* ALREADY MOUNTED ON *drive*

This is the response to a console-initiated mount of a volume onto the drive on which it is already mounted.

- ▷ Warnings detected in configuration
- ▷ Check print file

These messages are displayed at initialization if warnings have been encountered in the CSC\$PARAM file. The warning messages are printed in the PRINT\$ file.

SUBSYSTEM ERROR MESSAGES

ERROR DURING INITIALIZATION OF *control-unit* CAUSES DN OF *path*

During a system boot, OS 2200 initializes each path to a 4780 control unit to mark it as an active interface. This message displays if the initialization process fails, usually because all cartridge drives configured to the control unit are already assigned to another client. An internally-generated DN keyin brings down the path.

ERRORS DURING UP OF *iop* CAUSES DN OF *device*

When an IOP is brought UP, OS 2200 attempts an exclusive assignment of all 4780 cartridge drives configured under the IOP. This message displays when a cartridge drive can't be assigned, usually because the cartridge drive is already assigned to another client. The IOP is still brought into the application, but a DN keyin is internally generated for the cartridge drive.

ERRORS DURING DN OF *device*: UNABLE TO RELEASE *device*

Removing a cartridge drive from the application with the DN keyin causes the client to free or release exclusive use of the cartridge drive. This message appears when the release is unsuccessful.

ERRORS DURING DN OF *path*: UNABLE TO RELEASE *device*

Using the DN keyin to remove a path from the application frees or releases exclusive use of any cartridge drives configured in the path. This message displays when the release is unsuccessful.

ERRORS DURING TU OF *device*: UNABLE TO RELEASE *device*

The TU keyin forces the release of a cartridge drive from another client. This message displays when the release fails.

ERRORS DURING UP OF *path* CAUSES DN OF *device*

When a path to a 4780 control unit is brought UP, all cartridge drives configured for the path are exclusively assigned. This message displays when a cartridge drive can't be assigned on the designated path. The interface is still brought into the application, but a DN keyin is internally generated for the cartridge drive.

INITIALIZATION OF *device* FAILED CAUSES DN OF *device*

During a system boot, each UP and available 4780 cartridge drive is exclusively assigned. This message displays when the assignment fails. This is often because the cartridge drive is assigned to another client.

TU OF *device* FAILED: TU NOT ALLOWED ON UP/RV OR NA UNIT

A TU keyin only works on a cartridge drive that is DOWN. This message displays when a TU keyin is used on a cartridge drive that is in a state of UP, RV or NA.

UP OF *path* FAILED: UNABLE TO ACQUIRE PATHS TO TAPE UNITS

This message displays when no paths can be brought into the application. No processing continues for the UP keyin.

UP OF *device* FAILED: UNABLE TO ACQUIRE THE UNIT

This message displays when an attempt to exclusively assign the cartridge drive fails. This is usually because the 4780 control unit is off-line, or because the channel interface is disabled, or because the 4780 control unit has been acquired by another client.

UP OF *device* FAILED: UNIT ACQUIRED BY ANOTHER SYSTEM

This message displays when an attempt to exclusively assign the cartridge drive fails because the cartridge drive is already assigned to another client.

device - reel# DRIVE ASSIGNED TO ANOTHER SYSTEM

The detected condition is a rejection of an I/O request to a cartridge drive. It occurs because another client system has acquired exclusive use of the cartridge drive using CULP. OS 2200 internally generates a DN of the cartridge drive. This action is taken to eliminate the possible interference with the client that has somehow seized control of the cartridge drive. How control was taken away from this client is not considered in this action. The concern at the time of this message is ensuring that only one client has control of the cartridge drive.

device - reel# DATA STREAMING IS NOT OPERATIONAL

The 4780 control unit and the Unisys channel use a method known as *data streaming* to coordinate the transfer of data to and from cartridge drives. If some condition prevents this form of transfer, a much slower method is used. This message informs the system operator that the 4780 subsystem is no longer using the data streaming mode. This condition continues until either the current cartridge tape is dismounted, or until the 4780 control unit detects a client reboot. If this condition persists, notify your Unisys CSE to check the 4780 interface settings.

EXEC MESSAGES

When CSC uses the CARTTAPELIB\$ Exec interface, the following messages may be displayed by CSC rather than the Exec. These messages are documented in the *OS 2200 System Console Messages Reference Manual*.

I/O error *nn* tape cannot be read
Duplicate reel *reelid* on device *device*
Reel *reelid* cannot be mounted on *device*
device not compatible with assignment
reelid can only be mounted on *device*
reelid is already assigned by run *runid*
Write ring required in reel *reelid* on *device*
Remove ring in reel *reelid* on unit *device*
Premounting of cartridge is not allowed on *device*
Is tape on *device* a labeled tape
device NORING *runid*
device NOTRDY *runid*
device wrong reel, load *reelid*
device wrong reel, load unlabeled *reelid*
device wrong reel, load initialized volume
device bad label, load *reelid*
device bad label, load unlabeled *reelid*
AT n *device* load properly labeled reel

6. PROBLEM SOLVING

This chapter describes how to solve certain operational problems. Topics include:

- **How failures affect automated tape operations.** This section explains the theory behind server operations, and how requests and components can fail.
- **Handling abnormal conditions.** This section describes how to handle general problems.
- **Handling specific error conditions.** This section explains how to resolve specific errors, and refers to the messages listed in Chapter 5, “Message Reference.”
- **Reporting CSC problems.** This section describes procedures you can use to collect and submit diagnostic information for problems that are not addressed by the above topics.

HOW FAILURES AFFECT AUTOMATED OPERATIONS

The client communicates with the ACS through a control path consisting of various software and hardware components. Please refer to Chapter 1, “Overview of CSC,” for descriptions of these components.

If any control path component fails, user jobs may not be able to access the ACS. You can override the automatic operations of the ACS by manually mounting and dismounting cartridge tapes.

