

# StorageTek Automated Cartridge System Library Software Administrator's Guide



Release 8.5  
E96380-08  
March 2022

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Copyright © 2019, 2022, Oracle and/or its affiliates.

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

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

Oracle, Java, and MySQL are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

## Preface

---

Documentation Accessibility	xxxi
-----------------------------	------

## 1 Overview

---

About ACSLS	1-1
The acssa and acsss User IDs	1-1
Logging in Remotely	1-1
Starting and Stopping ACSLS	1-2
The acsss command	1-2
Changing ACSLS Run State	1-2
Using cmd_proc	1-2
Starting cmd_proc	1-2
Suspending and Resuming cmd_proc	1-3
Terminating cmd_proc	1-3
Using cmd_proc in Curses Mode	1-3
Example - Default Curses Mode	1-3
Limited Command History, Limited Length in Curses Mode	1-4
Using cmd_proc in Line Mode	1-4
Status Messages Intermixed with Commands in Line Mode	1-4
cmd_proc Keyboard Shortcuts in cmd_proc (curses and line modes)	1-4
Using cmd_proc in Shell-Friendly Mode (cmd_proc_shell)	1-5
Redirecting Input to cmd_proc - Input Files and Command Scripts	1-6
Redirecting Output from cmd_proc	1-6
ACSLS Directory Structure	1-7

## 2 Starting and Monitoring ACSLS

---

Starting ACSLS	2-1
Monitoring ACSLS	2-1
Stopping ACSLS	2-1
SMF Timeout on Solaris	2-2
ACSLS Startup Policy	2-2

Additional Startup Time	2-2
Desired (offline) Startup State for an ACS	2-3

### 3 ACCLS GUI

---

Starting the ACCLS GUI	3-1
Administering GUI Users and Passwords	3-2
Overview of the GUI	3-2
The GUI Welcome Screen	3-2
Masthead	3-3
Alarms Color Codes	3-4
Navigation Tree	3-5
Overview and Status	3-5
Configuration and Administration	3-6
Tape Library Operations	3-6
Tape Libraries & Drives	3-6
System Operations Excluded in the ACCLS 8.5 GUI	3-7
Firewall and the GUI	3-8
On Solaris	3-8
On Linux	3-8
Configuring a Self-Assigned Digital Certificate for HTTPS	3-9

### 4 Installing and Configuring Your Library Hardware

---

Installing Connections to Your Library Hardware	4-1
Adding the SCSI mchanger Device Driver	4-1
Enabling SL8500 or SL3000 Dual TCP/IP Support	4-2
Synchronizing SL4000 Library and ACCLS Server Clock	4-2
Configuring or Reconfiguring Library Hardware	4-2
Using acsss_config to Configure Your Library Hardware	4-3
Before Running acsss_config for the SL4000	4-4
Running acsss_config	4-5
Renumbering an ACS	4-10

### 5 Setting Variables that Control ACCLS Behavior

---

Accessing the ACCLS Feature Configuration Menu	5-2
Verifying and Changing Dynamic and Static Variables	5-3
Setting CSI Tuning Variables	5-3
Setting Event Logging Variables	5-6
Setting General Product Behavior Variables	5-8
Setting Access Control Variables	5-16

Setting Automatic Backup Variables	5-17
Rebuilding Access Control Information	5-18
Defining Event Notification Settings	5-18
Configuring or Updating your Library Hardware	5-19
Setting OSCI Variables with dv_config	5-19
Registering for Email Notification of System Events	5-19

## 6 Access Control

---

Volume Access Control	6-1
Enabling Volume Access Control	6-1
Associating a Client Identity with a User Name	6-2
Defining other users that are allowed access to the user's volumes	6-2
Establishing Ownership of Your Volumes	6-4
Ownership policies	6-4
Verifying Ownership	6-5
Volume Access Summary	6-5
Command Access Control	6-6
Enabling Command Access Control	6-6
Associating a Client Identity with a User Name	6-7
Defining What Commands are Available to Which Users	6-7
Command Names for Command Access Control allow and disallow Files	6-8
Logging Access Control Messages	6-9

## 7 Library Management

---

Specifying ACS Numbers	7-1
Reconfiguring ACSLS When Tape Drives are Moved, Added, or Removed	7-1
Updating Tape Drives Configured to ACSLS	7-2
Auditing the Library	7-3
How Audit Works	7-3
When to Run an Audit	7-3
Audit Intervals	7-4
Adding a SCSI or fibre-attached library to ACSLS	7-4
Using the Extended Store Feature	7-4
Managing a Mixed-Media Library	7-6
Displaying the ACSLS Mixed-Media Settings	7-7
Mixed-Media Restrictions for SCSI-Attached LSMs	7-7
Setting Scratch Preferences	7-7
User-Defined Mixed-Media Files	7-8
System-Defined Mixed-Media Files	7-8

How ACSLS Uses the Mixed-Media Files	7-8
Defining a Scratch Preferences File	7-9
Managing Dual TCP/IP Connections	7-9
Managing a Dual-LAN Client Configuration	7-10
Registering the IP Addresses of Primary and Secondary LANs	7-10
Installing a Second Ethernet Port for a Multi-Homed ACSLS Server	7-11
Setting the TCP/IP Connection Timeout Interval	7-11
Registering for Event Notification	7-12
Automatic Recovery to Desired State	7-12
Current state and desired state	7-12
Current state cascades down, desired state does not	7-13
Queue and Retry Mounts and Dismounts when Library is Temporarily Unavailable	7-14
Reconfiguring ACSLS When Tape Drives are Moved, Added, or Removed	7-14
Updating Tape Drives Configured to ACSLS	7-15
Media Validation	7-16
Adding Drives to the Media Validation Pool	7-16
Removing Drives from the Media Validation Pool	7-16

## 8 Cartridge Management

---

Populating the LSM	8-1
Using the CAP	8-2
CAP Types	8-2
CAP States	8-2
CAP Modes	8-3
CAP Priorities	8-4
Displaying CAP Information	8-5
Entering Cartridges	8-5
Terminating an Enter Request	8-6
Ejecting Cartridges	8-7
Recovering CAPs	8-7
General CAP Recovery Procedures	8-7
Complete Enters and Ejects Before Doing CAP Recovery	8-7
Recover a Hung CAP by Varying it with Offline Force Then Online	8-7
Recovering a CAP After Opening the Access Door	8-8
CAP Used for Automatic Enter Will Not Unlock	8-8
CAP Used for Manual Enter Will Not Unlock	8-8
CAP Used for Ejects Will Not Unlock	8-9
Automatically Applying Policies to New and Reactivated Cartridges	8-9
Automatic Assignment of Cleaning Cartridge Attributes	8-9
watch_vols Policies	8-10

Cleaning Cartridges	8-10
Automatic Cleaning by ACSLS	8-10
Cleaning Cartridge Maximum Usage	8-11
Manually Defining Cleaning Cartridges	8-11
Monitoring Cleaning Cartridges	8-12
Entering Cleaning Cartridges	8-12
Ejecting Used Cleaning Cartridges	8-13
Manually Cleaning a Drive	8-13
Cleaning Cartridges on Fibre-Attached Libraries	8-14
What To Do When Drives Are Not Being Cleaned	8-14
Verify Auto-Cleaning is Enabled	8-14
Verify there are Cleaning Cartridges for the Drives	8-14
Verify Auto-Cleaning is Disabled Using SL Console for an SL8500 or SL3000 Library	8-15
Check if Cleaning Cartridges are Marked as Questionable	8-16
Managing Scratch Cartridges	8-16
Displaying Scratch Pool and Scratch Cartridge Information	8-17
Adding Scratch Cartridges to the Library	8-17
Rebalancing Scratch Pools	8-17
Deleting Scratch Pools	8-18
Emptying a Scratch Pool	8-18
Deleting a Single Pool	8-18
Deleting All Empty Pools	8-18
Mounting Scratch Cartridges	8-19
Single-Media Environments	8-19
Mixed-Media Environments	8-19
Unscratching Cartridges	8-20
Using Absent and Ejected Cartridge Support	8-20
Absent, Ejected and Missing Cartridges	8-21
Cartridge (Volume) Status Reporting	8-21
Cartridge Recovery	8-22
Missing Cartridges	8-22
Absent and Ejected Cartridge	8-23
Cartridges Not Found	8-23
Cartridges Found	8-23
Using the Manual Volume Delete Utility	8-23
Identifying Expired Cartridges	8-24
Cartridge End-of-Life Percentages	8-24
Access Counts	8-25
ACSLS Cartridge Mount Count Details	8-25
Cartridge Warranty and End-of-life Thresholds	8-26

Moving Least-Recently Accessed Cartridges from Active LSMs	8-26
Identify the LSMs with too Few Free Cells and LSMs with Empty Cells	8-27
Review Cartridges in an LSM by Access Date	8-27
Ensure Access Date is Reported for Easy Sorting	8-27
Look at the Distribution of Last Access Dates for Cartridges in an LSM	8-28
Move the cartridges from Full LSMs to LSMs with Empty Cells	8-28
Create Lists of Cartridges to Move	8-28
Move the Cartridges to an LSM with Free Space	8-29
Manually Loading Cartridges Into Drives in a Disabled LSM	8-29

## 9 Database Administration

---

Utilities Used	9-1
Exporting the Database	9-2
Removing Unsupported Tape Libraries, Drives, and Cartridges Before Exporting to an Earlier Release	9-3
Removing Logical Libraries before Exporting to Linux	9-3
Exporting to a Disk File	9-3
Exporting to Tape	9-4
Importing the Database	9-5
Importing From a Disk File	9-6
Importing from Tape	9-8
Migrating mchangers for Fibre to a New Platform	9-9
Record Details about Fibre-attached Libraries on the Old ACSLS Server	9-9
Reconfigure ACSLS to Change mchanger Names for Fibre-attached Libraries	9-10
Verifying the Imported Database and Library Configuration	9-10
Automatic Database Backup	9-11
Performing Manual Backups to Tape	9-11
Backing up to a Specified Tape Device Attached to the ACSLS Server	9-12
Backing up to a UNIX File	9-12
Recovering and Restoring	9-12
Restoring the Database to the Most Recent Backup	9-13
Recovering from a Failed Server	9-14
Restoring ACSLS Control Files	9-14

## 10 Reporting and Logging

---

Creating a Custom Volume Report	10-1
Custom Volume Report	10-1
Custom Volume Report Example	10-2
Creating a Logging Volume Statistics Report	10-3



# 11 Utility Reference

---

Overview	11-2
Legacy Start/Stop Scripts	11-2
Utility Commands	11-3
acs_partition.sh	11-3
acs_renumber.sh	11-4
acs_rewallet.sh	11-6
The acsss Macro	11-7
Format	11-7
Options	11-8
bdb.acsss	11-8
Format	11-9
Options	11-9
Usage	11-9
Dynamic Configuration (config) utilities	11-10
Limitations of Dynamic Configuration	11-11
Things You Should Not Do	11-11
config acs	11-11
Format	11-12
Adding a new ACS	11-12
Reconfiguring an existing ACS	11-13
Limitation of config acs	11-13
config drives	11-13
Format	11-14
Usage	11-14
config lsm	11-14
Format	11-15
Usage	11-15
Limitation of config lsm	11-15
config ports	11-15
Format	11-16
Usage	11-16
Limitations of config ports	11-16
db_export.sh	11-16
Format	11-17
Options	11-17
Usage	11-17
db_import.sh	11-17
Format	11-18
Options	11-18

Usage	11-18
db_restore.sh	11-18
Format	11-18
Options	11-19
Usage	11-19
del_vol	11-19
Format	11-20
Options	11-20
Usage	11-20
Example	11-21
drives_media.sh	11-21
Format	11-21
Options	11-21
ejecting.sh	11-21
Format	11-22
Options	11-22
ejecting.sh Logs	11-24
free_cells.sh	11-24
Format	11-25
Option	11-25
getHba.sh	11-25
Format	11-25
greplog	11-27
Format	11-27
Options	11-27
Usage	11-27
install_scsi_Linux.sh	11-27
Format	11-28
lib_type.sh	11-28
Format	11-28
moving.sh	11-28
Format	11-29
Creating the Volume List File	11-30
Moving Cartridges Before Removing an LSM from an ACS	11-31
Moving Cartridges Before Removing Cells from a Partition	11-33
Moving Cartridges Before Changing or Removing Panels	11-33
Examples	11-34
probeFibre.sh	11-36
Format	11-36
Options	11-36
rdb.acsss	11-36

Format	11-36
Menu Options	11-37
showDevs.sh	11-39
Format	11-39
Options	11-39
showDrives.sh	11-40
Format	11-40
stats_report	11-40
Format	11-40
Usage	11-41
userAdmin.sh	11-42
Format	11-42
Examples	11-43
volrpt	11-43
Format	11-44
Options	11-44
Usage	11-45
Examples	11-46
watch_vols	11-48
Format	11-48
Usage	11-49
Example	11-49

## 12 Command Reference

---

General Command Syntax	12-2
Component Types and Identifiers	12-2
Common Command Error Messages	12-5
Commands	12-5
audit	12-5
Format	12-6
Usage	12-7
Examples	12-8
Command Area Messages	12-9
cancel	12-11
Format	12-11
Options	12-11
Examples	12-13
Command Area Messages	12-13
Display Area Messages	12-14
clear lock	12-14

Format	12-14
Options	12-14
Usage	12-14
Examples	12-15
Command Area Messages	12-15
Display Area Messages	12-16
define pool	12-16
Format	12-16
Options	12-16
Usage	12-16
Examples	12-17
Command Area Messages	12-17
Display Area Messages	12-17
delete pool	12-18
Format	12-18
Options	12-18
Usage	12-18
Examples	12-18
Command Area Messages	12-19
dismount	12-19
Format	12-19
Options	12-19
Usage	12-20
Dismount	12-20
Dismount force	12-20
Examples	12-21
Command Area Messages	12-21
Display Area Messages	12-23
eject	12-23
Format	12-23
Options	12-23
Usage	12-24
Examples	12-25
Command Area Messages	12-25
Display Area Messages	12-27
enter	12-27
Options	12-27
Usage	12-28
Examples	12-28
Command Area Messages	12-29
Display Area Messages	12-30

idle	12-31
Format	12-31
Options	12-31
Usage	12-31
Examples	12-31
Command Area Messages	12-32
Display Area Messages	12-32
lock	12-32
Format	12-33
Options	12-33
Usage	12-33
Examples	12-33
Command Area Messages	12-34
Display Area Messages	12-35
logoff	12-35
Format	12-35
Options	12-35
Usage	12-35
Examples	12-35
Command Area Messages	12-35
Display Area Messages	12-36
mount	12-36
Format	12-36
Options	12-36
Usage	12-36
Examples	12-37
Command Area Messages	12-37
Display Area Messages	12-38
mount *	12-38
Format	12-38
Options	12-38
Usage	12-39
Examples	12-40
Command Area Messages	12-41
Display Area Messages	12-42
move	12-42
Format	12-42
Options	12-43
Usage	12-43
Examples	12-43
Command Area Messages	12-43

query commands	12-44
Format	12-44
Command Area Messages	12-44
Display Area Messages	12-45
query acs	12-45
Format	12-45
Options	12-45
Usage	12-45
Examples	12-46
query cap	12-47
Format	12-47
Options	12-47
Usage	12-47
Examples	12-48
query clean	12-49
Format	12-49
Options	12-49
Usage	12-49
Examples	12-50
query drive	12-50
Format	12-50
Options	12-50
Usage	12-50
Examples	12-51
query lmu	12-51
Format	12-52
Options	12-52
Usage	12-52
Examples	12-54
query lock	12-54
Format	12-55
Options	12-55
Usage	12-55
Examples	12-56
query lsm	12-56
Format	12-56
Options	12-56
Examples	12-57
query mount	12-57
Format	12-58
Options	12-58

Usage	12-58
Examples	12-59
query mount *	12-59
Format	12-60
Options	12-60
Usage	12-60
Examples	12-61
query pool	12-61
Format	12-61
Options	12-61
Usage	12-62
Examples	12-62
query port	12-63
Format	12-63
Options	12-63
Usage	12-63
Examples	12-63
query request	12-64
Format	12-64
Options	12-64
Usage	12-64
Examples	12-64
query scratch	12-65
Format	12-65
Options	12-65
Usage	12-65
Examples	12-66
query server	12-66
Format	12-66
Options	12-66
Usage	12-66
Examples	12-67
query volume	12-67
Format	12-67
Options	12-67
Usage	12-67
Examples	12-68
set commands	12-69
Format	12-69
Command Area Messages	12-69
Display Area Messages	12-69

set cap mode	12-69
Format	12-69
Options	12-70
Usage	12-70
Examples	12-70
Command Area Messages	12-70
set cap priority	12-71
Format	12-71
Options	12-71
Usage	12-72
Examples	12-72
Command Area Messages	12-72
Display Area Messages	12-73
set clean	12-73
Format	12-73
Options	12-73
Usage	12-73
Examples	12-73
Command Area Messages	12-74
Display Area Messages	12-74
set lock	12-74
Format	12-74
Options	12-74
Usage	12-75
Examples	12-75
Command Area Messages	12-75
Display Area Messages	12-76
set owner	12-76
Format	12-76
Options	12-76
Usage	12-76
Examples	12-76
Command Area Messages	12-76
Display Area Messages	12-76
set scratch	12-76
Format	12-77
Options	12-77
Usage	12-77
Examples	12-77
Command Area Messages	12-78
Display Area Messages	12-78



show	12-78
Format	12-78
Options	12-79
Usage	12-79
Examples	12-79
Command Area Messages	12-79
Display Area Messages	12-80
start	12-80
Format	12-80
Options	12-80
Usage	12-80
Examples	12-80
Command Area Messages	12-80
Display Area Messages	12-81
switch lmu	12-81
Format	12-81
Options	12-81
Usage	12-81
Examples	12-81
Command Area Messages	12-82
Display Area Messages	12-83
unlock	12-83
Format	12-83
Options	12-83
Usage	12-83
Examples	12-84
Command Area Messages	12-84
Display Area Messages	12-85
uptime	12-85
Format	12-85
Usage	12-85
Example	12-85
vary	12-85
Format	12-85
Options	12-86
Usage	12-86
Examples	12-88
Command Area Messages	12-88
Display Area Messages	12-91

## 13 Display Command Reference

---

Using display Command Options	13-2
Using Wildcard Characters	13-2
Examples	13-2
Using Ranges	13-3
Examples	13-3
Format	13-3
Options	13-3
Start cmd_proc in Line Mode when Displaying Lines Longer than 80 Characters	13-4
Commands	13-4
display cap	13-4
Format	13-5
Fields	13-5
Options	13-5
Examples	13-7
display cell	13-7
Format	13-7
Fields	13-7
Options	13-8
Example	13-8
display drive	13-8
Format	13-9
Fields	13-9
Options	13-9
Examples	13-12
display lock	13-13
Format	13-13
Fields	13-13
Options	13-13
Example	13-14
display lsm	13-14
Format	13-14
Fields	13-14
Examples	13-16
display panel	13-17
Format	13-17
Fields	13-17
Options	13-17
Example	13-18
display pool	13-18

Format	13-18
Fields	13-18
Options	13-19
Examples	13-20
display port	13-20
Format	13-20
Fields	13-20
Options	13-20
Example	13-21
display volume	13-21
Format	13-22
Fields	13-22
Options	13-22
Examples	13-25

## 14 lib\_cmd

---

Introduction	14-1
Command	14-1
Pathname	14-1
Synopsis	14-1
Using lib_cmd	14-2
Option	14-3
Subcommands	14-3
Using lib_cmd in Batch Mode	14-8

## A ACSLS Backup and Recovery Tools

---

ACSLS Backup Tools	A-1
Automatic Backups	A-1
Manual Backups	A-2
Manual Database Exports	A-2
ACSLS Recovery Tools	A-3
Using rdb.acsss	A-3
Using db_import.sh	A-3
Disaster Scenarios	A-4
Database becomes corrupted	A-4
Ran acsss_config against the wrong library	A-4
Server failure – rebuilding the same server with new hardware	A-4
Server failure – rebuilding another ACSLS server with new hardware	A-5

## B Enterprise Library Connection Options

---

Overview	B-1
Displaying the status of ACSLS Communication with Libraries	B-2
Dual TCP/IP Support	B-2
Requirements	B-2
Configuration	B-2
Scenario 1 - Preferred Configuration	B-3
Scenario 2	B-4
Scenario 3	B-5
Scenario 4	B-5
Retaining Customized Routing Table Entries after a Reboot	B-6
Creating Scripts	B-6
Adding Custom Route To Be Initialized At Boot Time	B-7
Removing routing commands	B-7
Multi TCP/IP Support	B-8
Redundant Electronics	B-9
ACSLS Support for RE	B-10
Query and Retry of Mounts and Dismounts	B-11
switch lmu for Only a Single Library	B-11

## C ACSLS Support of the SL8500

---

Connecting to Multiple SL8500s Using Multi TCP/IP	C-2
Verifying that all SL8500 Components are Operational	C-2
Understanding SL8500 Internal Addresses and ACSLS Addresses	C-3
Using SL Console to Translate Addresses	C-4
Tape Drive Locations	C-5
Moving Cartridges Before Removing Cells from a Partition	C-6
SL8500 CAPs	C-6
Bulk CAP	C-6
Upgrading the SL8500 to Handle Bulk CAPs with ACSLS	C-7
Custom SL Console Messages Showing Purpose of Enters and Ejects	C-8
Rotational CAP	C-9
Enter or Eject Operations	C-9
Enter, Eject, and Audit Operations for Some ACSLS Clients	C-9
Minimizing Elevator and PTP Activity	C-10
Configuring Tape Drives to Support Your Workloads	C-11
Managing Cartridge Locations	C-11
Finding Missing Cartridges	C-12
Varying the SL8500 Offline	C-12
Use ACSLS to vary SL8500 components offline, not SL Console	C-12

When to vary SL8500 components offline to ACSLS	C-13
Before Opening the Access Door	C-13
If a CAP is inoperative	C-13
When closing the Service Safety Door	C-13
When using the Service Safety Door do not use these ACSLS commands and utilities	C-14
Using the Dynamic Configuration (config) utility	C-14
Expanding the SL8500	C-15
What is Involved	C-15
Diagram of SL8500 Modules:	C-16
Auditing an Expanded SL8500	C-17
Connecting SL8500s with Pass-Thru-Ports	C-19
Before You Install SL8500 PTP Connections	C-20
Adding New SL8500s	C-21
Adding a New SL8500 to the Left	C-21
Dynamically Configuring the New ACSLS Configuration	C-21
Adding a New SL8500 to the Right	C-21
Considerations When Adding an SL8500 to the Right	C-22
Dynamically Configuring the New ACSLS Configuration	C-22
Procedures for Merging ACSs	C-24
Merging ACSs Numbered from Right to Left	C-24
Procedure for Merging ACSs Numbered from Right-to-Left	C-24
Merging ACSs Numbered from Left-to-Right	C-25
Procedure for Merging ACSs Numbered from Left-to-Right	C-25
Removing PTPs and Splitting ACSs	C-26
Splitting ACSs, with the new ACS Created from SL8500(s) on the Left - Possible Scenarios	C-26
ACSLS Procedure for Splitting an ACS	C-27
Adding the New ACS on the Right Hand side of the Split	C-28

## D ACSLS Support of the SL3000

---

ACSLS Support	D-1
Before Configuring ACSLS for the SL3000	D-2
Connecting the SL3000 to ACSLS	D-2
Verifying that all SL3000 Components are Operational	D-2
Configuring SL3000 to ACSLS	D-2
Auditing the SL3000	D-3
SL3000 Addressing	D-3
CAP Numbering	D-4
SL3000 Modules	D-4
New Panel Types	D-5