OS 2200 uses the state of CSC to determine the status of the control path. When a control path component or CSC is down, OS 2200 takes action based on your configuration parameters.

Automated tape operations can only occur when the control path is completely functional. Control path failures are reported through messages sent to the operator console. These messages usually come from CSC. If CSC is not active, then OS 2200 sends its own messages.

Following are the possible states for the control path.

CSC Down

OS 2200 cannot request automated tape handling when CSC is not executing and in the ACTIVE state. If a request occurs when CSC is not ready to accept it, OS 2200 produces one of the following messages:

```
ACS MOUNT FOR volume ON tape-unit CANNOT BE SENT
ACS DISMOUNT FOR tape-unit CANNOT BE SENT
```

Starting CSC and waiting for it to complete initialization is the corrective action for this control path state.

Control Path Inoperative

CSC produces two types of console messages to report a control path failure. One type reports the failure of a specific request with the status NI_FAILURE (network interface failure). The other type is an informative message from CSC that is repeated periodically until the control path becomes operational. Following are the formats of the two types of messages:

```
MOUNT volume ON tape-unit FAILED WITH STATUS NI_FAILURE
CSC IS WAITING FOR problem-description
```

The corrective procedure is to resolve the problem described in the CSC IS WAITING message.

Normal

When the control path is fully functional, CSC produces only MOUNT and DISMOUNT completion messages on the operator console. Most MOUNT and DISMOUNT FAILED messages are not an indication of a control path failure. These messages tell of a condition detected by the server and reported to CSC through the control path.

Failure of ACS Requests

CSC sends requests to the ACS through ACSLS or NCS, which runs on the server. CSC and ACSLS or NCS follow strictly-defined rules about how to send and acknowledge requests. If a request fails, it may produce an error condition as described for each request.

LOCK Request

The LOCK request is for server housekeeping. It instructs the server to reserve use of the locked component for this client system. When an operator UPs a cartridge drive, CSC sends a LOCK request to reserve use of the cartridge drive.

If a transport lock fails, CSC periodically retries the lock request. CSC does not need the server lock. It functions normally even when lock requests fail.

UNLOCK Request

The UNLOCK request is for server housekeeping. It instructs the server to release exclusive use of an ACS cartridge drive. When an operator DOWNs a cartridge drive, CSC sends an UNLOCK request to release use of the cartridge drive.

If a transport unlock fails, another client that depends upon server locks may not be able to access that cartridge drive without intervention.

MOUNT Request

The MOUNT request loads a cartridge tape. OS 2200 sends a MOUNT request for each console LOAD message.

If the ACS doesn't satisfy a MOUNT request within two minutes, OS 2200 generates a service message. CSC uses service messages to retry the mount. Various reasons may be reported by the server as the cause of a mount failure. Many of these initiate recovery processing in CSC.

MOUNT Response

The MOUNT response informs OS 2200 that the ACS mounted the specified cartridge tape. OS 2200 uses a MOUNT response only to answer a request for an unlabeled cartridge tape. If OS 2200 doesn't receive a MOUNT response within ten minutes, the operator is prompted to enter a VOLUME-ID.

DISMOUNT Request

The DISMOUNT request unloads a cartridge tape or tapes. OS 2200 sends a DISMOUNT request at the same time it generates a request to rewind and unload the specified cartridge tape. OS 2200 generates a DISMOUNT request only once, and doesn't time it. If a DISMOUNT request is not sent, the ACS will leave the unloaded, rewound cartridge tape in the cartridge drive. CSC has logic to dismount the cartridge the next time that drive or cartridge is requested for use.

VOLUME STATUS CHANGE Request

The VOLUME STATUS CHANGE request, coming from the TLMS, changes the scratch status of a cartridge tape. A timeout or failure to send this request returns an abnormal completion status to the TLMS. The result is that the ACS and TLMS scratch statuses may fall out of synchronization.

QUERY MOUNT Request

The QUERY MOUNT request, which is timed, determines the cartridge drive to use (if any) to satisfy a cartridge tape assignment.

An assignment for a non-scratch cartridge tape may rely on the response to this request to select either an ACS or non-ACS cartridge drive. If this request fails or times out, OS 2200 may allocate the wrong type of cartridge drive.

An assignment in a single ACS environment using U47L relies on the response to this request to determine the LSM in which to allocate a cartridge drive. If this request fails or times out, OS 2200 might allocate a cartridge drive in a different LSM from the requested cartridge tape. If this occurs, the ACS must move the cartridge tape from its home LSM, through the pass-through port, to the allocated cartridge drive in the other LSM. This slows the mount process.

In a multi-ACS environment, if this request fails or times out, OS 2200 will allocate a cartridge drive in the run default ACS. The run default ACS may be different from the ACS where the cartridge tape resides. If this occurs, the cartridge tape must be ejected from its ACS and entered into the allocated, run default ACS.

Failure of Control Path Components

Failure in any control path component may produce the following results, as described for each component.

CDI

If CDI fails, use any of the CDI diagnostic tools described in Chapter 3, “CDI Command Reference,” or refer to the *CDI Troubleshooting Guide*. While CDI is unavailable, CSC will periodically send the following message to the system console, if CDI is the only interface available to the server.

```
CSC is waiting for a path that is up
```

CMS

If CMS fails, refer to the *CMS1100 Operations Reference Manual* for procedures to Down the CMS process or to restart CMS. While CMS is unavailable, CSC will periodically send the following message to the system console, if CMS is the only interface available to the server.

```
CSC is waiting for a path that is up
```

CSCUI Common Bank

If the CSCUI common banks are reloaded while CSC is UP, CSC will abort. If you must reload the CSCUI common bank, terminate CSC, then restart it.

PATH

When the active path fails, CSC will display a message and then attempt to switch to another available path. The following messages are displayed:

```
Active path ALPHA has failed  
Control path now using BRAVO
```

CSC periodically checks the state of all known paths. CSC does not try to use paths that are down. If the active path is downed, CSC will attempt to route the control path through another path. If the last usable path fails, then CSC will periodically send the following message to the operator console:

```
CSC IS WAITING FOR A PATH THAT IS UP
```

When CDI detects a CPA equipment check or equipment error, it resets the CPA. If this does not resolve the problem, first down and then up the CPA using the *CDI DN and *CDI UP commands. The following messages are displayed if CDI has detected this condition:

```
Equipment check[error] - Unit recovering  
CPAA(ALPHA) Equipment check[error].  
CPA Reset due to Ethernet Controller Error  
CPA Reset Successful
```

LAN Hardware or Server Hardware

CSC, CMS, and CDI cannot detect failures of LAN hardware or the server platform. If such a failure occurs, CSC cannot establish a connection to the server. CSC will periodically send the following message to the operator console as long as it cannot establish a connection to the server:

```
CSC IS WAITING FOR A SERVER CONNECTION
```

Server Software

If CSC can communicate with the server but the ACSLS or NCS software is not running on the server, CSC will periodically send the following message to the operator console:

```
CSC IS WAITING FOR ACSLS TO START
```

```
CSC IS WAITING FOR NCS TO START
```

Indeterminate Control Path Component Failures

CSC can accurately identify the source of a failure only when the condition exists before a request is initiated. If a failure occurs after a request starts and the failure source cannot be isolated, then CSC sends the following message to the operator console:

```
CSC IS WAITING FOR AN UNKNOWN RESOURCE
```

HANDLING ABNORMAL CONDITIONS

This section describes how to handle abnormal conditions that may occur in the data and control paths.

Data Path Conditions

The data paths for ACS cartridge drives and standalone cartridge drives are identical. To handle abnormal conditions on the ACS cartridge drive data path, you can use the same approaches as you would for manual cartridge tape operations.

Control Path Conditions

Error conditions in the control path can come from OS 2200, CSC, the communication link, or the server.

OS 2200 Errors

OS 2200 can fail because of an EXEC stop or a non-fatal error.

- **An EXEC stop** may occur for any number of reasons. These may or may not pertain to ACS or server operations. If an EXEC stop occurs, obtain a dump (if possible), then reboot the client and restart CSC.
- Dynamic date keyin changes can adversely affect CSC operations. If a time/date change is necessary, it is recommended that CSC and CDI be terminated, and restarted once the change has been made. If this procedure is followed, CSC operation will not be affected by the change.
- The CSC operating system functions report **non-fatal errors** via console messages, as in:

```
SYSTEM ERROR 440-xx SEQUENCE msg IPO P-ADDRESS: address
```

where *xx* is the error number, *msg* is the sequence number, and *address* is the program address.

CSC Errors

If the CSC run terminates, the CSC operating system functions react by routing mount request messages to the system console.

You can respond by:

- restarting CSC as soon as possible. CSC will then route any held requests to the ACS.
- using manual methods to mount and dismount cartridge tapes in the ACS.

Communication Errors

Communication errors can occur in:

- communication hardware (CPA, HLC, DCP, NIOP)
- LAN hardware
- channel connections
- communications software (CDI or CMS)

CSC assumes that an error in the communication link is temporary, and holds requests until communications are reestablished.

In general, to resolve a communication error, you should:

- Try to bring the component back on-line
- If the failure is with communications software, restart the software. If you have multiple interfaces, DOWN the interface for the failed component or software and UP a different interface
- If you can't resolve the problem quickly, terminate CSC to avoid excessive request queuing

CPA Errors

In OS 2200, each CPA is configured as a control unit with three devices. These devices must be UP to OS 2200 for the CPA to function. When CDI initializes, it attempts to bring UP the CPAs.

When CDI brings UP a CPA, CDI performs a functional POC test. For CDI to execute successfully, it must assign all three OS 2200 CPA devices. For a CPA to operate properly, the following OS 2200-defined components must be UP and functioning:

- the IOP
- the block multiplexer channel
- the CPA control unit
- the CPA itself

If any of these components are DOWN, CDI detects that the CPA isn't functioning and brings it DOWN.

To check on the last known status of your path(s), use the *CSC STATUS PATH command. The phrase "last known" is significant because CSC can take

up to one minute before it updates the known status of a path. Following are the statuses and a brief description of each.

DOWN The path is down in CDI. CSC sets a path to this state when the path is set to down by the DOWN command or when the path becomes unusable due to a communications software or hardware error.

UNTESTED No communication with the server has been attempted using this path. CSC puts the path in this state when the path state changes from down to up via the UP command. CSC periodically initiates a test of paths in the UNTESTED state.

UNUSABLE Communication with the server was tried and failed through this path. There is no delay between failure detection and setting the path to the UNUSABLE state. CSC periodically initiates a test of paths in the UNUSABLE state.

AVAILABLE Successful communication with the server has occurred through this path. It is available for use. CSC periodically initiates a test of paths in the AVAILABLE state to verify that they are actually available.

ACTIVE CSC uses the path in this state for sending requests to the server. In the absence of any server requests, CSC periodically initiates a test of the ACTIVE path to verify that communication with the server is possible.

If you suspect that a CPA is DOWN, use the following procedures to resolve the problem:

1. Determine if the CPA is indeed DOWN by using the *CDI FS command to check the CPA's status.
2. Bring UP the CPA by using the *CDI UP command. CDI will run the CPA functional POC test during the UP process. If the test fails, check the CDI configuration, correct the problem, and reissue the *CDI UP command.

CMS Errors

If you are using CMS as the interface, follow the standard procedures listed previously for resolving the problem.

Server Errors

CSC views any server component failure as a communication failure. To correct the problem, refer to the previous section on "Communication Errors" (p. 6-8).

HANDLING SPECIFIC ERROR CONDITIONS

This section describes how to handle these specific types of errors:

- initialization problems
- mount problems
- dismount problems
- write ring problems
- software failures

Any of these errors can occur during normal operations, or because of incomplete recovery. Even though recovery is attempted automatically, some unresolved or unrecovered conditions may remain after the recovery completes. Manual intervention may be required to resolve many of these problems.