Understanding Internal SL3000 Addressing	D-5
Using the AEM	D-5
Access Door	D-6
Safety Door	D-6
CAP Operations	D-6
Non-Disruptive Maintenance	D-7
Aliasing SL3000 CAP IDs	D-7
Moving Cartridges Before Removing Cells from a Partition	D-9
SL3000 CAP Behavior	D-9
Finding Missing SL3000 Cartridges	D-9
Varying the SL3000 Offline	D-10
Use ACSLS to vary SL3000 components offline, not SL Console	D-10
When to vary SL3000 components offline to ACSLS	D-10
Before Opening the Access Door	D-10
If a CAP is inoperative	D-11
Using the Dynamic Configuration (config) utility	D-11
Auditing New Cartridge Addresses	D-11
ACSLs Dual TCP/IP	D-12

## E ACSLS Support of the SL150

---

ACSLs Connectivity	E-1
ACSLs and SL150 Library Differences	E-1
Library Configuration	E-1
Identifying Library Locations	E-1
Addressing schemes	E-2
Panels	E-2
Row Numbers	E-2
Column numbers	E-2
Drive addresses	E-2
ACSLs Limitations	E-3
Configuring SL150 Library Settings	E-3
SL150 CAP Behavior	E-4
CAP (Mailslot) Not Opening During an Eject	E-4
Auditing the Library	E-5
When this is a new library	E-5
After adding, removing, or swapping modules	E-6
After Manually Adding or Removing Cartridges Through the Mailslot	E-6
After Changing Library Configuration Settings	E-6

## F ACSLS Support of the SL4000

---

ACSLs Support	F-1
Before Configuring ACSLS for the SL4000	F-2
Configuring the SL4000 to ACSLS	F-2
Changing the SL4000 User Name and Password	F-3
SL4000 Modules	F-3
SL4000 Panel Types	F-4
Understanding SL4000 Addressing	F-4
SL4000 Addressing	F-4
CAP Numbering	F-5
Varying the SL4000 Offline	F-6
Use ACSLS to Vary SL4000 Components Offline, not SL4000 GUI	F-6
When to Vary SL4000 Components Offline to ACSLS	F-6
Before Opening the Access Door	F-6
If a CAP is Inoperative	F-6
Using the Dynamic Configuration (config) utility	F-7
Moving Cartridges Before Removing Cells from a Partition	F-7
Finding Missing SL4000 Cartridges	F-7
Auditing New Cartridge Addresses	F-8
Unknown Cell Information Reported When Auditing the SL4000 Library	F-8
Bulk Operations with the AEM	F-9
Bulk Enters	F-9
Bulk Ejects	F-9
Aliasing SL4000 CAP IDs	F-9
SL4000 OCSI Retention Time	F-10
ACSLs Support for Library Upgrade of SL3000 to SL4000	F-10
Important Considerations	F-10
What About my Volume Information?	F-11
What About my Logical Libraries?	F-11
What About my Client Applications?	F-11
Using watch_vol	F-11

## G Logical Library Support

---

About Logical Libraries	G-1
Benefits	G-1
Limitations	G-2
Creating a Logical Library	G-2
Specifying a Physical ACS	G-3
Specifying the Attributes of the Logical Library	G-3
Assigning One or More Physical Drives	G-4

Assigning One or More Physical Volumes	G-4
Specifying Client Connections	G-5
Fast Load	G-6
Deleting a Logical Library	G-6
Troubleshooting	G-7
What if a client cannot see the logical library that I've assigned?	G-7
What if a client is connected correctly, but still doesn't see the logical library?	G-7
What if a client has trouble configuring to a logical library?	G-7
How to Configure an FC Port in Target Mode	G-7

## H Library Partitioning

---

Aspects Common to Both SL8500 and SL3000 Partitioning	H-1
Guidelines for Partitioning	H-2
Plan Your New Configuration	H-3
Minimize Disruptions when Partitioning an SL8500 or SL3000	H-3
Changes in Configuration	H-3
Keeping Outages to a Minimum	H-3
Challenges	H-3
Partitioning a Library or Changing Partition IDs	H-4
Re-partitioning a Library	H-4
Changing a Partitioned ACS to an Un-partitioned ACS	H-5
Viewing an ACS Partition ID	H-6
CAP Behavior	H-6
Partitioned Library	H-7
Dedicating a CAP in a Partitioned Library	H-7
Sharing CAPs in a Partitioned Library	H-7
When Another Host Needs the CAP	H-7
Assigning CAP Priorities for Shared CAPs	H-8
Reserving the CAP	H-8
Terminating a CAP Reservation	H-8
Moving Cartridges to a Specific Cell	H-9

## I Troubleshooting

---

ACSLs Event Log	I-1
Managing the Event log	I-1
Using greplug to Search Event Logs	I-2
Format	I-2
Options	I-2
Examples	I-2



Additional Logs	I-2
Trace Log Management	I-3
Java Component Logs	I-4
Key Observation Points	I-4
Diagnosing ACSLS Startup Problems	I-4
Testing Library Connections	I-5
testports	I-5
testlmutcp	I-5
testlmu	I-6
pinglmu.sh	I-6
probescsi.sh	I-6
probeFibre.sh	I-6
showDevs.sh	I-6
Troubleshooting an Offline SL4000 ACS	I-7
Testing a Client Connection	I-7
CAP in Fibre Library Attached Through a Bridged Drive is Locked	I-8
Gathering Diagnostic Information for Oracle Support	I-8
ACSLs and Security-Enhanced Linux (SE Linux)	I-8
Un-installing SE Linux Policy Modules for ACSLS	I-8
Managing SE Linux Enforcement	I-9
Verifying the GUI is Operational	I-10
GUI Troubleshooting Tips	I-11
Linux 7.3 and 7.6 Performance Issues	I-12
Missing Host Entry	I-12
SSSD	I-12
Diagnosis 1	I-12
Diagnosis 2	I-13
SL4000 OCSI Retention Time	I-13

## J Issues Installing ACSLS Client Applications

---

Installing an ACSAPI Client on Solaris 11	J-1
---	-----

## K Library Performance

---

Sending Enough Concurrent Mount and Dismount Requests	K-1
Connecting to Multiple SL8500s	K-2
Minimizing Pass-thru Activity Between LSMs	K-2
Floating Cartridges During a Dismount	K-3
Maintaining Empty Storage Cells in an LSM	K-3
Mount and Dismount Timeouts	K-3

Entering Cartridges into a Library	K-4
Ejecting Cartridges from a Library	K-4
Minimizing Pass-Thru Activity Using ACSAPI Requests and ACSLS Commands	K-4
Mounting a Specific Cartridge	K-4
query mount	K-4
mount	K-5
Mounting a Scratch Cartridge Selected by ACSLS	K-5
query mount scratch (also known as query mount *)	K-5
mount scratch (also known as query mount *)	K-5
Using these ACSAPI Requests & ACSLS Commands	K-6

## L Firewall Security Option

---

Running ACSLS Behind a Firewall	L-1
Addressing Security Areas	L-1
RPC	L-1
Security	L-2
Communications Components	L-2
Benefits of the Firewall-Secure Option	L-2
ACSLs Server-Side	L-2
ACSLs Server Port Restriction	L-3
Client-Side (CSC)	L-3
Client Port Restriction	L-3
Advantages	L-4
Turning on Firewall Secure Features and Setting Variables	L-4
ACSLs Variables	L-4
Displaying and Setting ACSLS Variables	L-6
ACSAPI Client System Variables	L-6
New Variables in CSC Toolkit 2.3	L-7
Displaying and Setting Environmental Variables on the Client	L-8
Firewall Secure Solution Scenarios	L-8
Firewall Security on ACSLS Server-Side Only	L-9
Firewall Security on Client-Side Only	L-10
Firewall Security on Both ACSLS Server and Client-Side with Portmapper	L-11
Firewall Security on Both ACSLS Server and Client-Side without Portmapper	L-12
Turning On the Firewall Security on ACSLS Servers	L-13
Turning Off Firewall Security on ACSLS Servers	L-15
Firewall-Secure Configuration	L-16
Troubleshooting Firewall-Secure Communications	L-17
Frequently Asked Questions	L-19

## M CSCI

---

Overview of CSCI	M-1
CSCI Connectivity	M-1
Architecture and Subsystems	M-1
Error Messages	M-2
Common Error Messages	M-2
Environmental Variables	M-3

## N Media Management

---

Overview	N-1
Restrictions	N-1
Examples	N-1
Solution	N-2
Managing Common Media with Recording Densities Reported by ACSLS	N-2
Procedure	N-3
Managing Common Media Recorded in Different Densities with ACSLS Pools	N-4
Procedure	N-4

## O XAPI Client Interface

---

The XAPI Client Interface to an ACSLS Server	O-1
The ACSLS XAPI Service	O-1
XAPI Variables	O-1

## P Accessibility Features in ACSLS

---

User Interfaces	P-1
The ACSLS (text-only) Command-line Interface (CLI)	P-1
Primary Administrative Interface: Unix Shell Functions	P-2
The ACSLS Graphical User Interface (GUI)	P-2
Accessibility Provisions in the ACSLS GUI	P-2
Special Steps Required to Configure Accessibility Modes in ACSLS	P-3
GUI Tree Menu	P-3

## Glossary

---

## Index

---

## List of Figures

---

3-1	ACSL S Graphical User Interface	3-3
3-2	ACSL S Overview and Status Page	3-5
B-1	Preferred Configuration	B-3
B-2	ACSL S and SL8500 or SL3000 Using a Public Network	B-4
B-3	SL8500 or SL3000 with Two Network Interfaces	B-5
B-4	ACSL S HA	B-5
B-5	ACSL S with Multi TCP/IP	B-8
B-6	ACSL S with Multi TCP/IP and Dual TCP/IP	B-9
B-7	ACSL S with RE	B-9
B-8	ACSL S with RE and Multi TCP/IP	B-10
B-9	ACSL S with RE and Dual TCP/IP	B-10
B-10	RE with Dual TCP/IP and Multi TCP/IP	B-11
C-1	SL8500 with an ACSLS Server Before Configuring ACSLS for the SL8500	C-1
C-2	Software and Physical Drive Numbering	C-5
C-3	SL8500 Customer Expansion Module	C-17
C-4	Four connected SL8500 Libraries	C-20
C-5	Auditing Order for Renumbered LSMs	C-23
C-6	ACSs to be Merged	C-24
C-7	Desired Configuration: Single ACS	C-24
C-8	Existing Configuration: Three Separate ACSs	C-25
C-9	Desired Configuration: Single ACS	C-25
C-10	Existing Configuration: One ACS	C-26
C-11	Desired Configuration: Two ACSs	C-27
D-1	SL3000 Addressing	D-3
F-1	SL4000 Addressing	F-5
J-1	Device Configuration Wizard - Drive Path Detection	J-1
J-2	Package Manager Search	J-2
J-3	Package Manager Search Results	J-2
J-4	Device Configuration Wizard - Drive Path and Robot Detection	J-3
K-1	Minimizing Pass-thru Activity Between LSMs	K-2
L-1	Firewall security on ACSLS Server-Side only	L-9
L-2	Firewall Security on Client-Side Only	L-10
L-3	Firewall Security on both ACSLS Server and Client-Side with Portmapper	L-11
L-4	Firewall Security on both ACSLS Server and Client-Side without Portmapper	L-13
M-1	CSCI Architecture and Subsystems	M-1



## List of Tables

---

1-1	cmd_proc Keyboard Shortcuts	1-4
1-2	ACSLs Directory Structure	1-7
6-1	Volume Access is Enabled - ACCESS	6-6
6-2	Volume Access is Enabled - NOACCESS	6-6
6-3	Command Access is Enabled - NOACCESS	6-8
6-4	Command Access is Enabled - ACCESS	6-9
7-1	How the Scope of an Audit Affects the Audit Interval	7-4
7-2	How ACSLS Uses Mixed-Media Files	7-8
8-1	CAP Types	8-2
8-2	CAP States	8-2
8-3	CAP Modes	8-3
8-4	Cap Priorities	8-4
8-5	Cartridge Enter Commands	8-6
8-6	End-of-Life Thresholds	8-26
12-1	ACSLs Component Types and Identifiers	12-3
12-2	Valid Components for Audit	12-6
12-3	Valid Components for Clear Lock	12-14
12-4	Valid Components for Lock	12-33
12-5	query lmu references	12-54
12-6	Valid Lock Types for query lock	12-55
12-7	query lock references	12-56
12-8	Valid Components for Unlock	12-83
12-9	Valid Components for Vary	12-86
12-10	Vary Offline Results	12-86
12-11	Vary Offline Force Results	12-87
12-12	Vary Online Results	12-87
13-1	Drive and Library waiting for Manual Intervention	13-12
C-1	SL8500 Addressing Descriptions	C-3
C-2	Recommendations to Minimize Elevator and PTP Activity	C-10
C-3	Recommendations for Managing Cartridges	C-11
E-1	Example of drive addressing	E-3
I-1	GUI Troubleshooting Tips	I-11
M-1	CSCI Server Subsystem Environment Variables	M-3

# Preface

StorageTek Automated Cartridge System Library Software (ACSL) is Oracle's StorageTek UNIX server software that controls StorageTek automated tape libraries. The StorageTek ACS family of products consists of fully automated, tape cartridge-based data storage and retrieval systems. StorageTek ACSL supports network access to different client systems that can range from workstations to mainframes to supercomputers running on a variety of operating systems.

This guide is for the individual responsible for administering StorageTek ACSL. It is expected that you already have a working knowledge of the following:

- UNIX file and directory structure
- How to use UNIX commands and utilities for your platform
- UNIX system files
- How to do typical UNIX system administrator tasks, such as logging on as root and setting up user accesses to a UNIX application

## Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

### Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

# 1

## Overview

This chapter provides an overview of ACSLS.

### About ACSLS

Automated Cartridge System Library Software (ACSL) is Oracle's StorageTek server software that controls a StorageTek tape library. An Automated Cartridge System (ACS) is a group of tape libraries connected through pass-thru-ports (PTPs). ACSLS accesses and manages information stored in one or more ACSs through command processing across a network. The software includes a system administration component and interfaces to client system applications, and library management facilities.

### The `acssa` and `acsss` User IDs

This section discusses the `acssa` and `acsss` user IDs, which administer ACSLS in support of client applications.

- The `acssa` login provides access to `cmd_proc`, the console user interface for library control operations, and a limited set of ACSLS utilities.

A typical shell environment for `acssa` includes one or more windows running `cmd_proc` and a window that monitors the running tail of the ACSLS event log.

The `acssa` login environment provides access to both of these resources:

```
$ cmd_proc
$ acs_tail $LOG_PATH/acsss_event.log
```

- The `acsss` login provides access to these and all other administrative utilities for general maintenance, configuration, database backup and restore, shell utilities, and general diagnostics.

### Logging in Remotely

Remote access to the ACSLS server is available from any system with an SSH client. An SSH client is a standard feature with any shell on most POSIX compliant operating systems, including Solaris, Linux, and MacOS. For Windows environments, you must install SSH client software such as putty, WinSCP or a similar commercial application.

To access the ACSLS server remotely as user `acssa`, enter the following command:

```
$ ssh acssa@hostname
```

where *hostname* is the host ID of the ACSLS server.

A typical remote environment for `acssa` includes one or more SSH login shells running `cmd_proc` and another shell to monitor the running tail of the ACSLS event log.

```
$ acs_tail $LOG_PATH/acsss_event.log
```



# Starting and Stopping ACSLS

This section introduces the `acsss` command.

## The `acsss` command

The `acsss` command is a start, stop, and status macro that manipulates the various services associated with the ACSLS application. See the following:

- [Starting ACSLS](#)
- [Monitoring ACSLS](#)
- [The `acsss` Macro](#)

## Changing ACSLS Run State

After initial startup and recovery, ACSLS is normally in the run state and ready to handle client activity. You can choose to put it into an idle state, which suspends client request processing. This is a good practice before shutting down ACSLS. You may also wish to hold off client requests temporarily, while updating or re-configuring a library.

To suspend client request processing by putting ACSLS in the idle state, enter the following command from `cmd_proc`:

```
idle
```

The user can continue to issue ACSLS operations using `cmd_proc` in idle state. To resume client request processing after an earlier `idle` command, enter the following command from `cmd_proc`:

```
start
```

ACSLs resumes request processing.

## Using `cmd_proc`

ACSLs must be in run state in order to process any `cmd_proc` commands other than `query server`. If you start a session when ACSLS is not running, you will receive a socket communication error.

## Starting `cmd_proc`

You can start `cmd_proc` from any terminal type that has been defined in `/etc/termcap`. When running in `curses` mode, the terminal must have a display size of 24x80 or larger. By default, an automatic `query server` request is performed by default when `cmd_proc` is started. You can suppress this initial `query server` request by starting `cmd_proc` with the `-q` option.

## Suspending and Resuming `cmd_proc`

You can suspend `cmd_proc` to perform UNIX commands, and then manually resume `cmd_proc`. Any in-process requests that you initiated will continue to completion while `cmd_proc` is suspended.

To suspend and resume `cmd_proc`:

1. While running `cmd_proc`, press `Control+z`.
2. The UNIX shell prompt appears. Perform desired UNIX operations.
3. To resume `cmd_proc`, enter the `fg` UNIX command.

## Terminating `cmd_proc`

To terminate `cmd_proc`:

1. While running `cmd_proc`, wait until all in-process activity is complete and the `ACSSA>` prompt has returned.
2. To exit `cmd_proc`, enter the `logoff` command:

```
logoff
```

The `cmd_proc` session terminates.

## Using `cmd_proc` in Curses Mode

The default mode for `cmd_proc` is a windowed mode using curses. This versatile interface works well with most terminal types and requires a minimum 24-line by 80-character window. The curses interface divides the screen into two sections, where messages bound for `STDERR` are sent to the window upper half and messages bound for `STDOUT` are sent to the bottom half.

When you use the `ACSL` `cmd_proc` command in its default mode, you see general server status messages displayed at the top of the window while your user-specific interactions are displayed at the bottom.

## Example - Default Curses Mode

The following example shows the `cmd_proc` window displayed when you log in as `acssa`. In curses mode, the `cmd_proc` window appears as a split screen. The top section is the message area and the bottom section is the command area, where you enter `ACSL` commands at the prompt. Note that the output display is actually larger, displayed within a 24-line area on the screen.

```
$ cmd_proc
----- Oracle ACSLS x.x.x-----
ACSSA> query server
2019-08-20 11:50:06          Server Status
  Identifier   State   Free Cell  Audit  Mount  Dismount  Enter  Eject
                Count   C/P    C/P    C/P    C/P    C/P
                run    1253   0/0   0/0   0/0   0/0   0/0
```

## Limited Command History, Limited Length in Curses Mode

One disadvantage of curses mode is its limited ability to retain a history of user interactions with the ACSLS server, since the space for those interactions is limited to the bottom half of the 24-line window.

The `cmd_proc` command in curses mode cannot display lines longer than 80 characters, and the `cmd_proc` window will hang if it attempts to display a line longer than 80 characters. If this happens, the `cmd_proc` window can be released with `Control+C` and `Control+D`.

The output from all `query` and other commands is less than 80 characters per line, and the default fields reported for all records by the `display` command require less than 80 characters. However, displaying many optional fields may result in lines that are longer than 80 characters.

## Using `cmd_proc` in Line Mode

To avoid the problems noted with curses mode, you can start `cmd_proc` in line mode. Running in line mode, `cmd_proc` operates within a conventional terminal window that includes a scrollable command history (limited only by the size of the buffer).

You can use `cmd_proc` in line mode by specifying the following option:

```
cmd_proc -l
```

## Status Messages Intermixed with Commands in Line Mode

One disadvantage of line mode operation is the mixture of `STDOUT` and `STDERR` messages that are output to the screen, displayed in the same spot where you may be attempting to enter a new command.

If your `cmd_proc` session is the only session on the system, this may not be a problem. But in a busy production environment where active operations are in progress with ACSLS, it may be difficult to work in a window where status information is being printed on the same line where you are composing an ACSLS request. While it is safe to ignore the system status output on the line where you are typing, you may prefer to redirect that output elsewhere. See [Redirecting Output from `cmd\_proc`](#).

## `cmd_proc` Keyboard Shortcuts in `cmd_proc` (curses and line modes)

The following table describes `cmd_proc` keyboard shortcuts, which are `<CTRL>+` keystroke combinations.

**Table 1-1** `cmd_proc` Keyboard Shortcuts

Key Combination	Action	Notes
<code>Control+c</code>	Cancels the last <code>cmd_proc</code> command.	<code>Control+c</code> is the keyboard shortcut for the <code>cancel</code> command. See <a href="#">cancel</a> for more information about the <code>cancel</code> command.

**Table 1-1 (Cont.) cmd\_proc Keyboard Shortcuts**

Key Combination	Action	Notes
Control+d	Returns to the <code>cmd_proc</code> prompt.	Control+d has no effect if the current command has completed. If the current command is processing, it completes but <code>cmd_proc</code> does not display a response message. If you have not entered the current command at the ACSSS prompt, Control+d deletes the command.
Control+h	Deletes the previous character on the command line.	On most keyboards, you can also use the Enter or backspace key.
Control+i	Refreshes the <code>cmd_proc</code> display.	This function is useful if the current <code>cmd_proc</code> display has been corrupted by noise on the communications lines.
Control+r	Refreshes the current command line.	This function is useful if the current command line display has been corrupted by noise on the communications lines.
Control+r	Deletes the current command line.	N/A
Control+z	Suspends <code>cmd_proc</code> and escapes to the shell environment.	Enter the C shell <code>fg</code> command to resume <code>cmd_proc</code> .

## Using cmd\_proc in Shell-Friendly Mode (`cmd_proc_shell`)

The `cmd_proc_shell` utility can be used to run `cmd_proc` as a conventional shell operation, executing ACSLS commands that are passed as input arguments. It supports execution of a single command or multiple commands.

The utility provides shell-friendly access to `cmd_proc` functions, while retaining all of the benefits and attributes of the shell environment including command history, command editing, output filtering and terminal scrolling. This can be helpful in ad-hoc shell sessions, and when used in shell scripts.

To supply multiple commands as arguments, a new line (either quoted or unquoted, as in the examples) is used to separate the individual commands. Spaces around the new line characters are not necessary, but are included here for readability.

Some examples of `cmd_proc_shell`:

```
cmd_proc_shell query server
cmd_proc_shell query server \n query port all
cmd_proc_shell 'query server \n query port all'
```

To view only offline CAPs:

```
cmd_proc_shell q cap all | grep -v online
```

 **Note:**

- The `cmd_proc_shell` utility does not require a `logoff` command -- a `logoff` is automatically performed.
- The utility does not provide prompts or keyboard shortcuts -- curses mode and line mode do have interactive prompting and command help/retry.

## Redirecting Input to `cmd_proc` - Input Files and Command Scripts

You can use an input file to supply the commands that you want `cmd_proc` to process. For example, the following input file (named `cmd_filename`) verifies ACSLS by mounting and dismounting a cartridge.

```
query drive 0,0,0,0
query volume JB1400
mount JB1400 0,0,0,0
dismount JPB1400 0,0,0,0 force
logoff
```

To start `cmd_proc` with an input file, enter the following command:

```
cmd_proc -q < cmd_filename
```

Alternatively, an input script can invoke `cmd_proc` directly, and supply commands in-line. Make sure that the script file is executable for user `acsss`.

```
ls -l cmd_scriptname
-rwxrwxrwx  1 acsss  acsls           49 Aug 22 13:49 cmd_scriptname

cat cmd_scriptname

cmd_proc -lq << -EOF
query drive 0,0,0,0
query volume JB1400
mount JB1400 0,0,0,0
dismount JPB1400 0,0,0,0 force
logoff
-EOF

$ ./cmd_scriptname
```

## Redirecting Output from `cmd_proc`

You may wish to redirect system messages to some other destination than the screen, to focus solely on the commands and command output. (In curses mode these appear in the upper half, including the copyright.) To direct those messages elsewhere, you can run `cmd_proc` (in various modes) as follows:

```
cmd_proc -l 2> /tmp/SysChatter.out
```

The expression `2>` instructs the shell to redirect `STDERR` to another location. In this example, the status messages are sent to a file in the `/tmp` directory. You can open a second shell window and view a running tail of that file.

You can also perform intended `cmd_proc` operations without viewing system status information at all, by redirecting `STDERR` to `/dev/null`:

```
cmd_proc -l 2> /dev/null
```

You can start `cmd_proc` using an input file, as above, and also redirect the `STDOUT` results to another file. To run a set of commands and capture the results for verification:

```
cmd_proc -q < cmd_filename > result-filename
```

By default, `cmd_proc` display area messages are written to `STDERR`. However, you can redirect these messages as well:

```
cmd_proc -q < cmd_filename > result-filename 2>> result-filename
```

## ACSLs Directory Structure

The following table shows a listing of the directories, subdirectories, and most commonly used files and shell scripts in the ACSLS directory structure.

Three variables are used for ACSLS paths. They are:

- `$installDir`  
This is the base installation directory and is `/export/home/` by default.
- `$ACS_HOME`  
Located at `$installDir/ACSSS/`, this is the home directory for the `acsss` user ID and where the ACSLS product is installed.  
`$ACS_HOME` is `/export/home/ACSSS` by default.
- `$ACSDB_BKUP`  
This is the directory where the ACSLS backups are saved.

**Table 1-2 ACSLS Directory Structure**

Directory	Contents
<code>\$installDir</code> (by default <code>/export/home/</code> )	The base installation directory.
<code>\$installDir/SSLM</code>	Home of ACSLS java components including the ACSLS GUI and the SMCE (logical library operation)
<code>\$installDir/SSLM/AcslsDomain</code>	The home directory of the ACSLS Web-based GUI application.
<code>\$installDir/wlinstall</code>	The bundled WebLogic application server package and related installation scripts.
<code>\$installDir/Oracle</code>	The un-bundled WebLogic home directory.
<code>\$installDir/acsls_thirdPartySoftware</code>	A collection of third-party license information and related republished source code.
<code>\$ACS_HOME</code> ( <code>\$installDir/ACSSS</code> ) (by default <code>/export/home/ACSSS/</code> )	Home directory for the <code>acsss</code> user ID. Also the ACSLS home directory. ( <code>ACS_HOME</code> environment variable points to this directory.)
<code>\$ACSDB_BKUP</code> (by default <code>/export/backup/</code> )	Database backups

**Table 1-2 (Cont.) ACSLS Directory Structure**

<b>Directory</b>	<b>Contents</b>
<code>\$ACS_HOME/config/</code>	Contains ACSLS configuration files.
<code>\$ACS_HOME/data/external/</code>	Contains customized files used in access control, mixed media, and cartridge reporting.
<code>\$ACS_HOME/data/external/access_control/</code>	Contains access control sample and customized files.
<code>\$ACS_HOME/data/internal/</code>	ACSLS internal configuration files. do not modify.
<code>\$ACS_HOME/diag/bin</code>	Contains diagnostic files and shell scripts.
<code>\$ACS_HOME/lib/</code>	Contains ACSLS installed shared libraries required at run-time.
<code>\$ACS_HOME/log/</code>	Contains ACSLS event log and utility event log files.
<code>\$ACS_HOME (\$installDir/ACSSA/)</code> (by default <code>/export/home/ACSSA/</code> )	acssa home directory.
<code>\$installDir/ascdb/</code> (by default <code>/export/home/acscdb/</code> )	Database home directory.
<code>\$LOG_PATH</code>	This is equivalent to <code>\$ACS_HOME/log</code> . This directory contains the <code>acsss_event.log</code> and other useful logs that pertain to ACSLS operation

# 2

## Starting and Monitoring ACSLS

Once ACSLS has been installed and configured with the attached library, the application can be enabled with the command, `acsss enable`. The `acsss` macro manipulates the multiple services that are associated with ACSLS, bringing them up and shutting them down in the proper order, and providing a high-level view of the overall system status.

Depending on the installation, an ACSLS application is an aggregate consisting of up to seven services that are installed on the Solaris or Linux system:

- `acsdb` - maintains the ACSLS library database.
- `acsls` - library control software that executes library operations.
- `weblogic` - web server for the ACSLS GUI.
- `surrogate` - communication link between java services and `acsls`.
- `rmi-registry` - lookup service for named java objects and methods.
- `smce` - SCSI Media Changer Emulation of logical libraries.
- `stmf` - target mode framework for logical libraries.

The first two services are common to all installations. The `weblogic`, `surrogate`, and `rmi-registry` services are present where the ACSLS GUI has been installed. The `smce` and `stmf` service is seen on Solaris systems where logical library support has been configured. All of these services are manipulated by the ACSLS user with the single macro, `acsss`.

### Starting ACSLS

As root, start ACSLS by running:

```
acsss enable
```

This command is the default method for bringing up ACSLS. It checks for dependencies and activates, in the proper order, the various ACSLS services and the ACSLS GUI. The services are configured to start automatically after system reboot.

### Monitoring ACSLS

For a quick status report of the various ACSLS services, run the command:

```
acsss status
```

### Stopping ACSLS

Stopping ACSLS is not a complete shutdown and enables the database and any GUI login sessions to remain active for maintenance operations after the `acsls` and `smce` services have been disabled. Use this procedure to shut down ACSLS and the database.

To stop the ACSLS, use the command:



```
acsss disable
```

## SMF Timeout on Solaris

The Solaris SMF utility allots a set amount of time for each service to become fully enabled. For the `acsls` service, this time limit is calculated on the basis of the library configuration: the number of LSMs, the number of drives, and the number of CAPs. A large library configuration takes longer for ACSLS to recover than a smaller configuration, so a longer SMF timeout period is allotted to a larger configuration.

On rare occasions, a faulty LSM may take more time to come up than the SMF time limit allows. When the time-out period has expired, SMF will restart the operation. This action has the potential for the start sequence to go into an endless loop, preventing ACSLS from ever recovering during difficult startup conditions.

A special file, `acsls_startup_policy`, is intended for use in such situations. This file, located in the `$ACS_HOME/data/external` directory, when configured adds extra time for startup recovery, or to exempt any specific ACS from being recovered during the SMF startup sequence. Detailed configuration instructions are contained in the header remarks of `acsls_startup_policy`. By adjusting the start-up parameters in this file, you can avoid ACSLS startup problems introduced by an abnormal library startup condition.

For more information, see [Diagnosing ACSLS Startup Problems](#).

## ACSLs Startup Policy

This file alters the normal startup parameters that apply when starting ACSLS. Changing the default startup values is not recommended without careful analysis and consultation with Oracle ACSLS Software Support.

## Additional Startup Time

This parameter applies to the SMF startup timeout for the `acsls` service on Solaris. The `acsls` startup timeout is calculated automatically by the current library configuration. A longer timeout is given to libraries that have more LSMs, more drives and more CAPs. This timeout will adjust automatically as the library configuration changes. you can view the calculated value by asserting the command:

```
acsss timeout
```

If the automatic calculated timeout is not sufficient, the SMF facility may intervene to restart the `acsls` service before sufficient time had elapsed to allow the previous startup sequence to finish.

Granting more time to the startup sequence can prevent such SMF intervention, but not without compromises. Adding too much time can mask troublesome aspects of the configuration that may require attention. Extending the normal timeout period delays SMFs' ability to alert an operator of any serious or irrecoverable startup problems.

To grant additional minutes for the `acsls` start sequence to complete, place integer value after the '=' in the following line:

```
additional_startup_time=0 # Minutes
```

## Desired (offline) Startup State for an ACS

When ACSLS comes up, it will bring all of the library resources to the last established desired state. If the desired state is online, the process of bringing the ACS online involves a recovery period in which the physical library resources of the given ACS are checked and verified against the database image of the configuration. This process transpires within a span from less than a minute to several minutes, depending on the size of the library configuration and on the existence of any unusual circumstances.

You can bypass this recovery time for any ACS by placing the desired state of that ACS and its associated ports offline. While such action speeds the online status of the `acsls` SMF service, subsequent manual action is necessary to vary the actual ACS and its port(s) online.

To set the desired startup state of an ACS and its ports to offline, remove the comment character (#) from the beginning of the appropriate line in the `acsls_startup_policy` file in the `$ACS_HOME/data/external/` directory.

For example, change:

```
# ACS0_desired_startup_state_is_offline
```

to:

```
ACS0_desired_startup_state_is_offline
```

# 3

## ACSLs GUI

Introduced in ACSLS 8.0, the Web-based ACSLS Graphical User Interface (GUI) provides a graphical console for manipulating and monitoring library operations and for the administration and operation of logical libraries. Refer to the online help for procedures and detailed information for using the GUI.

The GUI provides:

- An alternative library console with greatly expanded flexibility and ease of use. It provides most of the operations that are available in the legacy `cmd_proc` (exceptions are shown under [System Operations Excluded in the ACSLS 8.5 GUI](#)).

- Logical library support.

For more information about logical libraries, refer to [About Logical Libraries](#).

- For library administration and operation. It provides the ability to perform most legacy `cmd_proc` operations, along with new operations related to logical library management.
- Real-time monitoring of tape library components.
- A tree browser to navigate physical and logical configuration.
- Real time alerts that are visible from each screen.

An alert informs you of conditions that might result in damage to hardware, corruption of data, or corruption of application software. It always precedes the information to which it pertains.

- The ability to filter volume and drive displays with user-specified criteria.
- User-customized views of system events and system logs.
- Online help.

To create, manage, or delete logical libraries, you must use the ACSLS Graphical User Interface (GUI). The GUI is automatically included in the ACSLS 8.5 installation package. For more information on logical libraries, refer to [Logical Library Support](#).

## Starting the ACSLS GUI

The software that controls the ACSLS GUI starts automatically when the ACSLS services are enabled. The ACSLS GUI is deployed among other applications inside WebLogic on the Solaris platform.

To log in to the ACSLS GUI:

1. Open a browser and submit a URL with the server hostname or IP address:

```
https://myAcslsHostName.myDomainName:7002/SlimGUI/faces/Slim.jsp  
or
```

```
http://127.99.99.99:7001/SlimGUI/faces/Slim.jsp
```

It is best to use the fully-qualified host name or the IP address of the host system. Some pages, including the ACSLS help pages, may not display properly if the URL cannot be fully resolved by WebLogic.

If you use http with port 7001, WebLogic automatically re-routes you to https on port 7002.

Since WebLogic is using the secure https protocol, your browser may warn you the site security certificate has not been registered, and therefore is un-trusted. If you are confident that the URL is your local ACSLS server, you are safe to proceed. At this point, you should see the login screen. If your browser requires a security certificate, see [Configuring a Self-Assigned Digital Certificate for HTTPS](#).

2. On the login screen, enter a valid ACSLS user ID and the corresponding password. These are the user IDs that you set up when installing WebLogic or through `userAdmin.sh` (such as `acsls_admin`).
3. After successful login, you should see the ACSLS GUI Welcome page.

## Administering GUI Users and Passwords

Use the `userAdmin.sh` menu-driven utility to add and manage ACSLS GUI users and passwords. See [userAdmin.sh](#).

## Overview of the GUI

The following sections cover a basic overview of the GUI.

### The GUI Welcome Screen

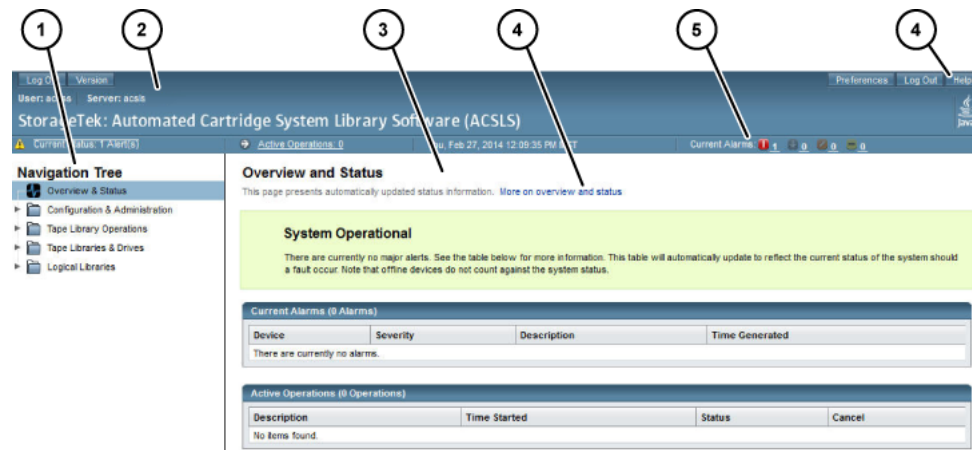
The ACSLS GUI Welcome screen displays three windows:

- The masthead in the top frame
- The navigation tree in the left-hand frame
- The Overview and Status page in the right-hand frame

After an initial load of the GUI into your browser, if the right-hand frame does not display, refresh your browser.

The following figure shows the ACSLS GUI.

Figure 3-1 ACSLS Graphical User Interface



Legend:

1. Navigation tree
2. Masthead
3. Page
4. Help links
5. Alarms

## Masthead

The masthead comprises the following buttons:

- **Log Out**  
The Log Out button logs you out of the GUI.  
For added security, it is recommended that you clear your browsing history every time you log out of the GUI.
- **Version**  
The Version button displays the current ACSLS version and copyright notice.
- **Preferences**  
The Preferences button opens up a page so you can specify GUI system preferences.
- **Logout**  
The Logout button ends the current session.
- **Help**  
The Help button launches the online help system
- **Active Operations**  
The Active Operations button shows the number of ACSLS operations in progress. This is a hyperlink to the Overview and Status page which reveals details about alarms and active operations.

- **Current Status**  
The Current Status button displays the number of active alerts. This is a hyperlink that takes you to the Overview and Status page.
- **Date and Time of Login**  
The Date and Time of Login is the time stamp showing the original login time of the current GUI session.
- **Current Alarms**  
The Current Alarms button is a dashboard view showing the number and severity of any current system alarms. The details of any current alarm is summarized on the Overview and Status page.

## Alarms Color Codes

The following table describes the alarms color codes:

Color	Severity Level	Device Affected
Red	Critical	This signifies a condition that seriously inhibits normal ACSLS operation. A critical alarm displays when the underlying problem requires immediate attention. The resulting impact leads to a system-wide halt to normal ACSLS library operations.
Black	Down	This alarm indicates that one or more subsystems within ACSLS is offline. The offline status may be due to intentional manual operation or may be due to other factors. Usually the presence of associated technical factors will be indicated by the yellow (minor) alarm or the orange (major) alarm icon.
Orange	Major	A major alarm indicates a perceived malfunction with the software or with any of the underlying components that enable ACSLS or the GUI to function.
Yellow	Minor	This color implies that the system is not fully functional. The system may be initializing, or there may be one or more discrete subsystems that are not operational. This interruption is viewed as temporary, or when the function it represents is discrete and not vital to the overall library control operation.

The masthead alarms are not illuminated in color unless there is at least one active alarm in that category. The actual number of alarms is displayed in numeric text next to the color icon. If there are no active alarms, the number shows zero (0).

For color-blind users, there is a "tool-tip" feature that allows you to hover over each color icon with the mouse. A small pop-up window displays the meaning of that graphical icon in plain text.

## System Preferences

This page is accessed from the Preferences button in the Masthead. The page provides numerous drop-down menus that allow you to alter system behavior depending on your personal preferences:

- **Default Tree Menu** - Specify an expanded or collapsed menu format.
- **Log Page Size** - Specify the number of events to be displayed on pages in the Log Viewer.

- Alert Update Interval - Specify how often the browser should probe the server for system alerts. If your GUI response time is slow, you can increase this interval.
- Number of Days to Retain Events - Specify how much history should be displayed on the System Events page.
- Enable Fast Load for SCSI Clients - Specify (yes) to return a successful response to the SCSI client immediately after a valid mount request has been received, or (no) to wait until the robotic mount operation has actually completed.

## Navigation Tree

The Navigation Tree will always be located in the left-hand frame of the ACSLS GUI. The collapsed tree offers navigational links to:

- The Overview and Status page
- Configuration and Administration pages
- Tape Library Operations pages
- Tape Libraries and Drives pages
- Logical Libraries pages

## Overview and Status

This page is the first place to go when you want to identify the specific malfunction or interruption of service indicated by the alarm icons in the Masthead. At the top of this page is a status summary that reflects the severity of the overall system condition.

**Figure 3-2 ACSLS Overview and Status Page**

**Overview and Status**  
This page presents automatically updated status information. [More on overview and status](#)

**System Operational**

There are currently no major alerts. See the table below for more information. This table will automatically update to reflect the current status of the system should a fault occur. Note that offline devices do not count against the system status.

Current Alarms (0 Alarms)			
Device	Severity	Description	Time Generated
There are currently no alarms.			

Active Operations (0 Operations)			
Description	Time Started	Status	Cancel
No items found.			

Two boxes display under the status summary. They are:

- Current alarms

Current alarms show the device affected, the severity level, a description, and the time the event took place. The severity levels, are:

- System Critical is shown with a red icon and red background color. This signifies a condition that seriously inhibits normal ACSLS operation.
- System Degraded is shown with a yellow icon and yellow background color. This implies that the system is not fully functional. The system may be initializing, or there may be one or more discrete subsystems that are not operational.

- System Operational is shown with a green background color. This indicates that all subsystems are functional and no errors have been detected with system components.
  - Active alarms
- Active operations table lists library operations that are currently in progress. Each operation is listed with a brief description, the time the action started, the current status, and a radio-button option to cancel the operation.

## Configuration and Administration

The Configuration and Administration section contains links to pages that enable you to monitor the ACSLS system and other pages that allow you to manage logical library configurations. When you expand this leaf of the tree, you get the following submenu:

- Log Viewer
- The default Log Viewer page displays a running tail of the various logs that monitor system events, including the `acs_ss_event.log`, the `smce_trace.log`. This page provides navigational buttons that enable you to scroll through the history of the log back to its beginning.
- System Events
- Every discrete library operation is recorded in the System Events log. Each record in this log contains an event time stamp, an event type, and a description of the event.
- Logical Library Configuration
- When you expand this leaf of the tree, it opens into various options for configuring and managing logical libraries:
- Create a logical library
  - Edit a logical library
  - Assign volumes to a logical library
  - Un-assign volumes from a logical library
  - Manage client connections.
- For more information on logical libraries, refer to [Logical Library Support](#).

## Tape Library Operations

Tape operations enable you to:

- Audit the ACS, LSM, or panel
- Mount volumes
- Dismount volumes
- Enter volumes
- Eject volumes

## Tape Libraries & Drives

The following detailed information is provided for tape libraries and drives:



- Library operations that let you stop or start processing
- Capability to view:
  - physical state of the library server
  - current activity that graphically shows mounts in the past hour and mounts in the past 24 hours
  - physical components which includes:
    - total number of ACSs
    - capacity
    - total volumes
    - total volumes by type
    - total free cells
    - graphical view of the cell usage
    - total LSMs
    - total LSMs by type
    - total drives
    - total drives by type
    - total number of CAPs
    - list of logical libraries

## System Operations Excluded in the ACSLS 8.5 GUI

Most installation utilities, diagnostics, and other shell-command utilities remain outside the scope of the ACSLS GUI console. There are a few operations in the legacy `cmd_proc` that have not been implemented in the ACSLS GUI for Release 8.5. These include:

- Define, delete, query, or display pools
- Display panels
- Idle or start the server
- Lock or unlock volumes
- Lock or unlock drives
- Mount scratch volumes
- Query or switch LMUs
- Perform a query mount
- Set scratch volumes
- Set or clear locks
- vary, query, or display ports
- venter

## Firewall and the GUI

If firewall protection is enabled on your ACSLS server, it will impact remote user access to the ACSLS GUI unless an explicit policy is configured with the firewall software.

### On Solaris

Consult the man pages for `ipf` and `ipfilter`. The `ipfilter` firewall is enabled (disabled) by `root` using the following command:

```
svcadm enable ipfilter (svcadm disable ipfilter)
```

- To learn the current status of `ipfilter`:

```
svcs ipfilter
```

Firewall policies can be defined in the file, `/etc/ipf/ipf.conf`. To allow free communication between components on the local host (such as, between ACSLS and WebLogic), include a statement, like the example below:

```
pass in quick from 127.0.0.1 to 127.0.0.1, or  
pass in quick from 127.0.0.1 to all
```

- To provide a policy allowing remote Web-based browsers to access the ACSLS GUI, you must open up ports `7001` and `7002`.

```
pass in quick from any to any port = 7001  
pass in quick from any to any port = 7002
```

For ACSLS ACSAPI clients, you must discover which ports are in use by ACSLS. Use the following command from your UNIX shell:

```
rpcinfo -p | egrep "300031 | 536871166"
```

The port IDs are listed in the last field of the display. You must add 'pass in quick' statements for each of these. It is also necessary to include a 'pass in quick' statement for the RPC portmapper port, `111`.

For further firewall information for ACSLS, see [Firewall Security Option](#).

The last statement in your proposed rule set, `block in from any`, assumes that no traffic should reach the host unless specifically allowed in previous statements.

### On Linux

Consult the man pages for `iptables`. The `iptables` firewall is enabled (or disabled) by user `root`, using the following command:

```
service iptables start (service iptables stop)
```

- To check the status of `iptables`:

```
service iptables status
```

The policy file for `iptables` is `/etc/sysconfig/iptables`. To include a policy that allows remote `http/https` access to the ACSLS GUI, you should update that file to include exceptions for ports `7001` and `7002`, using statements like the example below:

```
-A input -p tcp --dport 7001 -j ACCEPT  
-A input -p tcp --dport 7002 -j ACCEPT
```

For more information on iptables, see [Verifying the GUI is Operational](#).

For ACSLS ACSAPI clients, you must discover which ports are in use by ACSLS. Use the following command from your Linux shell:

```
rpcinfo -p | egrep "300031 | 536871166"
```

The port IDs are listed in the last field of the display. You must add exceptions for each of these ports in your iptables policy. It is also necessary to provide an exception statement for the RPC portmapper port, 111.

For further firewall information for ACSLS, see [Firewall Security Option](#).

## Configuring a Self-Assigned Digital Certificate for HTTPS

When you install WebLogic on your ACSLS server, a simple 512-bit public key is automatically available to support basic https exchanges with client browsers. Normally, no further configuration should be necessary. However, some browsers, notably the Microsoft Internet Explorer, require a lengthier key of no less than 1024 bits.

Internet Explorer and FireFox Version 39 and above require a WebLogic set-up procedure for use with https servers that do not employ certificates verified by a third-party digital signing authority.

Refer to "Configuring a Self-Assigned Digital Certificate for HTTPS" in the *ACSLs 8.5 Installation Guide* for procedures.

# 4

## Installing and Configuring Your Library Hardware

The installation and configuration of your library hardware involve the following tasks:

- [Installing Connections to Your Library Hardware](#)  
Installing connections to library hardware involves:
  - Adding the SCSI mchanger device driver
  - Optionally enabling Dual TCP/IP support (for the SL8500 or SL3000)
- [Using `acsst\_config` to Configure Your Library Hardware](#)  
This lets you:
  - Establish library communications
  - Configure the number of libraries to be supported and if they are partitioned
  - Establish if the connection to a library is SCSI/Fibre, TCP/IP, serial or SCSI/Fibre and the format to use
  - Reconfigure your library hardware

### Note:

Refer to the *ACSLs Product Information Guide* for the current list of supported libraries, drive types, media types, and the drive-to-media compatibility.

## Installing Connections to Your Library Hardware

This section discusses installing connections to your library hardware.

### Adding the SCSI mchanger Device Driver

The SCSI Media Changer (mchanger) is the device driver that communicates between ACSLS and any SCSI or fibre-attached library on Solaris operating systems. On Linux, ACSLS uses the native `sg` driver.

For both Solaris and Linux, ACSLS also creates the `/dev/mchanger` links which you specify when configuring your library hardware to ACSLS.

Adding support for SCSI libraries is an optional part of the installation process. However, there may be times when you must add a new SCSI library, or simply to add SCSI support, to an existing ACSLS installation. You can install the necessary drivers and create mchanger links at any time.