For more information on specific error messages listed in this section, please refer to Chapter 5, “Message Reference.”

Initialization Problems

CSC and the server each maintain status information on ACS cartridge drives and cartridge tapes. On startup, CSC synchronizes the CSC and server statuses. Although CSC maintains enough information to handle the status of the cartridge drives, it may not have enough information to determine the status of cartridge tapes in use or mounted in the ACS. Synchronization problems may produce error messages.

If CSC is started while another CSC run is active, the following console message is displayed:

```
▷ runid DETECTED ANOTHER ACTIVE CSC
```

When this condition is detected, the second CSC will automatically terminate.

```
▷ CSC Drive count: ACS nn (UP), ACS nn (DN), non-ACS nn
```

If this message shows that the number of non-ACS drives is non-zero, it indicates that there are drives in the CSC\$DRIVE element that either have bad server addresses or do not exist. Check the CSC\$DRIVE element for valid drive names and addresses.

If this message shows that the number of ACS UP drives is zero, it indicates that no ACS drives are available and no mount activity can take place until drives are UPed on the client.

The following messages are displayed at initialization if fatal errors have been encountered in the CSC\$PARAM file:

- ▷ Fatal errors detected in configuration
- ▷ Check print file

If these messages are displayed the CSC run will end. Check the PRINT\$ file to determine what errors were encountered, and then correct the errors. After the errors are corrected, ST CSC again.

- ▷ Warnings detected in configuration
- ▷ Check print file

If these messages are displayed the CSC run will continue. Check the PRINT\$ file to determine what warnings were encountered, and then correct the warnings.

Mount Problems

During automated cartridge tape operations, the CSC operating system functions normally suppress the display of load messages on the system console. When a problem prevents an ACS cartridge tape mount, CSC will display an error message.

OS 2200 MOUNT requests intended for the ACS will timeout if they cannot be passed from OS 2200 to CSC within two minutes. OS 2200 displays this timeout condition with the console message:

- ▷ MOUNT *volser ON transport* TIMED OUT

The following messages indicate two other types of mount problems:

- ▷ MOUNT *volser ON transport* FAILED WITH STATUS *status*
- ▷ MOUNT *volser ON transport* WILL NOT BE DONE

If a cartridge tape doesn't mount, and no error message displays, then the MOUNT request wasn't sent to the ACS because a DISMOUNT request probably occurred simultaneously. In this situation, the following console message is displayed:

▷ MOUNT *volser* ON *transport* CANCELED BY DISMOUNT

To correct the problem, check for activity on the specified cartridge drive. If there isn't any activity, check for a control path failure (see “

Software Failures,” page 6-15, later in this section).

Also, if a CTL-pool is used on the @ASG or @CAT statement, these messages indicate a problem with the CSC or ACSLS or NCS configuration.

- ▷ POOL *ctl-pool* FOR LABELED SCRATCH ON *transport* IS UNDEFINED
- ▷ POOL *ctl-pool* FOR UNLBLD SCRATCH ON *transport* IS UNDEFINED

Check the CSC\$PARAM element. This message indicates that there is either no CTL-pool matching the requested *ctl-pool* in the CSC configuration, or there is no ACS pool corresponding to the requested *ctl-pool* in the CSC configuration.

- ▷ LABELED CTL-POOL *ctl-pool* (*ACS-pool*) DOES NOT EXIST
- ▷ UNLABELED CTL-POOL *ctl-pool* (*ACS-pool*) DOES NOT EXIST

Check the ACSLS or NCS configuration and the CSC\$PARAM element. The ACS pool corresponding to the requested CTL-pool does not exist on the server. Either the pool should be configured on the server or the TRANSLATE_POOL configuration statement should be changed to specify an existing pool.

Dismount Problems

When OS 2200 initiates a Rewind with Interlock (REWI) request for an ACS cartridge drive, CSC intercepts and translates the request into a DISMOUNT request. CSC then forwards the DISMOUNT request to the server.

Most dismount problems generate the OS 2200 message:

- ▷ 0-transport *iii/ccc/cucu* INT-RQ REWI EXEC 8 ABM

The following messages also indicate a dismount problem:

- ▷ DISMOUNT *volser* WILL NOT BE DONE
- ▷ DISMOUNT *volser* FAILED WITH STATUS *status*
- ▷ DISMOUNT *volser* ALREADY REQUESTED
- ▷ DISMOUNT *volser* TIMED OUT

Write Ring Problems

If a write ring problem occurs, the following message displays on the system console:

```
▷ 0-WRITE RING REQUIRED IN volser ON UNIT transport - A,E
```

If OS 2200 indicates a write ring problem, take the following actions:

1. Don't answer the OS 2200 message immediately.
2. At the system console, enter:

```
*CSC DISMOUNT vol_id drive_id
```
3. ACSLM will verify the *vol_id* and *drive_id* in the ACSLS or NCS database, and instruct the LSM robot to:
 - Move to the specified cartridge drive
 - Dismount the cartridge tape from the cartridge drive
 - Return the cartridge tape to an available storage cell in the ACS
4. At the system console, enter:

```
*CSC EJECT vol_id
```

The LSM robot will automatically select a Cartridge Access Port (CAP) and place the cartridge tape in it.
5. Remove the cartridge tape from the CAP.
6. Turn the thumbwheel on the cartridge tape to the write-enabled position.
7. At the system console, enter:

```
*CSC ENTER cap_id
```

The format for *cap_id* is *acs#,lsm#,cap#*. The LSM robot will automatically enter the cartridge tape into the LSM.
8. To mount the cartridge tape using the Command Processor, enter:

```
MOUNT vol_id drive_id
```
9. Answer "A" to the OS 2200 message described above.