1. Login to the ACSLS server as `root` user and enter your password.
2. Go to the installation directory:

```
cd $ACS_HOME/install
```

3. Ensure that the SCSI library is functional and physically attached to the ACSLS server.

 **Note:**

(Solaris) Unless multi-pathing hardware has been deliberately configured for the SL500 library, it is important to disable multi-path I/O for any parent device driver such as the "fp" driver that is used in the library connection path. If you have difficulty making a successful connection to the SL500, you should confirm that multi-path I/O has been disabled in your <driver>.conf file, typically /kernel/drv/fp.conf.

```
mpxio-disable="yes"
```

4. Invoke the appropriate driver installation script:

- Solaris

```
./install_scsi_sol.sh
```

- Linux

```
./install_scsi_Linux.sh
```

## Enabling SL8500 or SL3000 Dual TCP/IP Support

If you have installed the SL8500 or SL3000 library and want to enable dual TCP/IP support, see [Dual TCP/IP Support](#) . These sections also discuss the customized routing table entries that you must create.

## Synchronizing SL4000 Library and ACSLS Server Clock

Ensure that the SL4000 library clock is synced within a couple minutes of the ACSLS server clock. This is important for communication functionality and filtering of messages related to ACSLS operations and the SL4000 library. Use the Linux `date` command and the SL4000 GUI to review and sync clocks between them. Alternatively, you can configure NTP functionality for both the ACSLS server and the library. Refer to the *SL4000 Library Guide* and your operating system publications for more information.

## Configuring or Reconfiguring Library Hardware

There are two ways to configure or reconfigure library hardware (new or changed libraries and/or tape drives):

- `acsss_config`

This command must be run while ACSLS is down. Use `acsss_config`:

- For your initial (first time) configuration of library hardware
- To remove ACSs
- To change or remove port connections to a library

- To reconfigure Fibre or SCSI-attached libraries
- Dynamic Configuration (`config`) utility
 

Run this utility while ACSLS remains running. Use `config` for all changes except those excluded above. The `config` utility:

  - Adds new libraries
  - Updates TCP/IP-attached library configurations
  - Adds port connections, LSMs, CAPs, and drives

## Using `acsss_config` to Configure Your Library Hardware

Using `acsss_config` enables configuration or re-configuration of library hardware:

- Each ACS must have at least one CAP. This can be a CAP that is shared with another partition.
- There must be at least one drive configured to the entire ACSLS system.
 

For example, if ACSLS supports four libraries; three of the libraries can contain zero (0) drives. However, the fourth library must contain at least one drive.
- During the execution of `acsss_config`, any ACS requires a username and password to communicate with the library. Contact your library administrator for these credentials. You must provide user name and password credentials. Otherwise, as `acsss_config` attempts to continue processing, library communication failure errors will occur and the utility will be forced to exit. Note that a single point of failure during `acsss_config` will require you to reconfigure the ACS, and any ACSs that may have been correctly configured.
- Configure your library hardware

You must use the `acsss_config` utility to configure your library hardware for the first time. The `acsss_config` utility is menu driven and selecting option 8 lets you to perform tasks such as configuring hardware for the first time, deleting libraries, or changing port connections to libraries.

### Note:

ACSLs lets you skip ACS numbers when you configure or reconfigure libraries (see option 6). Both `acsss_config` and Dynamic Configuration support specifying and skipping ACS numbers.

- Reconfigure your library hardware
 

Before using `acsss_config` to update the configuration information for all of your libraries, record the current configuration. Changing the ACS numbers assigned to your current libraries would cause all of your volumes to be marked absent and all of your drive's addresses to change. Save the output from a `query lmu all` command to record your current ACS numbers and port connections.

 **Note:**

If `acsss_config` removes an existing physical ACS from the configuration while logical libraries with client connections still exist for that ACS, the SMCE subsystem may come up in maintenance state. To avoid this, before using `acsss_config` to remove an ACS from the library configuration, any associated logical libraries must be deleted using the ACSLS GUI (or the `lib_cmd` CLI in ACSLS 8.2 and later releases). Deleting the associated libraries first ensures that all related information is correctly removed.

After your initial library configuration, you can use the `config` utility, to dynamically add or reconfigure ACSs, LSMs, or tape drives without stopping ACSLS.

For more information and procedures, see [The `acsss` Macro](#) .

 **Note:**

The ACSLS database must be updated using `acsss_config` or `config` whenever the physical configuration of the library changes. ACSLS will not run correctly if the configuration defined in its database does not match the configuration defined in the library.

**Example:**

Configuration changes requiring an update to the database include:

- Adding or removing an ACS, LSM (including a SCSI-attached LSM such as the SL500), Pass-Thru Port (PTP), transport, or standby LMU

 **Caution:**

After you install a new ACS or LSM, ensure that each attached library and LSM is fully (hardware) configured, powered on, and ready before you reconfigure ACSLS. Otherwise, `acsss_config` or `config` will not configure the library correctly.

- Adding or removing a port connection between the server system and the LMU

## Before Running `acsss_config` for the SL4000

Before running `acsss_config`, ensure that you have completed the following library configuration tasks using the SL4000 GUI:

- Define an SL4000 library certificate, including the **Library Name (CN)**.

The SL4000 library requires a properly generated certificate for authentication purposes before it can connect to ACSLS. If the library certificate is not properly generated, ACSLS will return errors when it attempts to connect to the library through `acsss_config`.

From the SL4000 GUI, click **Configuration** in the left navigation area and then select the **Certificates** tab.

When you define the new certificate or view an existing certificate, note the **Library Name (CN)** as it is required during the `acsss_config` process when you are prompted for the device or host.

This CN name must match that used in `acsss_config` and `config new acs`. If using a host name (DN), not an IP address, it must also resolve to the same exact name. This may require DNS and/or `resolv.conf` modifications.

 **Note:**

If you change a previously defined SSL certificate for an SL4000 library, the SL4000 library must be rebooted for the new certificate to take effect.

- Define an SL4000 user that the ACSLS SCI interface can use to connect to the SL4000 library.

 **Note:**

ACSL S SCI connection to an SL4000 library requires an SL4000 user credential with a user role at the User level. The SL4000 Administrator role can also be used for this credential.

From the SL4000 GUI, click **Users** in the left navigation area.

Assign the User role to the ACSLS SCI interface user and specify a user password. Note the user name and password as they are required during the `acsss_config` process.

- Ensure that the SL4000 library is SCI capable, or has an SCI capable partition.

From the SL4000 GUI, click **Partitioning** in the left navigation area to display the partitioning table.

- If you are not partitioning the library, ensure that the **Default** partition row does not specify `yes` in the SCSI column. Also, verify that partitions are only displayed in the **Default** and **Media Validation** rows. This signifies a non-partitioned SL4000.
- If you are partitioning the library, the SL4000 partition will have a partition ID of 21 or higher. Ensure that the ACSLS partition row does not specify `yes` in the **SCSI** column. Note the partition ID as it is required during the `acsss_config` process.

Refer to the "Configuring the Library" section in the *SL4000 Modular Library System Library Guide* for more information about using the SL4000 GUI to perform these tasks.

## Running `acsss_config`

To run `acsss_config`:

1. Log in as `acsss`.
2. If you are re-configuring your library hardware, enter the `query lmu all cmd_proc` command and record the following:
  - ACS numbers



- Partition IDs
  - Port connections
3. Shut down ACSLS if it is running.
  4. Run the configuration utility:

```
acsss_config
```

The ACSLS feature configuration screen appears.

 **Note:**

The configuration utility, `acsss_config`, configures the library correctly unless all libraries, LSMs, and transports are not fully configured, powered on, and ready.

The following menu appears:

```
ACSLS feature configuration
Please enter the number followed by Return for your choice from the
following menu to configure product behavior in that area.
Press? followed by the Return key for help.
1: Set CSI tuning variables
2: Set event logging variables
3: Set general product behavior variables
4: Set access control variables
5: Set automatic backup parameters
6: Rebuild Access Control information
7: Event Notification settings
8: Define or Change Library Hardware Configuration
E: Exit
Menu choice:
```

 **Note:**

For a discussion of options 1-7, refer to [Setting Variables that Control ACSLS Behavior](#).

5. Select option 8.

The following prompts guide you through the configuration process.

- Prompt: Configure library communications? (y/n):

Enter `y` to establish or update communications between the ACSLS server and any libraries. This includes adding, deleting, or changing port connections to libraries and adding, deleting, or changing library partitions.

Selecting `n` refreshes the configurations of your current libraries.

Bypassing configuring your library communications and just refreshing your hardware configuration is the easiest and best option if you do not need to: add or remove ACSs or add, change, or remove connections to libraries. You skip re-defining all of your current connections to your libraries and just refresh the library configuration recorded in the ACSLS database. This eliminates the risk of omitting a connection to a library or specifying it incorrectly.

- **Prompt:** Library server database exists and will be overwritten, continue? (y or n): y

This prompt only displays if library communications have been previously established and there is an existing database.

If you enter `y` to this prompt, changes are made to the configuration.

6. Specify the number of ACSs to be supported.

- **Prompt:** Number of ACSs to be supported:

Enter the number of ACSs (between 1 and 32) that are to be supported at your site. You must have at least "1" ACS. Hardware must be installed to support one or more ACSs.

You can skip ACS numbering when configuring or reconfiguring libraries without assigning the ACS numbers in sequence.

7. Specify each ACS number.

- **Prompt:** Please enter the first ACS number [default: 0]:

Enter the number for this ACS.

 **Caution:**

When you reconfigure existing ACSs, specify the same ACS numbers that are currently assigned to them.

You are then prompted to define the device connections for each ACS.

8. Specify and define library partitioning.

- **Prompt:** Is ACS #n in a partitioned library? (y or n)
  - If you enter `y`, you are asked for the partition ID for the ACS.  
This partition ID must match the partition ID configured on the library.
  - If this is not a partitioned library or is SCSI/Fibre-attached, enter `n`.  
ACSLs does not support partitioned SCSI/Fibre-attached libraries.

 **Note:**

An SL4000 library cannot be captured in a single ACS as a non-partitioned library. By default, an SL4000 library is a partitioned library with 2 omnipresent partitions 1 (Default partition) and 2 (Media validation partition).

9. Define the device connections (ports) for each ACS.

- **Prompt:** Number of connections to ACS #n

This prompt appears for each ACS configured and establishes the number of communication ports for each ACS. The connection types are:

- SCSI connection(s) using HTTPS
- TCP/IP connection between the ACSLS and the library. At least two connections are recommended.

- SCSI connection(s)

If you are connecting to multiple SL8500 libraries, a maximum of fifteen connections is allowed.

The ACSs must be physically connected. Enter a decimal number between 1 and 15.

- Prompt: Device or host - ACS#n, connection #n:

Enter the device or host for each connection.

Ensure that all connections specified for an ACS are actually connected to the same ACS.

 **Caution:**

Make sure you specify the correct hostname or IP address. Do not connect to the wrong library.

**SCI libraries:**

To specify the connection to a SCI attached library (SL4000), enter the Library Name (CN) value from the library certificate established during SL4000 configuration, prefixed by `https://`.

For example:

```
Device or host - ACS #0, connection #1: https://10.80.40.162
```

where `10.80.40.162` is the Library Name (CN) value.

If the library certificate specifies this value as a host name, you cannot use the corresponding IP address, or vice-versa. For example, if the library certificate CN name=`sl4000-acs4`, you must provide the matching URL (`https://sl4000-acs4`).

If using a host name (DN), not an IP address, it must also resolve to the same exact name. This may require DNS and/or `resolv.conf` modifications.

When prompted for authentication, enter the user name and password defined during SL4000 library configuration.

 **Note:**

ACSLS SCI connection to an SL4000 library requires an SL4000 user credential with a user role at the User level. The SL4000 Administrator role can also be used for this credential.

See [Before Running `acsconfig` for the SL4000](#) for more information about configuring SL4000 library certificates and users.

**TCP/IP Attached libraries:**

To specify the connection to a TCP/IP attached library (SL3000, SL8500), enter one of the following:

- IP address

- hostname
- fully qualified hostname

 **Note:**

If a host name is specified, it must be mapped to the IP address entered into an SL8500 or SL3000 library. This host name-to-IP address mapping is site-specific. Typically, this is done by the `/etc/hosts` file, your Domain Name Server, NIS, or NIS+.

Example:

Sample TCP/IP device names for SL8500 or SL3000 libraries:

```
Device or host - ACS #0, connection #0: hostname1
Device or host - ACS #0, connection #1: hostname2
Device or host - ACS #1, connection #0: fully_qualified_hostname
Device or host - ACS #2, connection #0: xxx.xxx.xxx.xx
```

 **Note:**

ACSLs only supports TCP/IP connections to an SL3000; ACSLS does not support Fibre-attached SL3000s.

**Fibre or SCSI-attached libraries:**

To specify the connection to a directly-attached library, these are connected through mchanger devices.

Example:

Sample device name for SCSI Library:

```
Device or host - ACS #1, connection #0: /dev/mchanger2
```

If you have more ACSs to define, you will be asked to specify their ACS numbers. See step 7.

**10. Create or update the configuration in the ACSLS database.**

- Prompt: This step builds a database image of your complete library hardware configuration. Before proceeding, make sure that your library is completely configured, that all subsystems are functional and powered on. Build/Verify library configuration? (y or n):

Enter `y`.

Before proceeding, ensure that your library is completely configured, that all subsystems are functional and powered on.

This step updates the database with all configuration changes, except adding or removing port connections.

- Prompt: Library server database exists and will be overwritten, continue? (y or n):

Enter `y`. As the script builds the library configuration, it displays the following message of each panel in each LSM.

```
ACS # n, LSM # nn, PANEL # nn, created
```

The script also generates a library configuration report and appends it to the following file:

```
$ACS_HOME/log/acsss_config.log
```

#### 11. Optionally, configure non-TCP/IP clients on Solaris.

- Prompt: Configure client system interfaces? (y or n):

Respond (y or n) when the system asks whether to configure client system interfaces.

Enter `n` unless you are using an ICL client with OSLAN protocol.

Entering `y` displays the following message:

```
CSI SELECTION TABLE
```

```
1) OSLAN CSI Not Selected
2) ONC/RPC CSI Always Selected
```

```
Do you want to change the CSI selection table (n):
```

If `y`, the following message appears:

```
Select OSLAN CSI (n):
```

ONC/RPC is always selected.

#### Note:

`acsss_config` automatically backups the database and then terminates.

#### 12. Run `acsss enable` to start the server.

You can view the Event Log to verify that the system is running.

ACSLs starts automatically when the server system is rebooted.

#### 13. Perform an audit of the Library

An audit updates the ACSLS database to match the actual inventory of library cartridges.

ACSLs is now ready for library operations. If you want to enter ACSLS commands, log in as `acssa`.

## Renumbering an ACS

If it is necessary or desirable for you to change the number of an existing ACS, see [acs\\_renumber.sh](#).

# 5

## Setting Variables that Control ACSLS Behavior

ACSLs has both static and dynamic variables:

- Static variables

Static variables are identified by a sentence in the prompt that describes the variable and allows you set a new value.

**Example:** Changes to `<variable_name>` will not take effect until product is restarted.

 **Note:**

```
/usr/sbin/ndd -set /dev/tpc tcp_ip_abort_cinterval 15000
```

- Dynamic variable

Changes to an ACSLS dynamic variable take effect when the variable is next referenced during ACSLS operation.

### Easy ways to display and update ACSLS variables:

- Display the current settings of all ACSLS variables (static and dynamic):

```
dv_config -d
```

- Save the current settings of the variables to a file:

```
dv_config -d > filename
```

- View the file:

```
vi filename
```

- Print values of dynamic options:

```
dv_print
```

- Display the current setting of one variable with a prompt that explains the variable and enables changing the setting:

```
dv_config -p variable_name > -u
```

where:

- -p

Prompt enables you to specify a new value for the variable.

- `<variable_name>`

Insert the desired variable name here.

- -u

Update the values of dynamic variables in shared memory if the variable is changed. The `-u` option is not used for static variables.

 **Note:**

Enter a question mark (?) at the prompt to see a more complete description of the variable. The prompt refreshes allowing you to change the variable.

You can also use the `acsss_config` menu to set the variables that control ACSLS behavior. With options 1-8, you are:

- [Setting CSI Tuning Variables](#)
- [Setting Event Logging Variables](#)
- [Setting General Product Behavior Variables](#)
- [Setting Access Control Variables](#)
- [Setting Automatic Backup Variables](#)
- [Rebuilding Access Control Information](#)
- [Defining Event Notification Settings](#)
- [Configuring or Updating your Library Hardware](#)
- [Registering for Email Notification of System Events](#)

## Accessing the ACSLS Feature Configuration Menu

When you install or upgrade ACSLS, system defaults have already been set based on most user environments. However, when necessary, you can use options 1-8 to change any of these settings. Options 1 through 7 allow you to change dynamic and static variables. For static variable to take effect, you must re-start ACSLS.

Use option 8 to add and define your library hardware. Use this option if you have just installed ACSLS for the first time, or, for example, when you must add a new library.

Select ? for help on each prompt.

To start `acsss_config`:

1. Exit out of the CDE until you reach the CDE login.
2. Log in as `acsss`.
3. Run the configuration script:

```
acsss_config
```

The ACSLS feature configuration screen appears.

The configuration script, `acsss_config`, fails unless all LMUs, LSMs, and transports are fully configured, turned on, and ready.

The following menu appears:

```
ACSLS Feature Configuration
```

```
Enter the number followed by Return for your choice from the following menu
```

to configure product behavior in that area.

Press ? followed by the Return key for help.

```
1: Set CSI tuning variables
2: Set event logging variables
3: Set general product behavior variables
4: Set access control variables
5: Set automatic backup parameters
6: Rebuild Access Control information
7: Event Notification settings
8: Define or Change Library Configuration
```

E: Exit

Menu choice:

## Verifying and Changing Dynamic and Static Variables

Options 1 through 8 allow you to change the following dynamic and static variables. For OSCI static variables, you must use `dv_config`. See [Setting OSCI Variables with `dv\_config`](#).

### Setting CSI Tuning Variables

The Client System Interface (CSI) handles communication between ACSLS and clients of other servers. It sets up how communications are handled for each client. If communication with one client is lost, the other clients are not affected and their communication continues without interruption. Multiple CSIs can run under ACSLS.

Select ? for help on each prompt.

Option 1 enables setting or changing the following:

- `CSI_CONNECT_AGETIME`

Prompt: *Maximum age in seconds of pending requests in CSI request queue [172800]*

A dynamic variable, this setting determines how long ACSLS holds on to client requests to which it has not responded.

Valid entry is: 600 to 315360000 seconds. Default is 172800 seconds.

- `CSI_RETRY_TIMEOUT`

Prompt: *Number of seconds between successive retries [4]*

The default is 4 seconds.

A dynamic variable, this option specifies the minimum amount of time, in seconds, that the CSI should wait between attempts to establish a network connection.

You should modify this value if timing problems occur between the CSC and CSI.

- `CSI_RETRY_TRIES`

Prompt: *Number of retries for the CSI before a timeout condition occurs [5]*.

A dynamic variable, this option specifies the number of attempts the CSI should make to transmit a message. Pending messages are discarded if a connection cannot be established within the number of retries specified. Default is 5 retries.

- `CSI_TCP_RPCSERVICE`



Prompt: *Changes to alter use of the TCP protocol will not take effect until the product is restarted. CSI support for RPC using the TCP protocol is enabled [TRUE].*

A static option, this option specifies if the CSI is to act as a TCP RPC server. Default is true.

- CSI\_UDP\_RPCSERVICE

Prompt: *Changes to alter the use of the UDP protocol will not take effect until the product is restarted. CSI support for RPC using the UDP protocol is enabled [TRUE].*

A static option, this option specifies if the CSI is to act as a UDP RPC server. You must accept the default for clients to communication with ACSLS over UDP. Default is TRUE.

ACSLs must be restarted for this option to take effect.

- CSI\_MULTI\_HOMED\_CL

Prompt: *Enable the CSI on server platform to handle request packets from a multi-homed client; i.e., a client from a system with more than one IP address. Also, respond to the IP address modified by NAT (Network Address Translation) or a VPN. (TRUE/FALSE) [FALSE]*

This option allows the CSI on the server to handle incoming packets where the IP address in the packet header is not the same as the IP address from which the packet was sent. The address in the request packet that the client sends to ACSLS is derived from a 'hostname' call by the client application, followed by a `get hostbyname` lookup. This puts the local host IP address into the request packet.

Normally, this works. However, there are exceptions:

- In most cases, a client to ACSLS has a single, fixed host address. However, a client system may have more than one network interface and use any of its multiple IP ports to submit requests to ACSLS.
- A client system may be behind a NAT, with an internal, private IP address that is not accessible externally.
- When the client is connected to the ACSLS server through a VPN, and the client's IP address is not the IP address to which ACSLS must respond.

In these cases, ACSLS must respond to an IP address other than the localhost address at the client application. The CSI\_MULTI\_HOMED\_CL variable is designed to handle such situations. Valid sections are:

- FALSE

This is the default setting for ACSLS operation. This causes ACSLS to always respond to the localhost's IP address when returning a response to the client.

- TRUE

ACSLs disregards the IP address in the request packet. Instead, when ACSLS responds to the client, it uses the address found in the IP header of the incoming RPC datagram associated with the ACSAPI request packet. That is the IP address from which the request was submitted. If Access Control is enabled, this alternate IP address must be specified in the internet.addresses file.

- CSI\_USE\_PORTMAPPER

Prompt: *Changes to alter use of the port mapper will not take effect until the product is restarted. Enable port mapper: (ALWAYS / NEVER /IF\_DUAL\_LAN\_NOT\_ENABLED) [/IF\_DUAL\_LAN\_NOT\_ENABLED].*

Default is IF\_DUAL\_LAN\_NOT\_ENABLED. A static option, the valid options are:

- ALWAYS - the port mapper should always be interrogated when the CSI cannot send a message to a client.
- NEVER - the port mapper should never be interrogated when the CSI cannot send a message to a client. Select this option if clients do not support a port mapper.
- IF\_DUAL\_LAN\_NOT\_ENABLED - the port mapper should be interrogated only if dual LAN support has not been enabled. If dual LAN support has been enabled, then it is assumed that clients do not support a port mapper. Choosing this option provides for backward compatibility in product behavior.

- SURROGATE\_PROCESSES

Prompt: *Number of ACSSURR persistent processes that should be started [0]:*

Valid entries: 0 or 1. Enter 0 if the LM Gateway is not installed.

- SURROGATE\_PORT

Prompt: *TCP/IP port number that the ACSLS surrogate (ACSSURR) socket will listen on for requests from a gateway system [50300].*

This variable applies only to the Library Management (LM) Gateway. Valid entries: 50300 - 99999.

- SURROGATE\_TIMEOUT

Prompt: *Number of seconds to wait for data packets to be read on surrogate/gateway sockets:*

This variable applies only to the Library Management (LM) Gateway. Valid entries: 1 - 600.

- SURROGATE\_QUEUE\_AGE

Prompt: *Number of minutes to wait before deleting a stale queue entry [5].*

This variable applies only to the Library Management (LM) Gateway. Valid entries: 5 - 60.

- START\_CSCI\_PROCESS

Prompt: *Automatically start CSCI at ACSLS startup (TRUE/FALSE) [FALSE]:*

This variable determines if the CSCI process is automatically started during ACSLS startup. The default value, FALSE, means that the CSCI is not started with ACSLS. Set this variable to TRUE to start the CSCI process with ACSLS.

- CSI\_FIREWALL\_SECURE

Prompt: *Enable CSI to be used behind a firewall (user-defined inbound port) (TRUE/FALSE) [FALSE]:*

A dynamic variable, this setting enables or disables the firewall security definition of a single, user-defined port for incoming requests to ACSLS. Values are:

- False - if you do not want the ports on the ACSLS server to be restricted.
- True - this is the default. ACSLS server operates behind a secured firewall.

- CSI\_INET\_PORT

Prompt: *Port number used by the CSI to receive incoming ACSLS requests [30031].*

This variable specifies the single, user-defined port used by the CSI for incoming TCP requests from one or more clients.

This variable is used only when the firewall secure CSI is enabled by setting `CSI_FIREWALL_SECURE` to `True`.

Default: 30031

Valid entry: 1024 - 65535, but not 50003

- `CSI_FAILED_RESPONSE_LIMIT`

Prompt: *Number of recent responses to an audit, enter, eject, or event registration request that the CSI cannot return to a client before the CSI will drop all remaining responses to that request. [5]*

For most requests, if the CSI cannot send one response to a client, it will drop all remaining responses to that request. This protects the CSI from being tied up attempting to send responses to a client that cannot receive them.

For audit, enter, eject, or event registration requests, the CSI continues to try to return some responses to the request after the first failure.

`CSI_FAILED_RESPONSE_LIMIT` specifies the number of failed attempts to return responses to a request before the CSI drops all remaining responses to the request. This protects CSI availability when there are communication problems or a client is no longer active

You should only modify these values if timing problems occur between the client and the CSI; refer to the *ACSLs Administration Guide* for details.

Enter a number between 1 and 9.

## Setting Event Logging Variables

Select ? for help on each prompt.

Option 2 enables setting or changing the following event logs:

- `EVENT_FILE_NUMBER`

Prompt: *Number of Event Log files to retain [9]:*

This option specifies the number of additional Event Log files to retain. If this option is enabled and the current Event Log file size reaches the threshold size, the log is automatically copied to another file. Once the specified number of files is reached, the data in the oldest file is overlaid.

Enter **0** to retain no additional Event Log files; a single event log file grows until manually truncated. In this case, when the event log file size reaches the threshold size, the unsolicited "Event log full" message is displayed periodically until the event log is renamed or deleted. Nine (9) is the default value when the ACSLS is installed or upgraded.

Enter a number between **1** and **9**, inclusive, to enable this feature and specify the number of log files to retain. When the event log reaches the threshold size, the data is moved into the file `event0.log`. The next time the threshold size is reached, the `event0.log` data is moved to `event1.log` and the Event Log data is moved to `event0.log`. This cycle continues until the specified number of retained files is reached, at which point the data is discarded from the oldest file.

- LOG\_PATH

Prompt: *Changes to the logging directory will not take effect until the product is restarted. What directory should logging information be in [\$ACS\_HOME/log]:*

Select a directory where log files will be placed. As installed, ACSLS logs information to the \$ACS\_HOME/log directory. In normal usage, the value of this variable won't be changed. An alternate path may be specified if there are disk space problems in the file system which contains \$ACS\_HOME. The path given must be an absolute path (one which starts with / or \$ACS\_HOME).

The ACSLS product must be restarted for this variable to take effect.

- LOG\_SIZE

Prompt: *Maximum library server event log size in Kilobytes (=1000 bytes).*

This option specifies the threshold size for the Event Log in Kbytes (1,000 bytes). Enter a non-negative number. Default is 500 when ACSLS is installed or upgraded.

Valid entry: 32-2147483

- TIME\_FORMAT

Prompt: *Date/time format for all logs [%Y-%m-%d%H:%M:%S]:*

This option specifies the format to use for printing the date and time information in the event and trace logs. The format is the same as is used by the C language function, strftime.

- LM\_RP\_TRAIL

Prompt: *Should library manager request process auditing be turned on? [TRUE]:*

When set to TRUE, this option enables an audit trail of all ACSLS requests. Each request is logged with a time stamp and the name of the client or user interface from which the request was launched. The resulting log is found in the file, rpTrail.log.

- RP\_TRAIL\_LOG\_SIZE

Prompt: *Maximum size in Kbytes for the rpTrail Log [1000]:*

This option specifies the threshold size for the rpTrail log expressed in Kbytes. You cannot enter a negative number. When ACSLS is installed, or ungraded, 1000 Kbytes (1MB) is the default value. When the rpTrail.log exceeds this size, it is compressed and archived, and a new rpTrail.log is created for subsequent request auditing.

Enter a value between 10 and 10000.

- RP\_TRAIL\_FILE\_NUM

Prompt: *Number of rpTrail archive files to retain [10]:*

This option specifies the number of archived rpTrail log files to retain. When the current rpTrail.log size exceeds the threshold size, the log is automatically compressed and archived with a time stamp (yyyy-mm-dd\_hh:mm) taken the moment it was compressed. The archived files are saved in the "log\_archives" directory. Once the specified number of archived logs is reached and a new one is added, the oldest file is removed.

You can retain between 1 and 999 archived log files.

- RP\_TRAIL\_DIAG

Prompt: *Log diagnostic time stamps for internal request processing? [FALSE]:*

This option specifies whether to include diagnostic information in the audit trail.

If set to TRUE, time stamps for "QUEUED AT", "FORKED/WRITTEN TO AT", and "FINAL RESPONSE SENT TO <recipient" AT" is included in the trail log.

- XAPI\_LOG\_SIZE

Prompt: *Changes to the XAPI log size will not take effect until the xapi server is restarted. Maximum XAPI log size in Mbytes (=1048576 bytes) [20.]*

This option specifies the threshold size for the XAPI Log, expressed in Mbytes (here defined as "1048576 bytes."). Enter a non-negative number. 20 is the default value of this option.

The XAPI server must be restarted for this variable to take effect.

- XAPI\_LOG\_FILE\_NUM

Prompt: *Number of XAPI Log archive files to retain [10].*

This option specifies the number of archived XAPI Log files to retain. When the current vlog.file size exceeds the threshold size, the log file is renamed with a 0 to n suffix. 0 is the newest file and n is the oldest. The archived files are saved in the XAPI\_WORK\_PATH directory. Once the specified number of archived logs is reached, the oldest file will be removed from the archive directory each time a new one is added to that directory. You can retain no fewer than one archived file and no more than 99. Enter a number between 1 and 99 to specify the number of archived log files to retain.

The XAPI server must be restarted for this variable to take effect.

- XAPI\_TRACE\_SIZE

Prompt: *Changes to the XAPI trace size will not take effect until the xapi server is restarted. Maximum XAPI trace size in Mbytes (=1048576 bytes) [50].*

This option specifies the threshold size for the XAPI Trace, expressed in Mbytes (here defined as "1048576 bytes."). Enter a non-negative number. 50 is the default value of this option

The XAPI server must be restarted for this variable to take effect.

- XAPI\_TRACE\_FILE\_NUM

Prompt: *Number of XAPI Trace archive files to retain.*

This option specifies the number of archived XAPI TRACE files to retain. When the current vtrace.file size exceeds the threshold size, the trace file is renamed with a 0 to n suffix. 0 is the newest file and n is the oldest. The archived files are saved in the XAPI\_WORK\_PATH directory. Once the specified number of archived logs is reached, the oldest file will be removed from the archive directory each time a new one is added to that directory. You can retain no fewer than one archived file and no more than 99. Enter a number between 1 and 99 to specify the number of archived log files to retain.

The XAPI server must be restarted for this variable to take effect.

## Setting General Product Behavior Variables

Select ? for help on each prompt.

Option 3 enables setting or changing the following:

- LIB\_VOL\_STATS

Prompt: *Enable Library Volume Statistics Gathering: (ON / OFF) [OFF]:*

This option specifies whether Library Volume Statistics (LVSTATS) information will be collected from the enter, eject, dismount, mount and audit operations. Enter ON to start a LVSTATS gathering and reporting session. Enter OFF to end an LVSTATS gathering and reporting session. When a session is started or terminated, an appropriate message will be printed to the log file `acsss_stats.log`. Judicious use of the options causes log file bracketing for each of the individual operations listed above.

- `VOL_STATS_FILE_NUM`

Prompt: *Number of acsss\_stats log files to retain [9]:*

This option specifies the number of additional `acsss_stats` log files to retain. If this option is enabled and the current `acsss_stats` log file sizes reaches the threshold size, the log is automatically copied to another file. Once the specified number of files is reached, the data in the oldest file is overlaid.

Enter 0 to retain no additional log files; a single `acsss_stats` log file grows until manually truncated. In this case, when the `acsss_stats` log file size reaches the threshold size, the unsolicited `acsss_stats log full` message is displayed periodically in the Event Log until the `acsss_stats` log is renamed or deleted. Nine (9) is the default value when the ACSLS is installed or upgraded.

Enter a number between 1 and 9, inclusive, to specify the number of log files to retain. When the `acsss_stats` log reaches the threshold size, the data is moved into the file `vol_stats0.log`. The next time the threshold size is reached, the `vol_stats0.log` data is moved to `vol_stats1.log` and the `acsss_stats` log data is moved to `vol_stats0.log`. This cycle continues until the specified number of retained files is reached, at which point the data is discarded from the oldest file.

- `VOL_STATS_FILE_SIZE`

Prompt: *Maximum library server access\_stats log size in Kbytes (=1000 bytes) [500]:*

This option specifies the threshold size for the `acsss_stats` log, expressed in Kbytes (here defined as "1000 bytes.") Enter a value of 32 or above. When ACSLS is installed or upgraded 500 is the default value for the size, in kbytes, for this option.

Valid entry: 32-10000

- `UNIFORM_CLEAN_USE`

Prompt: *Select cleaning cartridge ordering method [VALID\_SORT]:*

Valid options, are:

- `VALID_SORT` - orders the cleaning cartridges by the volume identifier. This uses up a cleaning cartridge before moving to the next one. If you select this option, ACSLS uses and returns cleaning cartridges in the same order as in previous releases of the ACSLS software. This is the default value when ACSLS is installed or updated.
- `LEAST_USED` - orders the cleaning cartridges by usage. If you select this option, ACSLS sorts the volume list in reverse order of usage and returns the cartridges with the fewest usages first. This spreads out usage of cleaning cartridges uniformly.
- `MOST_CAPACITY` - orders the cleaning cartridges by number of uses left. If you select this option, ACSLS sorts the volume list based on the number of uses left on the cleaning cartridge and returns the cartridges with the most uses left first. This uses up all cleaning cartridges close to the same time.

- `AUTO_CLEAN`

Prompt: *Enable Transport Auto Clean Option (TRUE/FALSE) [TRUE]:*

Valid options, are:

- TRUE - when a transport requires cleaning, ACSLS automatically cleans the transport before the next mount.

The drive cleaning operation in a SCSI or fibre-attached library is managed by the library firmware, not by ACSLS. Be sure to enable drive cleaning from the LSM control panel. Otherwise, you can periodically mount a cleaning cartridge from ACSLS by manual command as needed.

- FALSE - does not enable automatic cleaning

- AUTO\_START

Prompt: *Library Server startup state (RUN/IDLE) [RUN]:*

This option specifies the initial ACSLS state after recovery completes during server software initiation. Valid options are:

- RUN - user requests are processed immediately after recovery completes.
- IDLE - prevents user requests from being immediately processed after recovery completes.

Setting this option to IDLE could be useful if you must vary device states or perform some operational activity before you allow server access to your users.

- MAX\_ACSMT

Prompt: *Changes to the number of mount processes ACSLS supports will not take effect until the product is restarted. Number of mount processes [2]:*

Valid entry is 1 to 10.

Oracle recommends that you accept the default value at initial configuration, then change the value as required. Increasing this value may improve performance. For large configurations (those having more than eight LSMs), this value may need to be increased to improve mount/dismount performance.

 **Caution:**

If this number is set too high, ACSLS may not start up properly. In this case, either lower the number or raise the maximum allowable processes per user.

Default is 2.

- ENABLE\_ACSQY

Prompt: *Enable QUERY persistent processes (TRUE/FALSE) [TRUE]:*

This option allows you to execute queries as one or more persistent processes. Queries may be executed as persistent or request processes. Accept the default value at initial configuration, then change the value as required. Increasing this value may improve performance. Valid options, are:

- TRUE - enables persistent query processes.

 **Caution:**

With a minimum configuration system, more than 10 query persistent processes could use up system resources and not allow ACSLS to start up.

- FALSE - disables persistent query processes. Queries are invoked as request processes.

- MAX\_ACSQY

Prompt: *Changes to the number of query processes ACSLS supports will not take effect until the product is restarted. Number of query processes [2]:*

Valid entry is 1 to 10.

This option specifies the number of persistent query processes to create. This option is valid only if you enabled it in the above step. Valid numbers are 1 to 5.

Oracle recommends that you accept the default value (2) at initial configuration, then change the value as required. For large configurations (more than eight LSMs), this value may need to be increased to improve performance.

 **Note:**

If this number is set too high, ACSLS may not be able to start up properly. Either lower the number or increase the maximum allowable process per user.

ACSLS must be restarted for this option to take effect.

- MAX\_ACS\_PROCESSES

Prompt: *Changes to the maximum number of ACSLS processes will not take effect until the product is restarted. Number of ACSLS processes [60]:*

Valid numbers are: 32 to 256.

A transient process satisfies all requests except: `mount`; `dismount`; `lock`; `unlock`; `clear_lock`; and `query_lock`. The default, 40 processes, works for all ACSLS processes except extremely large configurations. Default is 60.

 **Note:**

It is recommended you contact Support for advice before changing this value.

ACSLS must be restarted for this option to take effect.

- TRACE\_ENTER

Prompt: *Enable automatic enter tracing facility (TRUE/FALSE) [FALSE]:*

This option specifies if the results of automatic enter operations are written in the Event Log. Valid options, are:

- TRUE - enables messages to be written to the event log at the end of each automatic enter operation. This is the only method for determining why volumes were not entered during an automatic enter operation.



- FALSE - disables this feature and helps minimize the number of messages written to the Event Log.
- TRACE\_VOLUME  
 Prompt: *Enable volume tracing facility (TRUE/FALSE) [FALSE]:*  
 This option specifies if unsolicited messages are displayed when adding/deleting a volume(s) in the database. Valid options, are:
  - TRUE - enables unsolicited messages to be displayed whenever a volume is added or deleted in the database. Operations that can generate this messages are: audit; mount; dismount; enter; eject; recover; and vary.
  - FALSE - disables this feature and helps minimize the number of messages written to the Event Log.
- ABSENT\_VOLUME\_RETENTION\_PERIOD  
 Prompt: Number of days to retain volumes that are identified as ABSENT or EJECTED in the database. [5]:  
 A dynamic variable, it sets the number of days (0-999) to keep absent or ejected cartridges in the database. Unless you reenter the volume, ACSLS automatically deletes the volume from the database when the retention period expires.  
 If you never want the volume to be deleted, enter 999. To delete the volume immediately, enter 0.

 **Caution:**

Setting a high retention period can cause a large number of absent or ejected volumes to be retained. This uses more space in the database and causes it to fill up faster.

- ENABLE\_STATUS\_VOLUME\_ABSENT  
 Prompt: *Report ABSENT and EJECTED volumes to ACSAPI clients. [FALSE]:*  
 Valid options, are:
  - TRUE - reports absent and ejected volumes to ACSAPI clients.
  - FALSE - absent and ejected volumes are not reported to ACSAPI clients (as occurred in previous versions of ACSLS).
- ENABLE\_STATUS\_VOLUME\_MISSING  
 Prompt: *Status for MISSING volumes reported to ACSAPI clients: TRUE=MISSING. FALSE=IN TRANSIT [FALSE]:*  
 This option sets the status code to be returned to ACSAPI clients for missing volumes. Valid options are:
  - TRUE - reports missing cartridges to ACSAPI clients.
  - FALSE - missing volumes are not reported to ACSAPI clients (as occurred in previous versions of ACSLS).
- \*\*\*\*ENABLE\_INET\_ACSLM Must be TRUE\*\*\*\*

This variable must be TRUE to allow the GUI and logical libraries to communicate with legacy ACSLS processes. [TRUE]:

Legacy ACSLS processes used only UNIX-based sockets for inter-process communication. This variable enables INET-based socket communication for the GUI and logical library support. It must be TRUE to enable ACSLS 8+ features. The ACSLS product must be restarted for changes to take effect. This variable defaults to FALSE. You must manually set it to TRUE if GUI or logical library support is added.

- ALPHANUM\_VOL\_RANGES

Prompt: *Support alphanumeric volume ranges for commands and utilities. Alphanumeric ranges will include all valid vol\_ids, in ASCII collating sequence. Note - Changes to alphanumeric volume range support will not take effect until the product is restarted. (TRUE/FALSE) [FALSE]:*

Specifies whether alphanumeric volume range support is active for commands and utilities. Valid options, are:

- FALSE (default) - when specifying volume range, the first characters of a *vol\_id* range must be the same and the last variable characters must be numeric. Example:  
AAA000-AAA999
- TRUE - Any valid *vol\_id* for the range can be specified, if the alphanumeric volume range includes all valid volume IDs. Valid volume ranges consist of any combination of numbers (0-9), letters (A-Z), dollar sign (\$), pound sign (#), and spaces (leading and trailing).

Example: A1Z27BC-G\$123R

- EJECT\_RESPONSE\_ON\_CAP\_FULL

Prompt: *During eject processing, send an intermediate response containing ejected volumes when the CAP is filled, rather than waiting for MAX\_ID volumes to be ejected. (TRUE/FALSE) [FALSE].*

This option specifies whether the eject process should send an intermediate response when a CAP is filled, or wait for `MAX_ID` volumes to be ejected. On configurations with CAPs larger than `MAX_ID` cells, this does not affect any behavior, as an intermediate response is still generated when `MAX_ID` volumes are ejected. This setting will have no effect on `cmd_proc`, and `cmd_proc` will always receive a response when the CAP has been filled.

 **Caution:**

Changing the value of this variable will affect ACSAPI clients that expect intermediate eject responses to always contain `MAX_ID` volumes.

- MOUNT\_RETRY\_DELAY

Prompt: *To prevent mount and dismount failures when a library is busy or temporarily unavailable, requests are queued and retried. Delay, in minutes, before queued mount and dismount requests are retried or ACSLS checks if a temporarily offline library or drive is available. Range of 1-6 minutes [2]:*

When mount and dismount requests would fail because the library is busy or temporarily unavailable, they are automatically queued. The requests are periodically retried, or library and drive availability is re-checked. This variable specifies the time interval, in minutes, between attempts to retry the sequence.

- MOUNT\_RETRY\_TIME\_LIMIT

Prompt: *To avoid mount and dismount failures when a library is busy or temporarily unavailable, requests are queued and retried. Time limit, in minutes, to queue mount and dismount requests. After this time limit, the request fails. Range of 5-80 minutes [5]:*

To prevent mount and dismount requests from failing when the library is temporarily unavailable, they are automatically queued. The requests are periodically retried, or library and drive availability is re-checked. This variable is the maximum time that ACSLS will continue to queue a mount or dismount request.

- AUTO\_CLEAN\_RETRY\_LIMIT

Prompt: *Number of times to retry an automatic cleaning operation before abandoning the attempt to clean the drive and proceeding to mount the requested data cartridge. Range of 0-5 retries [1].*

When a drive needs cleaning and automatic cleaning is enable, ACSLS attempts to clean the drive before the next mount to the drive. If the cleaning operation fails because the cleaning cartridge is beyond its maximum use, or is otherwise unusable, ACSLS tries to select another cleaning cartridge and mount it to clean the drive.

AUTO\_CLEAN\_RETRY\_LIMIT restricts the number of times that ACSLS retries the cleaning operation before continuing and mounting the data cartridge that was requested.

- XAPI\_PORT

Prompt: *Changes to the user-defined inbound port to the XAPI server will not take effect until the XAPI server is restarted. Port number used by the XAPI server to receive incoming XAPI requests. Do NOT specify port 50003 [50020].*

This option specifies the port used by the XAPI server for incoming TCP requests from clients. Enter a number between 1024 and 65535 to define the port used by the XAPI server. Do **not** specify port 50003.

- XAPI\_WORK\_PATH

Prompt: *Changes to the XAPI work directory will not take effect until the xapi server is restarted. What directory should XAPI logging/tracing information be in [\${ACS\_HOME}/log/xapi]*

Select a directory where XAPI server work files will be placed. As installed, XAPI server logs information to the `${ACS_HOME}/log/xapi` directory. In normal usage, the value of this variable won't be changed. An alternate path may be specified if there are disk space problems in the file system which contains `${ACS_HOME}`. The path given must be an absolute path (one which starts with `/` or `${ACS_HOME}`).

The XAPI server must be restarted for this variable to take effect.

- LIMIT\_CAP\_CONCURRENT\_MOVES

Prompt: *When using large numbers of CAPs for ejects and/or enters in an ACS with multiple libraries, limit the number of concurrent moves to/from CAPs to reserve library resources for mounts and dismounts. (TRUE/FALSE).*

For each eject or enter operation ACSLS moves multiple cartridges to or from the CAP concurrently. (The default is four concurrent moves for each CAP.) In an ACS (library complex) of multiple connected libraries when many CAPs are being used

for ejects and enters at the same time, this can limit the library resources available for other requests including mounts dismounts.

Set `LIMIT_CAP_CONCURRENT_MOVES` to `TRUE` to reserve library resources for mounts, dismounts and so forth, when using large numbers of CAPS for enters and ejects simultaneously.

- `xapi_startup_file`

Prompt: *Changes to the XAPI startup file name will not take effect until the xapi server is restarted. Name of the XAPI startup file [startup].*

This option specifies the name of the XPI startup file. This file resides in the `XAPI_WORK_PATH` directory and includes XAPI startup parameters.

The XAPI server must be restarted for this variable to take effect.

- `BULK_CAP_EJECT_HANDLE`

Prompt: *When ejecting cartridges to an SL8500 Bulk CAP, leave a slot in each CAP magazine empty so it can be used as a handle. (TRUE/FALSE)*

Set `BULK_CAP_EJECT_HANDLE` to `TRUE` to leave a slot in each Bulk CAP magazine empty during ejects. The empty slot can be used as a handle. Depending on the height of the rail, either the bottom or top slot will be left empty.

- `ENTER_CLOSE_TO_DRIVES`

Prompt: *When entering cartridges into an SL8500 library, move them to a storage cell close to the tape drives. (TRUE/FALSE)*

When `ENTER_CLOSE_TO_DRIVES` is `TRUE`, ACSLS moves a cartridge being entered into an SL8500 to a storage cell close to the tape drives. This facilitates mounting the cartridge on a drive in the future.

When `ENTER_CLOSE_TO_DRIVES` is `FALSE`, ACSLS allocates new home cells for cartridges being entered in a round robin manner, allocating a home cell on each panel in turn.

- `DISMOUNT_AWAY_FROM_DRIVES`

Prompt: *When floating a cartridge to a new home cell on a dismount in an SL8500 library, move it to a storage cell away from the tape drives. (TRUE/FALSE)*

When `DISMOUNT_AWAY_FROM_DRIVES` is `TRUE`, ACSLS moves a cartridge being floated to a new home cell on a dismount in an SL8500 to a storage cell away from the tape drives. This leaves room by the tape drives for new cartridges.

When `DISMOUNT_AWAY_FROM_DRIVES` is `FALSE`, ACSLS allocates new home cells for cartridges being floated on a dismount in a round-robin manner, allocating an empty cell on each panel in turn.

- `XAPI_TAPEPLEX_NAME`

Prompt: *Changes to the XAPI Tapeplex name will not take effect until the xapi server is restarted.*

*Name of the XAPI Tapeplex []*

This option specifies the name of the XAPI Tapeplex. The XAPI server must be restarted for this variable to take effect.

Please enter text whose length does not exceed 8.

## Setting Access Control Variables

Select ? for help on each prompt.

Option 4 allows you to set or change the following access control variables:

- `AC_CMD_ACCESS`

Prompt: *Changes to command access control will not take effect until the product is restarted. Access control is active for commands. (TRUE/FALSE) [FALSE].*

Specifies if access control is active for commands. This allows control over which ACSLS command each network client is allowed to execute.

The default, false, disables access control.

 **Note:**

You must stop, then restart ACSLS for this change to take effect.

- `AC_CMD_DEFAULT`

Prompt: *Default access for commands (ACCESS/NOACCESS) [ACCESS].*

Specifies the default access to commands if there is no relevant access control lists.

The default, ACCESS, allows command access unless otherwise denied through configuration information provided in the access control files.

- `AC_VOL_ACCESS`

Prompt: *Changes to volume access control will not take effect until the product is restarted. Access control is active for volumes. (TRUE/FALSE) [FALSE].*

Specifies if access control is active for volumes. This allows control over which ACSLS clients can access specific volumes. You must restart ACSLS for this option to take effect.