Software Failures

Software failures fall into three general categories:

- CSC failures
- server failures
- communications failures

All three failures cause automated cartridge tape operations to cease. CSC and server failures are covered in the following section. Please refer to the *CDI Troubleshooting Guide* for information about CDI communication failures. Please refer to the *CMS1100 Operations Reference Manual* for information about CMS communications failures.

CSC Failures

If CSC aborts or is terminated on the client, the server will continue to service requests that it received prior to the failure. Eventually, server and ACS operations will cease because the server is no longer receiving requests.

When initialized, CSC attempts to reestablish its pre-failure environment. This is done dynamically through a series of requests that are sent to both the server and OS 2200. These requests assist CSC in determining which cartridge tapes are mounted in which cartridge drives.

CSC uses status information returned by the server and OS 2200 to reestablish its cartridge drive information table. This table contains the following information for each cartridge drive:

- cartridge drive name
- ACS location
- LSM location
- panel location
- cartridge drive location
- volume ID (i.e., what CSC believes to be on the cartridge drive)
- lock ID

In addition to this data, the cartridge drive information table also contains the status for each cartridge drive. Cartridge drive statuses include:

- **UNKNOWN.** The initial state of a cartridge drive. No information is known about its operational status. This state may also be entered to force a drive state resynchronization between OS 2200 and the server.
- **SYNCHRONIZE.** The state of a cartridge drive is unknown and its internal state is being synchronized with OS 2200 and the server. The synchronization function will determine the cartridge drive's appropriate state upon completion.

- **NON-ACS.** Cartridge drives that are defined during the CSC configuration process, but are not known to the server, are placed into this state. No automated operations will be performed on cartridge drives in this state.
- **DOWN.** Cartridge drives that are in an ACS, but are considered down (DN) by OS 2200, are placed into this state. OS 2200 will not perform any operations on cartridge drives in this state. This state changes to “synchronize” when the OS 2200 drive status is changed to up (UP) or reserved (RV).
- **AVAILABLE.** The cartridge drive is available for use by OS 2200, and no cartridge tape is mounted.
- **LOADING.** This state shows that a mount operation is in progress for this cartridge drive. The mount task will complete and change the state to either “loaded” or “available” based on the success or failure of the mount operation.
- **LOADED.** The cartridge drive is available for use by OS 2200, and a cartridge tape is currently mounted.
- **UNLOADING.** A dismount operation is in progress for cartridge drives in this state. The dismount task will change the state to “available” when the operation is completed.

CSC recovery is automatic and transparent to the operator. If the recovery is successful, CSC and the server reconcile the statuses of cartridge tapes and cartridge drives, and automated cartridge tape operations resume. If the recovery fails, error conditions or anomalies may occur.

Server Failures

If ACSLS or NCS software aborts or is terminated, CSC continues sending requests to the server, momentarily unaware that an ACSLS or NCS failure has occurred. Eventually, these requests will time out, notifying the operator that a failure has occurred. CSC will continue sending requests, and these requests will continue to time out until ACSLS or NCS is restarted. If a great deal of time passes before the problem is resolved, you should terminate CSC and restart it when the server is available. During recovery, the server performs the following processes for each ACS in the library:

- verifies that all online ports can communicate with the ACS.
- verifies that the ACSLS or NCS database matches the LMU configuration.
- varies the ACS, LSM, and CAPs online, if possible.

Once these processes have completed successfully, server resumes accepting requests from CSC.

REPORTING CSC PROBLEMS

Anomalous conditions can occur in CSC operation due to new or unexpected hardware or software events or unusual combinations of events. If you encounter such a condition that is not addressed by the problem solving techniques in this chapter, the problem should be reported. This section describes how to gather and process diagnostic information that can be used for problem analysis.

Storage Technology Corporation distributes the CSC 5R1 software and documentation. Problem reporting is based on the type of server with which CSC is operating.

- When CSC is used with a server running the Nearline Control Solution software, problems are reported to Storage Technology Corporation.
- When CSC is used with a server running the ACSLS software, problems are reported to Unisys Corporation

Useful Information for Problem Analysis

During operation, CSC, OS 2200, and the library server either use or produce the following information:

- The CSC configuration elements tell about your particular configuration.
- The CSC print file includes messages sent to the operator console and additional information about the CSC operating environment.
- CSC produces a log file of significant operating events. Enabling debug mode using the *CSC DEBUG ON keyin causes CSC to consider many more events as significant and includes them in its log file. For repeatable problems, it is usually helpful to use CSC log information collected with debug mode enabled. The CSC log file gives the most complete picture of the CSC operating environment.
- OS 2200 uses the CSC diagnostic file to record the contents of memory used by CSC at the time that CSC terminates. If CSC aborts, either from an internal condition or via the *CSC EABT keyin, the diagnostic file contains information crucial to analyzing the problem. After a CSC abort, the final records that were not yet written to the CSC log file appear only in the diagnostic file.
- The communication software (CDI, CMS, or CPCOMM) has the ability to record information about data passing between CSC and the library server software. They also record events that could affect this data transfer.
- The OS 2200 console log shows messages produced by CSC and system related activity by other programs. When combined with operator keyins and

their responses, these give a more complete picture of the operating environment.

- The OS 2200 system log file includes all of the messages from the console log. It also includes entries for tape unit assignments and releases, entries for tape label processing and I/O errors, and CSCUI created entries. This gives the most complete picture of the OS 2200 operating environment.
- Both variations of the library server software create one or more files that log significant events related to automated tape operation. From the perspective of CSC problem analysis, the events around the time of the CSC problem are occasionally helpful.

Various combinations of the above produce a more complete picture of the tape automation system. For example, the CSC log combined with the OS 2200 system log give a fairly complete picture of all tape automation activity on the OS 2200 system. The CSC log and the communication log and the server log give a fairly complete picture of interactions between CSC, the server software, and the ACS hardware.