The default, false, disables access control for volumes.

 **Note:**

You must stop, then restart ACSLS for this change to take effect.

- `AC_VOL_DEFAULT`

Prompt: *Default access for volumes (ACCESS/NOACCESS) [NOACCESS].*

Specifies the default access to volumes if there is no relevant access control lists.

The default, NOACCESS, disallows volume access unless otherwise granted through configuration information provided in the access control files.

- `AC_LOG_ACCESS`

Prompt: *Messages will be logged when access to commands or volumes is denied (TRUE/FALSE) [FALSE].*

Specifies if a message is logged in the Event Log when access to a command or volume is denied. The command or volume id is logged and the access id and host id if this option is enabled.

## Setting Automatic Backup Variables

Select ? for help on each prompt.

Option 5 allows you to set or change the following automatic backup variables:

- Prompt: *Would you like to modify the automatic backup settings? ( y or n)*

Variable: None. The automatic backup settings are saved in crontab. Use `acsss_config` to modify these settings.

Valid options, are:

- Y - If yes, you are asked if you want to back up everyday. If "yes", then you must enter the time in `HH:MM` format.  
If no, you must select the day(s) of the week to back up. Enter the time in `HH:MM` format.
- N - you go to the next question regarding the retention period of SDM-managed database backup files.

- `RETENTION_PERIOD`

Prompt: *Set the number of days for the retention period of SDM-managed database backup files. See Help for explanation. [5].*

The option sets how many days DSM tracks the data on your storage cartridges before the cartridge is designated as not eligible for reuse. Valid number of days: 4-30

The retention period is the period of time for which you retain backups. Transaction log files are *retained* with each backup, increasing the space needed in the backup partition. When choosing a retention period, consider the following conditions specific to your site:

- The size and activity of your library
- The number of automatic and manual backups performed in a given period of time

### Note:

When you run a manual backup with the `bdb.acsss` utility, an automatic backup to local disk is also done, thereby increasing the number of backups in the backup partition.

The retention period should be greater than the interval between backups. For example, if you have a retention period of four days, you should take automated or manual backups at least three times a week, with no more than three days between backups.

- The period of time for which you would like to retain backups and transaction log files, keeping in mind that long retention periods increase the space needed in the backup partition.

## Rebuilding Access Control Information

Option 6 rebuilds your access control tables, if access control is alive and/or has been changed.

## Defining Event Notification Settings

Select ? for help on each prompt.

Option 7 allows you to define your event notification settings.

- CLIENT\_CHECK\_MESSAGE\_COUNT

Prompt: *Set the number of event messages sent to a client between two successive client checks by ACSLS. [100]*

This option specifies the number of event notification messages to send a client and the minimum check interval (set with next question) before ACSLS checks to verify the registered client is still alive.

Valid entry: 10 - 1000 messages

- CLIENT\_CHECK\_MIN\_INTERVAL

Prompt: *Enter the minimum interval between two successive client checks by ACSLS (in minutes). [30]*

ACSLs checks periodically if a registered client is still alive. This check is performed when the minimum interval and the event notification message count is exceeded.

Valid entry: 1 - 600 minutes

- CLIENT\_CHECK\_RESPONSE\_TIME

Prompt: *Enter the time a client has to respond to a client check with a check registration request (in seconds) [30]:*

When ACSLS checks periodically if a registered client is still alive, the client must respond with a check registration request within the specified response time. If the client does not respond within this time period, the clients' Event Notification registration is canceled.

Valid entry: 5 - 300 seconds

- CLIENT\_CHECK\_POLLING\_INTERVAL

Prompt: *Enter the polling interval for confirming Event Notification client registration (in minutes) [60]:*

ACSLs checks periodically if a registered client is still alive. At the polling interval specified, ACSLS sends a client check response to all clients who have not issued register, unregister, or check registration requests within this polling interval.

Valid entry: 10 - 1440 minutes.

You must re-start the `acsls` service if you have set any static variables. Use `acsss disable` followed by `acsss enable`.

You do not need do this for the dynamic variables.

## Configuring or Updating your Library Hardware

You must use `acsss_config` (option 8) to create your library hardware for the first time or to add new libraries or reconfigure existing libraries. After your initial library configuration, you can then use the `config` utility, to dynamically add, reconfigure, and remove LSMs and tape drives, add or reconfigure ACSs, and add port connections to libraries. For more information and procedures, see [Setting Variables that Control ACSLS Behavior](#).

See [Installing and Configuring Your Library Hardware](#) for more information.

## Setting OSCI Variables with `dv_config`

Use `dv_config` to confirm or update static OSCI variables.

- To update all OSCI variables, enter `dv_config -m 80`.
- To update a specific OSCI variable, enter `dv_config -p <OSCI variable name>`.

The ACSLS product must be restarted for these static variables to take effect.

ACSLs includes the following OSCI variables:

- `OSCI_PORT_RANGE_START`

Prompt: *First port of a range of 32 ports to be reserved for OSCI [80]:*

This option specifies the first port to be used for OSCI. One port is assigned to each SL4000 ACSs each. Since there can be maximum 32 ACSs only, the selected port and next 31 ports will be reserved for OSCI.

Valid entry: 50900 - 50969.

- `OSCI_CERT_FILE`

Prompt: *The location and name of the OSCI security certificate for SCI-enabled libraries [80]:*

When `OSCI_SSL` is enabled, this variable specifies the path and name of the OSCI security certificate file required by the HTTPS and SSL protocols.

Select a directory where you have placed the OSCI security certificate. As installed, ACSLS will search for a certificate file named `server.pem` in the `/export/home/ACSSS/data/external` directory. An alternate path or name may be specified if your installation requires security certificates be stored in a different location.

The `OSCI_CERT_FILE` variable must specify an absolute path (one which starts with a `/` character). ACSLS must be re-enabled for changes to the `OSCI_CERT_FILE` variable to take effect.

## Registering for Email Notification of System Events

Users with administrative duties may register for automatic e-mail notification of system events, including system boot events, and for ACSLS-HA systems cluster fail-over events.

To register for such events, users must add their e-mail address in the respective files under the directory:



```
$ACS_HOME/data/external/email_notification/  
  boot_notification  
  ha_failover_notification
```

Place the e-mail address of each intended recipient on a single line under the header remarks. Thereafter, every time the system boots, or the HA cluster fails over to the standby node, each registered user is notified by e-mail.

This capability assumes that the sendmail service has been enabled on the ACSLS server, and that network firewall constraints allow for e-mail communication from the data center.

# 6

## Access Control

Access Control provides the following:

- *Volume Access Control* allows volumes to be assigned to one client application. Other clients can be allowed to access the client's volumes.
- *Command Access Control* allows administrators to assign specific ACSLS commands to specific clients.

Both *volume access control* and *command access control* apply to users of client applications who submit requests through the ACSAPI.

*Access control* does not restrict access by administrative users who submit library requests using `cmd_proc` or the ACSLS GUI.

## Volume Access Control

When enabled, the volumes that are owned by a specific user are accessible only to that user or to trusted other users.

The first time you configure ACSLS for volume access control follow these steps:

1. Enable `volume access control` in ACSLS.
2. Associate a client application with a user name.
3. Define which other users may have access to the user's volumes.
4. Establish ownership of the volumes.

## Enabling Volume Access Control

To enable `volume access control` in ACSLS:

1. Run the configuration utility, `acsss_config`.  
The main menu displays.
2. Select Option 4 - Set Access Control Variables.  
Each variable is listed one at a time, and its current setting is displayed.
3. Click **Enter** to accept the current or default setting.
4. Select `[TRUE]` and click **Enter** when the utility displays the message:  
`Access control is active for volumes`
5. Select one of the following when the utility displays the message:  
`Default access for volumes [ACCESS/NOACCESS]`
  - Select `[ACCESS]` if your goal is to *disallow* access to *specific* users and allow access to all others.

This requires specific users to be listed in a `users.ALL.disallow` file or a specific `users.COMMAND.disallow` file. See [Defining other users that are allowed access to the user's volumes](#).

- Select [NOACCESS] if your goal is to *allow* access to *specific* users and disallow access to all others.

This requires specific users to be listed in a `users.ALL.allow` file or a specific `users.COMMAND.allow` file. See [Defining other users that are allowed access to the user's volumes](#).

If you want to log instances where access to volumes is denied, select [TRUE] in response to that prompt.

Whenever enabling or disabling volume access, you must restart ACSLS for the change to take effect.

## Associating a Client Identity with a User Name

Not all client applications pass a user ID with its ACSLS request packets. In cases where the client is not identified by a user name, you can assign a user ID.

1. Go to the `access_control` configuration directory:  
`$ACS_HOME/data/external/access_control`
2. Create a file by the name `internet.addresses` or copy the `internet.addresses.SAMPLE` file.
3. In this file, create a record for each client. Each record contains at least two fields: the client IP address followed by a corresponding user name. You can include additional fields for comments.  
Separate the fields with spaces or tabs, as shown in the following example:

```
192.0.2.1 ulyssis payroll department
```

You can create as many client-user associations as you have client applications.

- Where client applications pass the user name with the ACSLS request, the `internet.addresses` file authenticates the user name with the designated IP address and denies access where both fields do not agree with the values in the request packet. Where multiple clients are hosted from a common platform, the same IP address may be included multiple times in this file, and this address can be associated with as many user names as are rightly applied to that IP address.
  - Where client applications do not pass the user name with the request, the `internet.addresses` file establishes a user name for that client. In this case, only one user name may be associated with any client IP address.
4. Save any updates to the `internet.addresses` file:

- Run `acsss_config`.
- Select Option 6 - "Rebuild Access Control Information".

ACSLs dynamically recognizes the change.

or SNA and OSLAN clients that do not use TCP/IP, refer to the `lu62.names` or `adi.names` file in the `access_control` directory.

## Defining other users that are allowed access to the user's volumes

To grant other users access to a users' owned volumes:

1. Create a file `users.ALL.allow` or `users.ALL.disallow` in the `access_control` directory. You can copy the templates `users.SAMPLE.allow` or `users.SAMPLE.disallow`.
2. Add a record in the file for each owner, placing the owners' user ID at the left margin.
3. Specify affected users on the same line with each owner.
4. Separate the user names with spaces or tabs, as shown in the following example:

```
owner_john  user-Allie  user-andre
```

User names listed in the `users.allow` and `users.disallow` files must be unique, without regard to case. The type case of characters in the user name is ignored.

Users who are not listed on the same line with the owner are given the default (`ACCESS` or `NOACCESS`) relationship to the owner's volumes.

 **Note:**

You cannot have the same `owner_ID` and `user_ID` pair in both the `users.COMMAND.allow` and `users.COMMAND.disallow` files for the same command or `ALL`. You also cannot have the duplicate `owner_ID` and `user_ID` pair in the same `users.COMMAND.allow` and `users.COMMAND.disallow` files. This includes repeating the same `user_ID` on the same line. If there are more allowed users for an owner than will fit on one line, the list of allowed users can be continued on subsequent lines. Each line must start with the owner ID.

5. Optionally, you can establish exceptions to the volume access policy you have defined. Generally, users are allowed full access, or no access to volumes that are under access control. However, it is possible to allow users certain restricted access to other users' volumes.

For example, you can set a policy that allows any user to query volumes that are owned by a specific user, even though they may not mount or dismount those volumes. Exceptions can be applied to any of the commands that are affected by access control:

To configure volume access policy exceptions for certain commands:

- You must create a `users.COMMAND.allow` or `users.COMMAND.disallow` file (where `COMMAND` is replaced by the specific command you want to grant or restrict).

The `users.COMMAND.allow` and `users.COMMAND.disallow` files must have a command component with the name specified exactly as listed below, with the name of the command in uppercase. Controlling access to other variants of commands (such as `QUERY_VOLUME`) is not supported.

```
DISMOUNT
EJECT
LOCK
MOUNT (1)
MOUNT_READONLY (2)
QUERY
REGISTER
SET_CLEAN
SET_SCRATCH
UNLOCK
```

## Notes:

- MOUNT (1) - MOUNT policies also apply to mount scratch. Policies do not apply to mount readonly
  - MOUNT\_READOLNY (2) - Policies for mount readonly are defined separately from mount.
  - The considerations above about no duplicate owner ID and allowed user ID pairs and continuing lists of allowed users on subsequent lines also apply to lists of disallowed users.
- For each owner, place the owner's name at the left margin, followed by the users for whom the policy applies.
6. Save any updates to the policies you define:
- Run `acsss_config`
  - Select Option 6 - "Rebuild Access Control Information".

ACSLs dynamically recognizes the change.

## Establishing Ownership of Your Volumes

Volume access control applies only to volumes that have explicit ownership. Unowned volumes in the library are accessible to any user. To explicitly set volume ownership use the `cmd_proc` interface:

```
ACSSA>set owner "daffy" volume V00100-V00199
Set: owner set for volumes V00100-V00199
Set: Set completed, Success.
```

You can remove ownership in a similar fashion by using an empty string:

```
ACSSA> set owner "" volume V00100-V00199
Set: owner set for volumes V00100-V00199
```

This operation clears the ownership from all of the volumes in the range. For more information, see [set owner](#) .

Volume ownership can be set automatically by the `watch_vols` utility. For more information, see [watch\\_vols](#) .

## Ownership policies

A policy for setting and removing ownership automatically can also be defined in ACSLS. For example, you can set a policy in which any scratch volume that is mounted becomes owned by the user who mounted it. Thereafter, the volume is owned by that user. The same policy could be enhanced to remove ownership whenever the volume is returned to scratch status. A policy could be written such that all entered volumes are assigned to a default user, or to the user who requested the enter, or if the volume was previously owned, to its prior owner. Considerable flexibility is offered with this feature.

Ownership policies are defined in the `ownership.assignments` file which resides in the `access_control` directory. You can set a policy in this file to assign or to un-assign ownership automatically with each `enter` or `automatic enter`, `set scratch`, or `mount scratch` operation. The `ownership.assignments` file allows you to define a

default owner. Whenever a volume encounters any of these operations, its ownership can be assigned to:

- `Owner_default` (the default owner)
- Same (the previous owner)
- Requestor (the user issuing the current request)
- Unowned (retract ownership from the volume)

 **Note:**

Instructions for defining ownership policies are described in detail in the `ownership.assignments` file. This file includes a complete list of commands that can be used to set volume ownership.

- Save any updates to the policies you define:
  - Run `acsess_config`
  - Select Option 6 - Rebuild Access Control Information.

ACSLs dynamically recognizes the change.

## Verifying Ownership

To verify ownership you can run `volrpt` using the `owner_id.volrpt` template.

```
cd ~acsss/data/external/volrpt
volrpt -f owner_id.volrpt
```

This produces a display of all the volumes in the library listed with their associated owner.

## Volume Access Summary

The following commands are supported by Volume Access Control:

```
dismount*
display
eject
enter
lock
set_clean
set_scratch
mount
query_mount
query_scratch
query_volume
unlock
```

Access control does not apply to `dismount force` since the `force` option instructs StorageTek ACSLS to ignore the volume ID and to dismount to volume unconditionally, unless it is in `STATUS_DRIVE_RESERVED` state.

The following table summarizes the contexts that apply when `volume access control` is enabled.

**Table 6-1 Volume Access is Enabled - ACCESS**

Default access for volumes is ACCESS	Access Allowed	Access Denied
Access is through <code>cmd_proc</code>	X	-
The specified volume is unowned	X	-
The user is the owner of the volume	X	-
The user is associated with the owner in <code>users.ALL.disallow</code>	-	X
If the user is not associated with the owner in <code>users.ALL.disallow</code>	X	-

**Table 6-2 Volume Access is Enabled - NOACCESS**

Default access for volumes is NOACCESS	Access Allowed	Access Denied
Access is through <code>cmd_proc</code>	X	-
The specified volume is unowned	X	-
The user is the owner of the volume	X	-
The user is associated with the owner in <code>users.ALL.allow</code>	X	-
If the user is not associated with the owner in <code>users.ALL.allow</code>	-	X

## Command Access Control

Command access control allows an ACSLS administrator to restrict certain classes of commands to specific client applications or specific users across the network. Controlled access applies only to user commands that are submitted through the ACSAPI and it does not apply to local users who submit commands using `cmd_proc`.

The process to configure ACSLS for *command access control* involves three steps.

The first time you configure ACSLS for command access control follow these steps:

1. Enable command access control in ACSLS.
2. Associate a client identity with a user name.
3. Define what commands are available to which users.

## Enabling Command Access Control

To enable *command access control* in ACSLS,

1. Run the configuration utility, `acsss_config`.  
The main menu displays.
2. Select Option 4 - Set Access Control Variables.

Each variable is listed, one at a time, and its current setting is displayed.

3. Click **Enter** to accept the current or default setting.
4. Select **TRUE** and click **Enter** when the utility displays the message `Access control is active for commands`.
5. When the message "Default access for commands" is displayed:
  - Select `ACCESS` if you want to *allow* all users access to *all* commands.  
To block specific users from issuing commands, they must be listed in a `command.ALL.disallow` file or a specific `command.XXX.disallow` file, where:  
`XXX` is the command for which access control is intended
  - Select `[NOACCESS]` if you want to *deny user* access to commands.  
To allow specific users to issue commands, they must be listed in a `command.ALL.allow` file or a specific `command.XXX.allow` file.

 **Note:**

- If you want to log instances where access to commands is denied, enter "TRUE" in response to that prompt.
- Whenever enabling or disabling command access, you must restart ACSLS for the change to take effect.

## Associating a Client Identity with a User Name

Refer to the procedures under [Associating a Client Identity with a User Name](#).

## Defining What Commands are Available to Which Users

This process depends upon the default behavior you have selected when you enabled *command access control*. You must create a policy file in the `$ACS_HOME/data/external/access_control` directory.

- If the default behavior you defined above is `[NOACCESS]`, you must create a `command.ALL.allow` file that contains the user ID of each client that is to have access to all ACSLS commands. Each user ID should be listed on a separate line in the file.  
If you want to grant only specific commands to specific users, you must create `command.XXX.allow` files for each command the users are allowed to execute. For example, to grant permission for specific users to enter volumes into the library, you would create a file with the name `command.ENTER.allow` and list the ID of each qualified 'enter' user on a separate line in the file.
- If the default behavior you defined above is `[ACCESS]`, you must create a `command.ALL.disallow` file that contains the user ID of each client that is not to have access to all ACSLS commands. Each user ID should be listed on a separate line in the file.



 **Note:**

You cannot have the same user\_ID in both the `command.XXX.allow` and `command.XXX.disallow` `command.XXX` files for the same command or *ALL*.

## Command Names for Command Access Control allow and disallow Files

The `command.XXX.allow` and `command.XXX.disallow` files must have a command component with the name specified exactly as listed below, with the name of the command in uppercase. Controlling access to other variants of commands (such as `QUERY_VOLUME`) is not supported.

```
AUDIT
CANCEL
CHECK_REGISTRATION
CLEAR_LOCK
DEFINE_POOL
DELETE_POOL
DISMOUNTDISMOUNT_FORCE
DISPLAY
EJECT
ENTER          (1)
IDLE
LOCK
MOUNT          (2)
QUERY
QUERY_LOCK
REGISTER
SET_CAP
SET_CLEAN
SET_OWNER
SET_SCRATCH
START
UNLOCK
UNREGISTER
VARY
```

 **Note:**

ENTER (1) - Policies apply to virtual enter and manual enter, but not for automatic enter. MOUNT (2) - Policies also apply to `mount scratch` and `mount readonly`.

Use the following table as a quick reference for determining when command access is allowed.

**Table 6-3 Command Access is Enabled - NOACCESS**

Default Access for Commands is NOACCESS	Access Allowed	Access Denied
The request is entered from <code>cmd_proc</code>	X	-
The user_ID is listed in <code>command.COMMAND.allow</code>	X	-

**Table 6-3 (Cont.) Command Access is Enabled - NOACCESS**

Default Access for Commands is NOACCESS	Access Allowed	Access Denied
The user_ID is listed in <code>command.ALL.allow</code>	X	-
-- All other conditions --	-	X

**Table 6-4 Command Access is Enabled - ACCESS**

Default Access for Commands is ACCESS	Access Allowed	Access Denied
The request is entered from <code>cmd_proc</code>	X	-
The user_ID is listed in <code>command.COMMAND.disallow</code>	-	X
The user_ID is listed in <code>command.ALL.disallow</code>	-	X
-- All other conditions --	X	-

- Save any updates to the policies you define:
  - Run `acsss_config`
  - Select Option 6 - "Rebuild Access Control Information".
 ACSLS dynamically recognizes the change.

## Logging Access Control Messages

You can set a policy to log all transactions that failed because the user was denied access. The message displays the user name and the command that was attempted.

To enable access control logging:

1. Run `acsss_config` and select Option 4 - "Set Access Control Variables"
2. Change [FALSE] to [TRUE] at the following prompt: "Messages will be logged when access to commands or volumes is denied."
3. Select Option 6 - "Rebuild access control information."

ACSLs recognizes the change and begins logging each time a command request was denied.

# 7

## Library Management

Managing your libraries may consist of the following tasks:

- [Specifying ACS Numbers](#)
- [Auditing the Library](#)
- [Adding a SCSI or fibre-attached library to ACSLS](#)
- [Using the Extended Store Feature](#)
- [Managing a Mixed-Media Library](#)
- [Managing Dual TCP/IP Connections](#)
- [Managing a Dual-LAN Client Configuration](#)
- [Registering the IP Addresses of Primary and Secondary LANs](#)
- [Setting the TCP/IP Connection Timeout Interval](#)
- [Registering for Event Notification](#)
- [Automatic Recovery to Desired State](#)
- [Queue and Retry Mounts and Dismounts when Library is Temporarily Unavailable](#)
- [Reconfiguring ACSLS When Tape Drives are Moved, Added, or Removed](#)
- [Media Validation](#)

### Specifying ACS Numbers

ACSLs lets you configure or reconfigure libraries without assigning all ACS numbers in sequence.

**Example:**

You want to migrate from an SL8500 library to an SL3000 library. The SL8500 ACS is now ACS 0 and the SL3000 ACS is ACS 1.

Skipping ACS numbers lets you migrate all of the cartridges and drives to the SL3000 ACS and remove the SL8500 ACS without having to renumber the SL8500 ACS.

If you renumbered the SL3000 ACS all volumes would be deleted, and then re-added in a subsequent audit. It would also cause all of the `drive_ids` to change.

To change the number of an existing ACS, see [acs\\_renumber.sh](#) .

### Reconfiguring ACSLS When Tape Drives are Moved, Added, or Removed

You must reconfigure ACSLS to update the drive serial numbers and drive types in the ACSLS database whenever tape drives are swapped between locations or a tape drive in the

library is replaced with a different tape drive. Reconfigure to add or remove tape drives when they are inserted into or removed from a library. Adding tape drives to the Media Validation pool removes them from access by ACSLS, and removing tape drives from the Media Validation pool adds them to ACSLS.

If a tape drive is replacing an existing drive, the drive type and serial number is not updated until ACSLS reads the tape drive status from the library. This happens at:

- ACSLS startup
- When the ACS or LSM goes not ready and then ready, causing ACSLS to recover the library
- When the customer varies the ACS, LSM, or tape drive offline and back online
- When the customer reconfigures the tape drives, LSM, or ACS.

When a tape drive is added or removed, the ACSLS configuration must be updated to add or delete the drive in the ACSLS database.

Updating the ACSLS tape drive configuration avoids errors on mounts and prevents mounting a cartridge in the wrong tape drive.

## Updating Tape Drives Configured to ACSLS

If tape drives have been moved or replaced in a library, use Dynamic Configuration to update the drive types and serial numbers. If one tape drive is replacing another at the same location, the update does not require customer confirmation. If tape drives are being removed from or inserted into a library, the customer must confirm the configuration change.