How To Collect Diagnostic Information

CSC includes two runs to assist in collecting diagnostic information. CSCLOG collects and formats information from the CSC log file. CSCDUMP processes the CSC diagnostic file into a usable format. Both of these runs also create a listing of the CSC configuration elements. The output of CSCLOG and CSCDUMP are in the cataloged files SYSSLIB\$*CSCLOG-PRT and SYSSLIB\$*CSCDMP-PRT, respectively.

The examples in this section assume that the qualifier for the CSC installation is SYSSLIB\$. Use the *CSC REPORT FILES keyin to determine the qualifiers, filenames, and cycles at your site.

Starting CSCLOG and CSCDUMP

The CSCLOG and CSCDUMP runs are copied to the CSC product file during CSC installation. They are started using the following keyins:

```
ST SYS$LIB$*CSC.CSCLOG, val, , acct/uid
ST SYS$LIB$*CSC.CSCDUMP, val, , acct/uid
```

Where:

val determines which log or diagnostic file is processed:
 If val is 0 (zero) or omitted, the current cycle of the log or diagnostic file is processed.
 If val is 1, the minus 1 cycle of the log diagnostic file is processed.
 If val is 2, a message is sent to the operator console to get the cycle number.

acct/uid is the account and userid for the CSC run.

Alternatives for Collecting Diagnostic Information

The following table lists the possible types of CSC diagnostic information and tells how it is collected.

Table 6-1. CSC Diagnostic Information and How it is Collected

Diagnostic Information	How to Collect
CSC configuration elements	Run either CSCLOG or CSCDMP. An alternate is to manually copy the elements CSC\$PARAM, CDI\$PARAM, CSC\$DRIVE, CSCUI\$-CONFIG from the CSC parameter file.
CSC print file	Find the CSC print in the file SYS\$LIB\$*CSC-PRT-FILE for the time of the problem.
CSC log file	Run CSCLOG, An alternative is to submit the file SYS\$LIB\$*CSC-LOG-FILE. If transferred via FTP, it must be sent in binary mode.
CSC diagnostic file	Run CSCDMP. An alternative is to submit the file SYS\$LIB\$*CSC-DIAG\$. If transferred via FTP, it must be sent in binary mode.
OS 2200 console log	Submit either a listing of the console log or the OS 2200 system log from which the console log can be extracted.

What Diagnostic Information To Collect

Following are guidelines for what data to submit for analysis of a CSC problem based on the nature of the problem. If you have any doubt if information is needed, send it. In problem analysis it is better to have too much data to process than to miss the one crucial piece of information.

If possible, use the *CSC REPORT FILES keyin to determine the names and cycles of your CSC files.

Always send:

- The CSC print file
- The output from CSCLOG (or ALL of the equivalent raw data).

If CSC aborts, also send:

- The output from CSCDMP (or ALL of the equivalent raw data)

If there are problems with server communication, also send:

- Information from the server logs related to client communication. Refer to the appropriate library server documentation to determine the location of these logs.
- If available, the print output or log information from the communication program.

If multiple runs and/or OS 2200 were affected by the problem, also send:

- The OS 2200 console log, or
- The OS 2200 system log

APPENDIX A. CSC RUN DEFAULTS AND XQT OPTIONS

This appendix provides reference material on CSC run defaults and XQT options.

CSC RUN DEFAULTS

There are several run defaults, assumptions, and requirements when using CSC. Among them, CSC:

- Runs as a batch job with a runid of “CSC”.
- Requires real-time account/userid priority.
- Requires a userid with the SSCARTLIB privilege.
- Is a multi-activity program.
- Uses @RUN card options to give it 17 extra PCT blocks.
- Activities run at the highest real-time priority allowed for the account/userid.
- Reads a run-time parameter file when initialized.
- Allows console commands that begin with “*CSC,” regardless of the runid or generated runid.
- Maintains diagnostic files.
- Has a default debugging level of 2.

CSC XQT OPTIONS

Table A-1 lists the CSC XQT options. ‘R’ and ‘T’ are the default options. You should note that the ‘L’ option may create large amounts of output to the CSC PRINT\$ file.

Table A-1. CSC XQT Options

Option	Description
W	Invokes the dynamic timer for variable speed LSM(s).
L	Print traces in the PRINT\$ file.
R	Execute in real time.
T	Create the file CSC-LOG-FILE and capture trace entries.

INDEX

A

Abnormal conditions. *see also* Error conditions
communications, 6-8
control path, 6-7
CPA, 6-8, 6-9
CSC, 6-7
data path, 6-7
handling, 6-7
OS 2200, 6-7
ABORT command, CSC, 4-3
ACS requests
failure of, 6-3
ACSL database
ensuring integrity of, 2-12
ACTIVATE command, CSC, 4-4
Answerable Messages
reference, 5-2

B

Booting
CULP use and, 2-9
Breakpoint PRINT\$
BRKPT command, 3-2
BRKPT command, CDI, 3-2
BRKPT command, CSC, 4-5

C

Cartridge drive environment
establishing, 2-9
Cartridge tapes
locating within ACS, 2-13
physically mounting, 2-13

CDI

commands

DN, 3-3
FS, 2-8, 3-4
HELP, 3-6
ID, 3-7
PING, 3-8
STATUS, 2-8, 3-9
TCP ABORT, 3-10
TCP CONNECTIONS, 3-11
TCP HELP, 3-12
TCP KILL, 3-13
TCP STATUS, 3-14
TERM, 3-16
UP, 3-17

failures and, 6-5

monitoring, 2-7

obtaining status information, 2-8

CDI command reference, 3-1

Changing security levels, 2-4

CLEAR command, CSC, 4-6

CMD command, CSC, 4-7

CMS

failures and, 6-5

Collecting diagnostic information, 6-18

alternatives, 6-19

what to collect, 6-20

Command syntax notation, xiii

Commands

CDI

BRKPT, 3-2
DN, 2-9, 3-3
FS, 2-8, 3-4
HELP, 3-6
ID, 3-7
PING, 3-8
STATUS, 2-8, 3-9
TCP ABORT, 3-10
TCP CONNECTIONS, 3-11

Commands (*cont'd*)

CDI (*cont'd*)

- TCP HELP, 3-12
- TCP KILL, 3-13
- TCP STATUS, 3-14
- TERM, 3-16
- UP, 2-8, 3-17

CSC

- ABORT, 4-3
- ACTIVATE, 4-4
- BRKPT, 4-5
- CLEAR, 4-6
- CMD, 4-7
- CYCLE, 4-8
- DEBUG OFF, 4-9
- DEBUG ON, 4-10
- DEBUG STATUS, 4-11
- DISMOUNT, 4-12
- DOWN, 4-13
- EABT, 4-14
- EJECT, 4-15
- ENTER, 4-16
- HELP, 4-17
- LEVEL, 4-18
- MEMORY, 4-19
- MOUNT, 4-20
- QUERY, 4-21
- QUEUES, 2-7, 4-25
- REPORT, 4-26
- STATUS, 2-6, 4-28
- TERM, 4-31
- UP, 4-32
- VENTER, 4-33
- VERBOSE, 4-34

operating system

- DN, 2-10
- PM, 2-10
- RV, 2-10
- TU, 2-10
- UP, 2-10

Communications

- handling abnormal conditions, 6-8
- overview, 1-5

Connection

- checking with TCP STATUS command, 3-14
- removing data structure with CDI TCP KILL command, 3-13

Console interface

- overview, 2-3

Control path

- controlling, 2-7
- handling abnormal conditions, 6-7
- monitoring and controlling, 2-6

Control path components

- CDI failures, 6-5

- CMS failures, 6-5

- CSCUI common bank failures, 6-5

- failures and, 6-5

- indeterminate failures, 6-6

- LAN hardware failures, 6-6

- PATH failures, 6-5

- server hardware failures, 6-6

- server software failures, 6-6

- Controlling network, interface state, 2-8

- Controlling the control path, 2-7

CPA

- handling abnormal conditions, 6-8, 6-9

CSC

- checking memory via MEMORY command, 4-19

- checking queue status, 2-7

- checking status of CSC queues, 2-7

commands

- ABORT, 4-3

- ACTIVATE, 4-4

- BRKPT, 4-5

- CLEAR, 4-6

- CMD, 4-7

- CYCLE, 4-8

- DEBUG OFF, 4-9

- DEBUG ON, 4-10

- DEBUG STATUS, 4-11

- DISMOUNT, 4-12

- DOWN, 4-13

- EABT, 4-14

- EJECT, 4-15

- ENTER, 4-16

- HELP, 4-17

- LEVEL, 4-18

- MEMORY, 4-19

- MOUNT, 4-20

- QUERY, 4-21

- QUEUES, 4-25

- REPORT, 4-26

- STATUS, 4-28

- TERM, 4-31

- UP, 4-32

- VENTER, 4-33

- VERBOSE, 4-34

- communications overview, 1-5

- console interface, 2-3

- displaying software levels via LEVEL command, 4-18

- features overview, 1-2

- handling abnormal conditions, 6-7

- introduction to, 1-2

- manual operations, 2-11

- modes of operation, 2-11

- monitoring, 2-3

- normal operations, 2-11

- obtaining status information, 2-6
- CSC (*cont'd*)
 - operations overview, 1-4
 - overview, 1-1
 - reporting problems, 6-17
 - run defaults, quick reference, A-2
 - starting, 2-2
 - stopping, 2-2
 - terminating via EABT command, 4-14
 - terminating with ABORT command, 4-3
 - using, 2-1
 - XQT options, quick reference, A-2
- CSC command reference, 4-1
- CSC failures, 6-15
- CSC monitoring, 2-6
- CSCDUMP
 - starting, 6-19
- CSCLOG
 - starting, 6-19
- CSCUI Common Bank
 - failures and, 6-5
- CULP
 - compatibility with non-CULP systems, 2-10
 - use at boot time, 2-9
- CYCLE command, CSC, 4-8

D

- Data path
 - handling abnormal conditions, 6-7
- DEBUG OFF command, CSC, 4-9
- DEBUG ON command, CSC, 4-10
- DEBUG STATUS command, CSC, 4-11
- Demand terminal security, 2-3
- Diagnostic information
 - collecting, 6-18
- DISMOUNT command, CSC, 4-12
- DISMOUNT request
 - failures and, 6-4
- Dismounting tapes
 - handling abnormal conditions, 6-13
- Displaying
 - network interface status, 2-8
- DN command, CDI, 2-9, 3-3
- DN command, operating system, 2-10
- Documentation
 - related, xiii
 - usage guidelines, xi
- DOWN (DN) command, CSC, 4-13
- DOWNing
 - network interface, CDI DN command, 3-3

E

- EABT command, CSC, 4-14
- EJECT command, CSC, 4-15
- ENTER command, CSC, 4-16
- Error conditions. *see also* Abnormal conditions
 - CSC failure, 6-15
 - dismounting tapes, 6-13
 - handling specific errors, 6-10
 - initialization, 6-10
 - mounting tapes, 6-11
 - server failure, 6-16
 - software failures, 6-15
 - write ring problems, 6-14
- Establishing the cartridge drive environment, 2-9
- Exec messages
 - reference, 5-19

F

- Facility commands
 - overview, 2-10
- Features of CSC, 1-2
- Finding ACS cartridge tapes, 2-13
- FS command, CDI, 2-8, 3-4

H

- Handling abnormal conditions, 6-7
- Handling specific error conditions, 6-10
- HELP command, CDI, 3-6
- HELP command, CSC, 4-17
- Host
 - checking with CDI PING command, 3-8

I

- ID command, CDI, 3-7
- Informational messages
 - reference, 5-3
- Initialization
 - handling abnormal conditions, 6-10
- Interface
 - checking status with CDI FS command, 3-4
 - downing with CDI DN command, 3-3
 - upping with UP command, 3-17
- Introduction to CSC, 1-2