Notes:

- Make sure all affected components are ready before issuing the `config` request.
- Updating the ACSLS database using Dynamic Configuration is done when ACSLS is enabled. Dynamic Config is non-disruptive, and ACSLS can continue processing requests while the configuration is updated.
- After issuing `config lsm` or `config acs`, it is a good idea to audit the affected LSM or ACS.

Use Dynamic Configuration utility commands from the Unix command prompt to update tape drive configurations. You must be logged in as `acsss`.

- `config drive <panel_id>`

If the change only affects the tape drives on a single panel or SL8500 rail, use `config drive <panel_id>` to update the drive configuration for all tape drives on the panel.

- `config lsm <lsm_id>`

Use `config lsm <lsm_id>` to update the drive configuration for all tape drives in an SL3000 with two drive panels.

 **Note:**

`config lsm <lsm_id>` also updates the CAPS and storage capacity in the LSM, and should be followed by auditing the LSM.

- `config acs <acs_id>`

Use `config acs <acs_id>` to update the configuration for everything in an ACS (for example, an SL8500 library complex).

 **Note:**

`config acs <acs_id>` also updates the CAPS and storage capacity of the entire ACS, and should be followed by auditing the ACS.

## Auditing the Library

An `audit` updates the ACSLS database to match the actual inventory of library cartridges.

### How Audit Works

An audit ejects duplicate and invalid volumes. The cartridges have:

- An external label that duplicates one already scanned.
- A missing or unreadable external label and no virtual label.
- An invalid media type.
- An invalid volume ID.

ACSLS records any database changes from the audit in the Event Log, and also displays `cmd_proc` messages during the audit. Audits only apply to LSM storage cells, not to tape drives or CAPs. For information about running an audit, see [audit](#).

### When to Run an Audit

You run an audit to:

- Create volume information in the database for newly configured libraries.
- Add volumes to the database when the cartridges were not entered through the CAP.  
Example: You added an LSM to your library, opened the LSM door, and manually added cartridges to the LSM.
- Resolve discrepancies between the library and the database.  
Example: Run an audit if you opened an LSM door and manually removed cartridges instead of ejecting them through the CAP. The audit either marks the volume absent or deletes the removed volumes from the database.

Audit an SL3000 after it is configured or reconfigured to ACSLS. The library reports the potential cell locations that are not accessible to ACSLS when ACSLS audits it. Inaccessible cell locations include:

- locations where CAPs, drives, and operator panels are installed.
- cell locations that the robots cannot access.
- cell locations that are not activated.
- cell locations that are not in this partition.

## Audit Intervals

Audit intervals depend on several factors, including: your ACSLS configuration; whether the library is reporting cell contents from its database or by examining each cell; your library configuration; the number of database changes required; and the scope of the audit. The following table describes how the differences in the scope of an audit affect the audit intervals.

**Table 7-1 How the Scope of an Audit Affects the Audit Interval**

This audit takes less time	Than this audit
A diagnostic ACS/LSM	An online ACS/LSM
An ACS/LSM dedicated to the audit	An ACS/LSM processing other requests
A full panel	A (partially or completely) empty panel
A drive panel	A standard panel
An inside panel	An outside panel

In addition, consider the LSM type for the component you audit. The average audit time for an SL8500 or SL3000 LSM is five minutes (if the SL8500 physical audit completes before the ACSLS audit).

## Adding a SCSI or fibre-attached library to ACSLS

The SCSI Media Changer (mchanger) is the device driver that communicates between ACSLS and any SCSI library. An mchanger must be created for each SCSI or fibre-attached library that is connected to ACSLS.

For more information, see [Adding the SCSI mchanger Device Driver](#).

## Using the Extended Store Feature

The following sections provide information about using the Extended Store Feature.



### Note:

This feature does not apply to a single LSM without a pass-thru port.

When a cartridge is mounted, its “home location” is the storage cell from which it was mounted. Normally, when a cartridge is dismounted and its home location is in a different LSM than the drive, ACSLS attempts to assign a new home location in the closest LSM (the shortest pass-thru distance from the tape drive). This is called “floating” a cartridge to a new home location.

ACSLs attempts to return a cartridge to its home location after a dismount if that home location is in an LSM that is enabled for the Extended Store feature.

 **Note:**

Events such as the volumes' home LSM being offline or a label mismatch on a dismount may prevent ACSLS from returning a cartridge to its home location. If such an event occurs, the cartridge is stored as close as possible to the LSM from which it was dismounted.

If you have an Extended Store LSM that you use for cartridge archive, this feature helps to ensure that cartridges from that LSM return to their home locations in that LSM. For example, if Extended Store LSM 3 is enabled for this feature and a cartridge from LSM 3 is mounted on a drive attached to LSM 1, after the cartridge is dismounted, ACSLS attempts to return the cartridge to its home location in LSM 3. If LSM 3 is not enabled, ACSLS tries to store the cartridge in a new cell in LSM 1.

You can only enable entire LSMs for this feature. You cannot enable LSM subcomponents, such as a panel or individual cell. To enable an entire ACS for this feature, you must enable each LSM in the ACS.

 **Note:**

Enabling the Extended Store feature will increase pass-thru activity when cartridges are dismounted. This may significantly decrease library performance.

To enable an LSM for the Extended Store feature, modify the `lsm_fixed_volume` file (found in the `$ACS_HOME/data/external/fixed_volume` directory). After you modify the file, stop and restart ACSLS to enable the specified LSMs.

Use the following conventions when you modify the sample file:

- Comment and blank lines are allowed throughout the file.
- Each LSM identifier line consists of the ACS number, followed by a comma, followed by the LSM number. No spaces or tabs are allowed between the ACS number and the LSM number.
- Each LSM identifier must be on a separate line.
- No explicit ordering of the LSM identifier lines is required.
- No explicit end-of-file delimiter is required

Example of the Extended Store Feature Control File:

```
# This lsm_fixed_volume file must be found in the
# $ACS_HOME/data/external/fixed_volume
# directory. This is a sample lsm_fixed_volume file that may be
# edited your particular configuration.
# Comments may appear anywhere in this file, but must include a
# pound sign in the first column.
# Blanks lines are also allowed throughout the file for # readability, and
# will be ignored.
# For all the LSM identifiers found in this file, a "best" attempt
# will be
```

```

# made to return the volume to its home location at dismount.
# A valid LSM identifier consists of the ACS number, separated by a comma,
# and followed by the LSM number. Leading or trailing blanks are
# ignored.
#   ACS,LSM
#   0,0
#   0,1
# ACS 1, LSM 0 through 3
#   ACS,LSM
#   1,0
#   1,1
#   1,2
#   1,3

```

The following example is of a modified control file. In this example, the comment (#) character has been removed from the lines shown in bold to specify that LSMs 0,0 and 0,1 are enabled.

```

# This lsm_fixed_volume file must be found in the
# $ACS_HOME/data/external/fixed_volume
# directory. This is a sample lsm_fixed_volume file that may be
# edited your participation.
# Comments may appear anywhere in this file, but must include a
# pound sign in the first column.
# Blanks lines are also allowed throughout the file for
# readability, and
# will be ignored.
# For all the LSM identifiers found in this file, a "best" attempt
# will be
# made to return the volume to its home location at dismount.
# A valid LSM identifier consists of the ACS number, separated by a comma,
# and followed by the LSM number. Leading or trailing blanks are
# ignored.
#   ACS,LSM
#   0,0
#   0,1
# ACS 1, LSM 0 through 3
#   ACS,LSM
#   1,0
#   1,1
#   1,2
#   1,3

```

## Managing a Mixed-Media Library

With mixed-media, ACSLS supports a mixture of tape drive and media (cartridge) types in the same library. ACSLS mixed-media support prevents the robot from mounting incompatible media types in a tape drive. For example, in an SL8500, the robot will not mount an LTO cartridge in a T10000 tape drive.

ACSLs mixed-media support requires Media ID labels with media characters on cartridges.

### Note:

Refer to the ACSLS Product Information Guide for the current list of drive types, media types and drive-to-media compatibility supported.



ACSL S has limited knowledge of the recording format used on cartridges. ACSLS does not have access to the data path to tape drives, so ACSLS cannot detect and prevent recording format incompatibilities. However, recent T9840 and T10000 tape drives report the media recording format on a dismount. ACSLS saves this information in the database, and it can be displayed with the command:

```
display volume [vol_id(s)] -f recording_format_family recording_format_model
```

For more information, see [display volume](#) .

## Displaying the ACSLS Mixed-Media Settings

The `drives_media.sh` utility displays the drive types, media types, and drive-to-media compatibility settings currently supported by ACSLS. As support is added for new drives and media, they are displayed.

To display the ACSLS Mixed-Media settings, enter:

```
drives_media.sh
```

The information writes to the screen (standard output).

To output the ACSLS Mixed-Media settings to files in the `/tmp` directory, enter:

```
drives_media.sh -f
```

The information will be written to three files. (If the files already exist, they will be overwritten.)

```
/tmp/drive_types.txt  
/tmp/media_types.txt  
/tmp/media_compatibility.txt
```

## Mixed-Media Restrictions for SCSI-Attached LSMs

Note the following restrictions for SCSI-attached LSMs:

- Because some tape drives do not support dynamic write-protect settings are through the control path, you must use caution when mounting cartridges using the 'read-only' option. All StorageTek drives support this feature. For non-StorageTek drives, you are advised to confirm that dynamic write-protection is supported. Specifically, LTO drives do not support the "read-only" option.

Unless the drive supports this feature, you run the risk of losing data on cartridges that you assumed are to be mounted with the write-disable protection of a read-only mount.

- Automatic-cleaning operations in SCSI-attached libraries is handled by the library microcode and not by ACSLS. The library control panel provides a menu for users to enable library control of automatic-cleaning operations. For more information, see your library documentation.
- You can select a normal load or fast load option through the library. Not all tape management systems, however, support the fast load option.

## Setting Scratch Preferences

On `scratch mount` requests you can explicitly specify the media type you want to use, or you can have ACSLS select a media type.

In order for ACSLS to select a media type, you must pre-define a prioritized list of compatible media types for each drive type. This list is referred to as the “scratch preferences.”

- There is one set of preferences for the entire server; preferences are not defined by client.
- If a compatible media type is not listed for a drive, the media will not be selected.

The following sections describe the user and system defined files that ACSLS uses in determining scratch preferences.

## User-Defined Mixed-Media Files

The following are the user-defined mixed-media files that are located in `$ACS_HOME/data/external/mixed_media/`:

- `scratch_preferences.dat`  
User-defined preferences file. Primary source of preference definitions.
- `scratch_preferences.SAMPLE`  
Sample preferences file can be copied to create the `scratch_preferences.dat` file.

## System-Defined Mixed-Media Files

Display the ACSLS mixed-media settings using `drives_media.sh -f`

Review the media compatibility settings saved in:

- `/tmp/media_compatibility.txt`  
This is the system-defined compatibility settings file. It is used only if the user-defined preferences file does not exist or is missing a drive type
- `/tmp/drive_types.txt`  
System defined list of supported drive types.
- `/tmp/media_types.txt`  
System defined list of supported media types.

## How ACSLS Uses the Mixed-Media Files

The following table describes how ACSLS uses the mixed-media files to select a media type for a scratch mount request.

**Table 7-2 How ACSLS Uses Mixed-Media Files**

If the <code>scratch_preferences.dat</code> file ...	ACSLs does this...
Does not exist.	Uses the definitions from the system <code>media_compatibility</code> file
Lists more than one media type for a drive.	Selects the media types in the order listed.

**Table 7-2 (Cont.) How ACSLS Uses Mixed-Media Files**

If the <code>scratch_preferences.dat</code> file ...	ACSLS does this...
Lists no media type for a particular drive.	Uses the data from the system <code>media_compatibility</code> file
Does not list a particular drive type.	Uses the data from the system <code>media_compatibility</code> file

## Defining a Scratch Preferences File

Use this procedure to define a `scratch_preferences.dat` file, which contains an ordered list of scratch cartridge types to select for given drive types. ACSLS uses this file for `mount *` command where a media type is not explicitly specified.

The following example shows the contents of the `scratch_preferences.SAMPLE`.

```
Drive Type Name   Media Type Preference Name
4480              3480
SD3              DD3A
SD3              DD3B
SD3              DD3C
```

To define a scratch preferences file, do the following:

1. Log in as `acsss`.
2. Change to the external mixed media directory:

```
cd $ACS_HOME/data/external/mixed_media
```
3. Copy the sample scratch preferences file to create the user-defined file:

```
cp scratch_preferences.SAMPLE scratch_preferences.dat
```
4. Using a text editor such as `vi`, modify the list of preferences in the `scratch_preferences.dat` file:
  - Follow the instructions in the comments at the top of the file.
  - If you want to use more than one media type for a drive type, enter each media type on a separate line; the order of preference is from top to bottom.
5. Save the file.
6. From `cmd_proc`, restart ACSLS:

```
start
```

## Managing Dual TCP/IP Connections

Dual TCP/IP is an option that can be purchased for the SL8500 and SL3000. It provides two TCP/IP connections to the libraries. However, you can continue to use the libraries with only one of the two connections operational.

The purpose of dual TCP/IP is to automatically recognize and avoid a failing communication path. Since this is automated, there is no need for you to manually switch from an inoperative connection. For more information, see [Dual TCP/IP Support](#).

## Managing a Dual-LAN Client Configuration

ACSL 5.2 and above supports dual-LAN client configurations, which consist of a primary LAN and a secondary (backup) LAN. If the primary LAN fails, the client switches to the secondary LAN. ACSL removes all outstanding messages to that client and begins communicating using the secondary LAN. Because ACSL removes all outstanding messages before switching over to the secondary LAN, a client will not receive a success message even though a request completed successfully.

For example, a client requests ACSL to eject ten cartridges. ACSL starts ejecting the cartridges, then communications fail between ACSL and the primary client LAN. ACSL removes all outstanding messages to that client and begins communicating through the secondary LAN. ACSL successfully completes ejecting all ten cartridges, but sends no success message to the client. The client must verify that the request completed successfully. In this example, if the client issues a query volume request against the IDs of the ejected volumes, ACSL returns a `volume not found error` message, which confirms that ACSL ejected the cartridges.

ACSL ensures that any transient requests complete successfully if a LAN communications switch over occurs. After switch over, however, the client must cancel any persistent outstanding requests (such as CAP operations) originally submitted on the primary LAN and resubmit these requests through communications from the secondary LAN. Resources (such as, CAPs, locks, drives, and so forth) allocated through primary LAN communications will remain allocated after switchover to the secondary LAN.

The following sections tell how to configure ACSL for dual-LAN client operations by doing the following:

- Register the IP addresses of primary and secondary LANs by creating a `csc_ip_switch.dat` file
- Install a second ethernet port for a multi-homed ACSL server
- Set the TCP/IP connection timeout interval to decrease the system switchover time to the backup LAN.

### Registering the IP Addresses of Primary and Secondary LANs

To register the IP addresses of the primary and secondary LANs with ACSL, create the file `csc_ip_switch.dat` in the `$ACSS_HOME/data/internal/client_config/` directory. The following is an example of a `csc_ip_switch.dat` file.

```
#The following entry is System Zed's primary and secondary LAN IP addresses.  
129.80.30.40 129.80.30.50
```

As shown in the example above, the `#` sign precedes a comment. The entries consist of a client system's primary LAN IP address in the left column followed by one or more blanks, then a client systems' secondary LAN IP address in the right column. In this example, System Zed's primary LAN IP address is `129.80.30.40` and its secondary LAN IP address is `129.80.30.50`.

After you create or update the `csc_ip_switch.dat` file, if ACSL is running, you must stop and restart ACSL.

If ACSLS successfully reads the `csc_ip_switch.dat` file on restart, ACSLS logs the success message `2010 I DUAL PATH OPTION ACTIVATED` in the ACSLS Event Log. Otherwise, dual-LAN support is not activated.

## Installing a Second Ethernet Port for a Multi-Homed ACSLS Server

This section describes the procedure for installing a second ethernet port for a multi-homed ACSLS server. The second ethernet port provides the attachment to the second control path adapter, which controls the backup LAN. This procedure requires the SBus Buffered Ethernet card. You can order this Ethernet card from Oracle as part number X1053A.

The installation procedure includes:

- Installing the hardware and rebuilding the kernel
- Defining a host name for the new ethernet port
- Creating the `/etc/notrouter` file

To install a second ethernet port, do the following:

1. Install the SBus Buffered Ethernet card according to the manufacturer's instructions.
2. Reconfigure the system kernel for the new device:
  - a. Power the system on. When it begins to boot, press `[[STOP]]-[[A]]` to enter the PROM monitor.
  - b. At the `ok` prompt, boot the server:

```
boot -r
```

3. When the boot has completed, login as `root`.
4. Create a hostname for the second ethernet port:

```
echo 2nd_host_name > /etc/hostname/hme1
```

Where `2nd_host_name` is the hostname for the second ethernet port.

5. Enter the following command:

```
touch /etc/notrouter
```

This creates the `/etc/notrouter` file.

6. Reboot the server:

```
reboot
```

This completes the procedure.

## Setting the TCP/IP Connection Timeout Interval

The UNIX system variable `tcp_ip_abort_cinterval` sets the TCP/IP connection timeout interval between the client and the ACSLS server. Changing the default value (180 seconds) of this variable may decrease the switchover time to the backup LAN. Note, however, that actual switchover time depends on the configuration and type of failure.

For example, if an ACSLS server port fails and multiple clients are communicating through this port, ACSLS serially recovers communications with each client. Multi-client recovery for a

failing port therefore takes longer than recovery with a single client communicating with a failing port.

To decrease the switchover time to the backup LAN, do the following:

1. Log in as `root` on the ACSLS server.
2. From the prompt, enter the following command:

```
/usr/sbin/ndd -set /dev/tpc tcp_ip_abort_cinterval 15000
```

This command changes the TCP/IP connection timeout interval to 15 seconds (the default is 180 seconds).

**HINT:** To make this command persistent (across server reboots), add the command to the `/etc/rc2.d/S69inet` file in the “Set configurable parameters” section.

## Registering for Event Notification

Event notification allows tracking of events occurring in tape libraries for ACSAPI clients. This feature is provided by the CSC Toolkit. Specifically, with event notification, the ACSAPI client can do the following:

- Register for library resource events and/or volume events
- Un-register for these events
- Check registration status for library events and be notified when they occur

Client registration requests and event notification messages are delivered to the client until the registration is dropped. Event notification periodically checks the registration status of the client to verify that the client is alive. This avoids sending responses to clients that are no longer active and the unnecessary use of network resources.

The following types of events can be tracked:

- Addition and deletion of volumes and when maximum usage is exceeded for cleaning cartridges in the ACSLS database.
- Changes in library component status such as an LSM or drive changing from online to offline, diagnostic, or recovery; or a CAP that is opened or closed.
- Hardware failures such as an inoperable robotic hand.

## Automatic Recovery to Desired State

ACSLS now restores tape libraries and drives to the availability you desire. This is done by tracking both the current state and the desired state for ACSs, ports, LSMs, and tape drives. When a library or drive is no longer accessible by ACSLS or is inoperative, ACSLS changes its current state to offline. After the library or drive becomes accessible or operative again, ACSLS will automatically recover it and bring it back online if its desired state is online.

### Current state and desired state

- Desired state manages library and tape drive availability. Desired state is the availability that you want for an ACS, port connection, LSM, or tape drive. The initial desired state for all library components is online. You can set the desired

state through an explicit vary command. (This is a vary from `cmd_proc`, the ACSLS GUI, or an ACSAPI client. It is not a vary that ACSLS internally generates because of a library status change.) The desired state for library components can be viewed using the query `lmu` and `display` commands.

- The current state (specified as "state") for an ACS, port, LSM, or drive is the current availability of the component limited by desired state. Current state reflects whether a library component is ready and communicating, limited by the desired state of the component and higher level components.

For example, if the desired state of an LSM is online but it is not ready, its current state is offline. When the LSM comes ready again, the library sends ACSLS a message, and ACSLS automatically recovers the LSM and brings its current state back online.

However, if an LSMs' desired state is offline, ACSLS sets the current state to offline. If the LSM goes not ready and comes back to ready status, ACSLS will leave the LSMs' current state offline.

## Current state cascades down, desired state does not

The current state of a library component cascades down to lower level components.

- Specifically, if an ACS is no longer accessible, the current state of the ACS is offline. The current state of all LSMs and drives are also set to offline because they are also inaccessible.
- When the desired state of an ACS is changed to offline by a vary command, the current state of the ACS is set to offline, and the current state of all LSMs and drives in the ACS are set to offline.
- Likewise, when the current state of an LSM changes to offline, the current state of all drives in the LSMs become offline. The desired state of the drives does not change.

Changing the desired state of a library component does not affect the desired state of lower level components.

- Changing the desired state of an ACS does not affect the desired state of the LSMs and drives in the ACS.
- Changing the desired state of an LSM does not affect the desired state of the drives in the LSM.
- This lets you vary selected drives in an LSM offline to make them inaccessible. You can then vary the LSM offline for maintenance. When you vary the LSM back online, the selected drives remain offline, while the other drives come back online.

Availability of physical libraries and drives affects logical libraries and drives.

- The availability of logical libraries and tape drives in logical libraries is also governed by the desired state that you set for both the underlying physical library and the logical library.
- If the desired states for both the physical library and logical library are online, the current states of logical libraries and logical tape drives reflect the current state of the underlying physical libraries and drives.
- However, if the desired state for the physical ACS or drive is online, but the desired state of the logical library or drive is offline, the logical library or drive remains offline and unavailable.

## Queue and Retry Mounts and Dismounts when Library is Temporarily Unavailable

ACSLs queries mount and dismount requests when temporary library outages are detected. Mount and dismount requests from all sources are automatically queued and retried, if the desired state of all required library components is online. This means mount and dismount requests from the following sources are all automatically queued and retried:

- ACSAPI clients
- `cmd_proc`
- Fibre-attached clients for tape drives in logical libraries

Temporary outages occur when the desired state of the library hardware is **online**, but the current state is **offline**. Examples of temporary outages, are: when an LSM door is open; when ACSLS loses communications with a library; or during an LC switch operation. During a temporary library or tape drive outage, mounts and dismounts are queued, and are retried when the library is available.

If the desired state of the library hardware is *offline*, then ACSLS fails mount or dismount requests with the appropriate error status.

With ACSLS 7.3.1, two dynamic variables, `MOUNT_RETRY_DELAY` and `MOUNT_RETRY_TIME_LIMIT`, were introduced for managing the queuing and retrying of mounts and dismounts. The:

- `MOUNT_RETRY_DELAY` controls how often queued mounts and dismounts are retried or availability of libraries and drives are re-checked.
- `MOUNT_RETRY_TIME_LIMIT` is the time limit to queue and retry mounts and dismounts. After this, requests fail.

## Reconfiguring ACSLS When Tape Drives are Moved, Added, or Removed

You must reconfigure ACSLS to update the drive serial numbers and drive types in the ACSLS database whenever tape drives are swapped between locations or a tape drive in the library is replaced with a different tape drive. You must reconfigure ACSLS to add or remove tape drives when they are inserted into or removed from a library. Adding tape drives to the Media Validation pool removes them from access by ACSLS, and removing tape drives from the Media Validation pool allows ACSLS to access them. In these cases you also need to reconfigure ACSLS.

If a tape drive is replacing an existing drive, the drive type and serial number is not updated until ACSLS reads the tape drive status from the library. This happens at:

- ACSLS startup.
- When the ACS or LSM goes into a *not ready* and then *ready* state, causing ACSLS to recover the library.
- When you vary the ACS, LSM, or tape drive offline and back online.
- When you reconfigure the tape drives, LSM, or ACS.



When a tape drive is added or removed, the ACSLS configuration must be updated to add or delete the drive in the ACSLS database. Updating the ACSLS tape drive configuration avoids errors on mounts and prevents mounting a cartridge to the wrong tape drive.

## Updating Tape Drives Configured to ACSLS

If tape drives have been moved or replaced in a library, use Dynamic Configuration to update the drive types and serial numbers. If one tape drive is replacing another at the same location, the update does not require customer confirmation. If tape drives are being removed from or inserted into a library, the customer must confirm the configuration change.