L

- LAN hardware
 - failures and, 6-6
- LEVEL command, CSC, 4-18
- Library locations, 4-2
- Locating ACS cartridge tapes, 2-13
- Locations in a Library, 4-2
- LOCK request
 - failures and, 6-3

M

- Manual operations, 2-11
- MEMORY command, CSC, 4-19
- Messages
 - displaying queued via QUEUES command, 4-25
- Messages reference, 5-1
 - 4780 subsystem, 5-16
 - answerable, 5-2
 - Exec, 5-19
 - informational, 5-3
- Modes of operation, 2-11
- Monitoring CDI, 2-7
- Monitoring CSC, 2-3, 2-6
- Monitoring the control path, 2-6
- MOUNT command, CSC, 4-20
- MOUNT request
 - failures and, 6-3
- MOUNT response
 - failures and, 6-4
- Mounting ACS cartridge tapes, 2-13
- Mounting tapes
 - failures and, 6-2
 - handling abnormal conditions, 6-11
 - physically, 2-13
 - using command processor, 2-12

N

- Network
 - checking status with TCP STATUS command, 3-14
 - control path component failures, 6-5
 - controlling interface state, 2-8
 - removing data structure from a connection, 3-13
 - upping interface with UP command, 3-17

- Network interface
 - checking status with CDI FS command, 3-4
 - displaying status, 2-8
 - downing with CDI DN command, 3-3
- Normal operations
 - overview, 2-11

O

- Operating modes, 2-11
 - manual, 2-11
 - normal, 2-11
- Operational problems
 - solving, 6-1
- Operations overview
 - CSC, 1-4
- OS 2200
 - handling abnormal conditions, 6-7
- Overview
 - CSC - ACS communications, 1-5
 - CSC features, 1-2
 - recovery, 2-14
 - TCP commands, 2-9
- Overview of CSC, 1-1

P

- PATH
 - failures and, 6-5
- Physically mounting ACS cartridge tapes, 2-13
- PING command, CDI, 3-8
- PM command, operating system, 2-10
- Problem analysis
 - useful information, 6-17
- Problem solving, 6-1
 - failures and automated operations, 6-2
 - failures and control path components, 6-5
- Problems, CSC
 - reporting, 6-17

Q

- QUERY command, CSC, 4-21
- QUERY MOUNT request
 - failures and, 6-4
- Queues
 - checking status of CSC, 2-7
- QUEUES command, CSC, 2-7, 4-25

R

- Recovery
 - overview, 2-14
- REPORT command, CSC, 4-26
- Reporting CSC problems, 6-17
- Requests
 - DISMOUNT failure, 6-4
 - failure of ACS, 6-3
 - LOCK failure, 6-3
 - MOUNT failure, 6-3
 - QUERY MOUNT failure, 6-4
 - UNLOCK failure, 6-3
 - VOLUME STATUS CHANGE failure, 6-4
- Responses
 - MOUNT failure, 6-4
- Run defaults
 - CSC quick reference, A-2
- RV command, operating system, 2-10

S

- Security
 - changing levels, 2-4
 - demand terminal, 2-3
- Server failures, 6-16
- Server hardware
 - failures and, 6-6
- Server software
 - failures and, 6-6
- Software failures
 - handling abnormal conditions, 6-15
- Solving operational problems, 6-1
- Starting CSC, 2-2
- Starting CSCDUMP, 6-19
- Starting CSCLOG, 6-19
- Status
 - checking with CDI FS command, 3-4
 - obtaining CDI, 2-8
 - obtaining CSC, 2-6
- STATUS command, CDI, 2-8, 3-9
- STATUS command, CSC, 2-6, 4-28
- Stopping CSC, 2-2
- Subsystem (4780) Messages
 - reference, 5-16
- Syntax
 - notation guidelines, xiii

T

- Tape Automatic Volume Recognition. *see* TAVR
- Tapes
 - dismount problems, 6-13
 - mount failures, 6-2
 - mount problems, 6-11
 - mounting physically, 2-13
 - mounting using command processor, 2-12
 - write ring problems, 6-14
- TAVR
 - overview, 2-10
- TCP ABORT command, 3-10
- TCP commands
 - overview, 2-9
- TCP CONNECTIONS command, 3-11
- TCP HELP command, 3-12
- TCP KILL command, 3-13
- TCP STATUS command, 3-14
- TERM command, CDI, 3-16
- TERM command, CSC, 4-31
- Terminating
 - CSC using EABT command, 4-14
 - network using CDI TERM command, 3-16
- Terminating CSC
 - using CSC ABORT command, 4-3
 - via TERM command, 4-31
- TU command, operating system, 2-10

U

- UNLOCK request
 - failures and, 6-3
- UP command, CDI, 2-8, 3-17
- UP command, CSC, 4-32
- UP command, operating system, 2-10
- Upping
 - network using UP command, 3-17
- Using CSC, 2-1

V

- VENTER command, CSC, 4-33
- VERBOSE command, CSC, 4-34
- Version
 - displaying CDI with ID command, 3-7
- VOLUME STATUS CHANGE request
 - failures and, 6-4

W

Write rings
handling abnormal conditions, 6-14

X

XQT options
CSC quick reference, A-2

EFFECTIVE PAGES

Manual:
Issue Date:
Reissue Date:

CSC Operations Guide 312537201
February 2005

This document has 144 total pages, including:

Title			
Copyrights, Disclaimers & Trademarks			
Contents, Tables & Figures			
Preface	xi	thru	xiv
Chapter 1:	1-1	thru	1-8
Chapter 2:	2-1	thru	2-14
Chapter 3:	3-1	thru	3-18
Chapter 4:	4-1	thru	4-38
Chapter 5:	5-1	thru	5-21
Chapter 6:	6-1	thru	6-20
Appendix:	A-1	thru	A-2
Index:	Index-1	thru	Index-6

Effective Pages
Reader's Comment Form
Business Reply Mailer