When updating tape drives:

- Make sure all affected components are ready before issuing the `config` request.
- Updating the ACSLS database using Dynamic Configuration is done when ACSLS is enabled. Dynamic Config is non-disruptive, and ACSLS can continue processing requests while the configuration is updated.
- After issuing `config lsm` or `config acs`, it is a good idea to audit the affected LSM or ACS.

Issue these Dynamic Configuration utility commands from the Unix command prompt to update tape drive configurations. You must be logged in as `acsss`.

- `config drive <panel_id>`

If the change only affects the tape drives on a single panel or SL8500 rail, use `config drive <panel_id>` to update the drive configuration for all tape drives on the panel.

- `config lsm <lsm_id>`

Use `config lsm <lsm_id>` to update the drive configuration for all tape drives in an SL3000 with two drive panels.

 **Note:**

`config lsm <lsm_id>` also updates the CAPs and storage capacity in the LSM, and should be followed by auditing the LSM.

- `config acs <acs_id>`

Use `config acs <acs_id>` to update the configuration for everything in an ACS (such as an SL8500 library complex).

 **Note:**

`config acs <acs_id>` also updates the CAPs and storage capacity of the entire ACS, and should be followed by auditing the ACS.

## Media Validation

Media validation allows customers to verify all T10000 tape cartridge types using SLConsole or StorageTek Tape Analytics (STA). A dedicated “media validation pool” of T10000C and T10000D drives is used. The drives in the media validation pool are not available to ACSLS. If the drives were previously configured to ACSLS, the library reports them as “drive not installed” if ACSLS tries to access them.

### Adding Drives to the Media Validation Pool

When the drive is removed from ACSLS control and added to the media validation drive pool, the LSM (SL8500 rail or SL3000 library) that contains the drive first goes Not Ready and then Ready to the ACSLS. ACSLS also receives a Configuration Changed message from the library.

An ACSLS host automatically updates the drive to an offline state. If the drive will remain in the media validation pool, use the `config drives <panel_id>` utility to remove the drive from the ACSLS configuration.

**Note:**

`config drives` can be run while the library is online and mounts and dismounts to other drives are occurring.

### Removing Drives from the Media Validation Pool

After removing a drive from the media validation pool:

- For a nonpartitioned library, the drive slot is immediately available to ACSLS.
- For a partitioned library, a drive slot removed from the validation pool is not allocated to a partition. Use SL Console to assign the drive slot to a partition.

When the drive is removed from the media validation drive pool and available to the host, the library sends a Configuration Changed message to ACSLS.

If the drive is in the ACSLS configuration, vary the drive online. If the drive is not in the ACSLS configuration, add it with the `config drives <panel_id>` utility.

**Note:**

`config drives` can be run while the library is online and mounts and dismounts to other drives are occurring.

# 8

## Cartridge Management

ACSLS provides sophisticated cartridge management features. These features are provided in several ways:

- Automatically, such as the recovery of lost cartridges.
- Enabled by default, such as retaining information about absent and ejected cartridges.
- Customer defined, such as assigning volume attributes when cartridges are added to the database by audit or when they are entered through a CAP.

Using appropriate cartridge management features enhance the value that ACSLS provides.

Cartridge management consists of the following:

- [Populating the LSM](#)
- [Using the CAP](#)
- [Entering Cartridges](#)
- [Ejecting Cartridges](#)
- [Recovering CAPs](#)
- [Automatically Applying Policies to New and Reactivated Cartridges](#)
- [Cleaning Cartridges](#)
- [Managing Scratch Cartridges](#)
- [Using Absent and Ejected Cartridge Support](#)
- [Absent, Ejected and Missing Cartridges](#)
- [Cartridge Recovery](#)
- [Missing Cartridges](#)
- [Absent and Ejected Cartridge](#)
- [Using the Manual Volume Delete Utility](#)
- [Identifying Expired Cartridges](#)
- [Moving Least-Recently Accessed Cartridges from Active LSMs](#)
- [Manually Loading Cartridges Into Drives in a Disabled LSM](#)

### Populating the LSM

Cartridges can either be manually placed in cells when the library is offline, or entered into a library through a CAP.

An essential requirement for the library and ACSLS to function properly is the availability of a few free cells in each LSM to accommodate dismount, pass through, and eject operations. You should reserve at least one free cell for each tape drive installed in each LSM.

To determine the free cell count of an LSM, issue the following command:

```
query lsm lsm_id
```

In the SL8500, each rail is defined as an LSM.

## Using the CAP

The Following section reviews CAP Types, States, Modes and Priorities.

### CAP Types

Each type of CAP has a standard capacity and method for loading it with cartridges. An LSM may have more than one type of CAP. The following table shows the supported CAP types, identifiers and capacities, and loading methods.

**Table 8-1 CAP Types**

CAP Type	Identifier & Capacity	Loading Method
SL3000	CAP 6 and optionally CAP 1-5, CAP 7-10; each holds 26 cartridges.	13 cartridges are placed into each of 2 removable magazines which are loaded into the CAP.
SL4000	CAP 6 and optionally CAP 1-5, CAP 7-10; each holds 26 cartridges.	13 cartridges are placed into each of 2 removable magazines which are loaded into the CAP.
SL8500 Rotational	CAP 0 and optional CAP 1; each holds 39 cartridges.	13 cartridges are placed into each of 3 removable magazines which are loaded into the CAP.
SL8500 Bulk	CAP 0 and CAP 1; each holds 33 or 36 cartridges	11 or 12 cartridges are placed into each of 3 removable magazines which are loaded into the CAP. See <a href="#">Bulk CAP</a> .

### CAP States

A CAPs' state determines whether it is available for entering and ejecting cartridges. The following table describes the valid CAP states. See [Displaying CAP Information](#) for procedures for determining the CAP state. See the command, [query pool](#) for information about changing device states.



**Note:**

See [Understanding SL8500 Internal Addresses and ACSLS Addresses](#) for specifics regarding the SL8500 library.

**Table 8-2 CAP States**

State	Description	How requests are handled
online	Normal operating state.	All requests are accepted and processed.
offline	The CAP is logically disabled.	All requests are rejected.

Table 8-2 (Cont.) CAP States

State	Description	How requests are handled
<code>offline-pending</code>	Transition state. Occurs when the CAP is taken from online to offline.	All new requests are rejected. Current and pending requests are processed to completion.
<code>diagnostic</code>	The CAP is available for diagnostic activity without interference from client applications.	Requests from client applications are rejected. Requests from <code>cmd_proc</code> are processed.
<code>recovery</code>	Transition state. Occurs when the CAP is taken from offline to online.	New requests are rejected.

## CAP Modes

The CAP mode controls how a CAP will be used for cartridge enters and ejects. The following table describes the valid CAP modes. See [Displaying CAP Information](#) for procedures for determining the CAP mode. See the command, [query cap](#) for information about changing the CAP mode.

**HINT:** You cannot change a CAP mode while the CAP is in use. That is, if the door is open during either manual or automatic enter operations, you cannot change its mode until you complete the enter operation.

Table 8-3 CAP Modes

Mode	Description	Effects on enter/eject
<code>manual</code>	The CAP is locked when not in use. This is the initial mode for all multi-cartridge CAPs.	You can enter or eject cartridges only after explicitly issuing a command. You either specify the <code>cap_id</code> on the command, or allow ACSLS to automatically select a CAP, based on previously defined CAP priorities.  Some client applications require CAPs to be in manual mode. See the documentation for your tape management system.

Table 8-3 (Cont.) CAP Modes

Mode	Description	Effects on enter/eject
automatic	<p>The CAP is unlocked when not in use. This is the initial mode for all priority CAPs.</p> <p>You cannot set the CAP mode to automatic in a partitioned library. The exception to this is a dedicated CAP (assigned to only one partition), in the SL3000 which can be set to automatic mode.</p> <p>When the SL8500 access door is opened and closed, the SL8500 leaves the CAP locked. When the CAP is locked, it cannot be used for automatic mode enters</p>	<p>You can enter cartridges without explicitly issuing an <code>enter</code> command. The enter is initiated when you open the CAP door, place the cartridge(s) inside, and close the CAP.</p> <p>You cannot <code>cancel</code> an automatic enter operation that is in progress using the <code>cancel</code> command. To terminate an automatic enter in progress:</p> <p><i>If the CAP door is open</i>, remove all of the cartridges and close the door.</p> <p><i>If the CAP door is closed</i> and the cartridges are being moved into the library, allow the remaining cartridges to be entered into the library. The <code>enter</code> then terminates.</p> <p>To eject cartridges you must explicitly issue an <code>eject</code> command. You can either specify the <code>cap_id</code> on the command or allow ACSLS to automatically select a CAP, based on previously defined CAP priorities.</p> <p>If ACSLS shows the CAP in automatic mode, but it is locked and cannot be opened and used for automatic enters: synchronize ACSLS and the SL8500; then return the CAP to automatic enter.</p> <pre>set cap mode manual cap_id set cap mode automatic cap_id</pre>

## CAP Priorities

CAP priorities specify how ACSLS automatically selects CAPs when the CAP request specifies an asterisk (\*) for the CAP ID. The following table describes the CAP priorities and their effect. See [Displaying CAP Information](#) for procedures for determining the CAP priority. See [query cap](#) for information about changing the CAP priority.

Table 8-4 Cap Priorities

Priority	Effect
16 (highest)	first used
15 (next highest)	next used
-	-
1 (lowest)	last used
0	never automatically selected (initial priority for all CAPs)

CAP priorities and automatic CAP selection apply to the following commands:

- `audit`
- `eject`
- `enter`

- `venter`

When you enter any of these commands with an asterisk (\*) for all or part of the `cap_id`, ACSLS automatically selects an available CAP with highest nonzero priority for each ACS or LSM specified in the request.

**Examples:**

- `audit * server`  
ACSLs selects the highest non-zero priority CAP in each ACS.
- `enter 0,1,*`  
ACSLs selects the highest nonzero priority CAP in LSM 0,1.

## Displaying CAP Information

The following are some guidelines for using the `query cap` command to display current CAP information.

- To display information for selected CAPS, enter:  
`query cap cap_id cap_id ...`
- To display information for all CAPS in the library, enter:  
`query cap all`

## Entering Cartridges

You can choose to enter cartridges manually or automatically.

- To enter cartridges manually, you must issue the `enter` command. This unlocks the CAP so cartridges can be entered.
- An automatic enter is initiated by opening a CAP that is in automatic mode. When a CAP is in automatic mode, you need not issue an enter command.

The following steps describe the enter process:

1. Once you start the enter, the CAP is unlocked and reserved. It cannot be used by another host.
2. After you open the CAP, place the cartridges into the CAP and close the CAP. The CAP is now locked.

The ACSLS library robot inspects/audits the cartridges in the CAP. All cartridges being entered must have valid external labels that do not duplicate other `vol_ids` already managed by this ACSLS server.

 **Note:**

Virtual enter allows you to enter unlabeled cartridges into some libraries.

3. ACSLS allocates home cells in the library to valid cartridges and moves them to their assigned home cell locations.

Duplicate cartridges and cartridges without external labels are left in the CAP and must be removed.

4. Upon completion, the CAP is unlocked so more cartridges can be entered.
  - If the CAP is in automatic mode, the automatic enter is complete and the CAP is unreserved and available.
  - If this is a manual enter, the CAP is still reserved for the manual enter. To terminate the manual enter, cancel it with either the `cancel` command or with `Ctrl + c` at `cmd_proc` where the enter was started.

For additional information on the `enter` command, see "enter".



#### Note:

If cartridge tracing is enabled, the event log records all cartridge enters.

**Table 8-5 Cartridge Enter Commands**

Task	Command
Entering Cartridges in Automatic Mode	<code>set cap mode automatic <i>cap_id</i></code>
Entering Cartridges in Manual Mode	<code>enter <i>cap_id</i></code>
Entering Cartridges with Virtual Labels (venter)	<code>venter <i>cap_id vol_id vol_id</i></code> Do <i>not</i> open the LSM door and place cartridges with missing or unreadable labels in a storage cell because ACSLS cannot manage these cartridges. During an audit, ACSLS will eject cartridges with missing or unreadable labels that were placed in storage cells.

## Terminating an Enter Request

Use these procedures to terminate or cancel a current or pending manual enter or virtual enter.

You cannot cancel an automatic enter operation that is in progress using the `cancel` command. To terminate an automatic enter in progress:

- If the CAP door is open, remove all of the cartridges and close the door
- If the CAP door is closed and the cartridges are being moved into the library, you must allow the remaining cartridges to be entered into the library. The enter then terminates.

To cancel a manual enter:

1. Display all current and pending library activity:

```
query request all
```

2. Note the `request_id` of the enter/venter request you want to cancel.

3. From the `cmd_proc`, enter:

```
cancel request_id
```

Where `request_id` is the identifier of the request you want to cancel.



4. Wait for the CAP to unlock, open the CAP, and remove all cartridges.

`cmd_proc` displays a message indicating the number of cartridges entered into the library before the cancel request was received. These cartridges will remain under ACSLS control.

See [enter](#) .

## Ejecting Cartridges

To eject cartridges from the library you must issue the `eject` command.

The following steps describe the eject process:

1. Once you start the eject, the CAP is locked. It cannot be used by another host.
2. The robot places the specified cartridge(s) in the designated CAP, then ACSLS makes the cell locations where the cartridges were stored available for other cartridges.
3. Open the CAP, remove all cartridges from the CAP, and close the CAP. ACSLS then inspects the CAP to ensure it is empty. The CAP is now available for another operation, such as `enter` or `audit`.

If you specify more than a CAP full of cartridges on the `eject` command, empty the CAP when it fills, close the CAP, and ACSLS will continue the eject process until all cartridges are ejected.

For additional information on the `eject` command, see [eject](#) . Also refer to [ejecting.sh](#) .

If volume statistics gathering is enabled, the `acsss_stats.log` records all cartridge ejections. See [Setting General Product Behavior Variables](#).

## Recovering CAPs

This section discusses CAP recovery.

### General CAP Recovery Procedures

The following are general CAP recovery procedures.

#### Complete Enters and Ejects Before Doing CAP Recovery

If possible, let an `enter` or `eject` complete, instead of trying to cancel it and recover the CAP. This results in less complications, and less risk of a hung CAP.

- Let the entry of a CAP-full of cartridges complete, and then terminate a manual `enter` by cancelling it. (A CAP in automatic mode only enters a single CAP-full of cartridges at a time.)
- If possible, let all the cartridges specified in the `eject` command be ejected. Otherwise, let ACSLS eject a full CAP of cartridges and empty the CAP before trying to cancel to eject.

#### Recover a Hung CAP by Varying it with Offline Force Then Online

You must vary a CAP offline with force to recover it. Varying a CAP offline force and then back online both recovers the CAP and usually terminates a hung `enter` or `eject` that is using the CAP.

1. Vary the CAP offline with force.

```
vary cap cap_id offline force
```

Only the current robotic request completes, then the CAP goes offline immediately. Pending requests are discarded, and new requests are rejected.

A hung manual `enter` or `eject` is usually canceled.

2. Cancel the `enter` or `eject` request if it is still active.

To see if the `enter` or `eject` request is still active:

```
query request all
```

If the `enter` or `eject` is still active, cancel it by entering the following command:

```
cancel request_id
```

3. Vary the CAP back online.

```
vary cap cap_id online
```

This should recover the CAP and make it available for other requests.

## Recovering a CAP After Opening the Access Door

ACSLs now unlocks a CAP in automatic enter mode after an SL8500 or SL3000 access door is opened and closed, or the SL8500 or SL3000 re-initializes.

After an SL8500 or SL3000 library re-initializes, if the CAP is locked and you must recover it, follow the appropriate procedure below to recover the CAP.

### CAP Used for Automatic Enter Will Not Unlock

To recover a CAP that will not unlock for an automatic enter, you must synchronize the state of the CAP between both ACSLS and the library.

1. Set the CAP mode to manual to terminate automatic enter mode:

```
set cap mode manual cap-id
```

2. Set the CAP back to automatic mode:

```
set cap mode automatic cap_id
```

### CAP Used for Manual Enter Will Not Unlock

To recover a CAP that will not unlock for a manual enter, you must synchronize the state of the CAP between both ACSLS and the library.

1. Vary the CAP offline with force.

```
vary cap cap_id offline force
```

2. Vary the CAP back online.

```
vary cap cap_id online
```

3. Restart your manual enter.

```
enter cap_id
```

## CAP Used for Ejects Will Not Unlock

To recover a CAP where you were doing an eject, you must remove any cartridges left in the locked CAP, and synchronize the state of the CAP between both ACSLS and the library.

1. Remove any cartridges in the CAP.

- a. Vary the CAP offline with force.

```
vary cap cap_id offline force
```

- b. Vary the CAP back online.

```
vary cap cap_id online
```

2. Select one of the following:  
If the CAP is in auto mode:

- a. Set the CAP mode to manual to terminate the automatic enter mode.

```
set cap manual cap_id
```

- b. Set the CAP to automatic mode. This unlocks the CAP.

```
set cap mode automatic cap_id
```

- c. Open the CAP and remove any cartridges left in the CAP.

If the CAP is not in auto mode:

- a. Start a manual enter.

```
enter cap_id
```

- b. Remove any cartridges left in the CAP.

- c. Cancel the enter.

Use either `Ctrl + c` in the `cmd_proc` that is waiting for the enter, or cancel the enter request ID.

3. Restart your eject.

```
enter cap_id vol_id|volrange...
```

## Automatically Applying Policies to New and Reactivated Cartridges

This section discusses how to automatically apply policies to new and reactivated cartridges.

### Automatic Assignment of Cleaning Cartridge Attributes

Modern cleaning cartridges are labeled with media types that are reserved for only cleaning cartridges. For example, T10000 backward-compatible cleaning cartridges are labeled with a media domain and type of "CL", and LTO universal cleaning cartridges are labeled "CU".

Because ACSLS understands that cartridges with these media domains and types can only be cleaning cartridges, it automatically sets cleaning cartridge attributes when these cartridges are added by audit, enter, or Cartridge Recovery. This includes identifying them as cleaning cartridges and setting their maximum cleaning usage.

## watch\_vols Policies

The `watch_vols` utility can automatically assign attributes to cartridges added to the database or reactivated by audit, also when they are entered or re-entered. The policies are specified in the `vol_attr.dat` file and are selected by `vol_id` or `vol_range`. This utility can automatically:

- Assign volume ownership based on `vol_id` ranges or specific volumes listed in a `vol_attr.dat` policy table.
- Assign cartridges to a scratch pool.
- Move new and reactivated cartridges to a specific LSM.
- Assign cartridges to a logical library.

For more information, see [watch\\_vols](#).

## Cleaning Cartridges

Tape drives must be cleaned periodically to remove stain contamination and adherent debris from the read/write recording heads. Drive control units track how much tape passes through each drive and sends a message to ACSLS when a drive requires cleaning.

For more information on cleaning cartridges, see:

- [Automatic Cleaning by ACSLS](#)
- [Cleaning Cartridge Maximum Usage](#)
- [Entering Cleaning Cartridges](#)
- [Ejecting Used Cleaning Cartridges](#)
- [Manually Cleaning a Drive](#)
- [Cleaning Cartridges on Fibre-Attached Libraries](#)

## Automatic Cleaning by ACSLS

ACSLS can perform automatic cleaning for TCP/IP attached libraries (SL8500 and SL3000), but not for Fibre or SCSI-attached libraries (SL150).

If auto-cleaning is enabled, ACSLS automatically mounts a cleaning cartridge on the tape drive when required. Automatic cleaning is enabled when the `AUTO_CLEAN` dynamic variable is set to `TRUE` (default).

Modern tape drives request cleaning as needed. The drive notifies the library, which forwards the message to ACSLS. ACSLS records that the drive must be cleaned. When ACSLS processes the next mount request for the drive, it precedes the mount with a cleaning operation. This includes selecting a compatible cleaning cartridge, mounting the cleaning cartridge, dismounting the cleaning cartridge, and then proceeding to mount the data cartridge specified in the original mount request.

If ACSLS encounters a recoverable problem during the cleaning operation, such as mounting a spent (used-up) cleaning cartridge, it selects another cleaning cartridge and retries the cleaning operation. The `AUTO_CLEAN_RETRY_LIMIT` dynamic variable

governs the number of retries, with a default of one retry and a range of 0-5 retries. Use `acsss_config`, and select General Product Behavior Variables to display and change this variable.

The `UNIFORM_CLEAN_USE` dynamic variable defines the method used to select cleaning cartridges. Options, are:

- `VOLID_SORT` – Order by `vol_id`. Uses up one cleaner before using the next.
- `LEAST_USED` – Order by usage. Spreads usage evenly.
- `MOST_CAPACITY` – Order by usage left. Uses up all cleaners at the same time.

The default is `VOLID_SORT`. Use `acsss_config`, and select General Product Behavior Variables to display and change this variable.

For more information on automatic cleaning by ACSLS, see:

- [Cleaning Cartridge Maximum Usage](#)
- [Entering Cleaning Cartridges](#)
- `AUTO_CLEAN` in [Setting CSI Tuning Variables](#)
- `AUTO_CLEAN_RETRY_LIMIT` in [Setting CSI Tuning Variables](#)

## Cleaning Cartridge Maximum Usage

Each different cleaning cartridge type has a maximum number of uses before the drive reports that it is used-up (expired or spent). This maximum usage varies depending on the type of cleaning cartridge. When ACSLS adds cleaning cartridges, the cartridges' max usage is recorded in the ACSLS database. ACSLS only selects cleaning cartridges for automatic cleaning when the cartridges' `access_count` (number of times the cartridge has been mounted, and so forth) is less than the `max` usage. When the tape drive reports that a cleaning cartridge is used-up (spent), ACSLS sets the access count to be greater than `max` usage.

The maximum usage that ACSLS automatically sets for cleaning cartridges is higher than the actual number of cleaning uses that the cartridge supports. This is because some applications schedule mounts of cleaning cartridges without the drives having requested cleaning. If the drive is not ready to be cleaned, it may do a "pseudo-clean" to avoid wearing out the heads prematurely. This means that the drives' access count has been incremented without actually using the cleaning cartridge. A higher `max` usage value allows these cartridges to be used until the drive reports that they are spent.

## Manually Defining Cleaning Cartridges

You can use the `set clean` command to define a cleaning cartridge and set its maximum usage.

```
set clean max_usage vol_id | volrange
```

Where:

- `max_usage` is the number of times a cleaning cartridge is used before ACSLS stops selecting the cartridge to clean cartridges.
- `vol_id | volrange` specifies the cleaning cartridge or range of cartridges.

Use `set clean` to:

- Change a cleaning cartridge's maximum usage count.

For example the cleaning cartridge has been manually mounted on drives that did not need to be cleaned, and the `access_count` was incremented but only a "pseudo-clean" was done. Set a higher `max_usage` to get full usage from the cleaning cartridge.

```
set clean max_usage vol_id volrange
```

- Set a cartridge's cleaning cartridge attribute off. For example, if you incorrectly defined a data cartridge as a cleaning cartridge, set the cartridge's cleaning cartridge attribute off to redefine the cartridge as a data cartridge.

```
set clean off vol_id volrange
```

## Monitoring Cleaning Cartridges

You must monitor the cleaning cartridges in your library by ejecting the used cleaning cartridges. Enter new cleaning cartridges as needed.

- To display all cleaning cartridges:

```
query clean all
```

- To display all cleaning cartridges of one `media_type` in an ACS, use the display command:

```
display volume * -home acs ,*,*,*,* -media media_type
```

- To display a cartridges' maximum cleaning usage and current usage:

```
display volume * -home acs ,*,*,*,* -media media_type
```

```
display volume * -home acs,*,*,*,* -media media_type -f vol_id acs lsm media  
max_use access_count
```

- To display all cleaning cartridges in an ACS, with the maximum cleaning usage and current usage:

```
display volume CLN* -home acs ,*,*,*,* -f acs lsm type media max_use  
access_count
```

- To display all used cleaning cartridges (these cartridges should be ejected and replaced with new cleaning cartridges):

```
display volume * -spent_clean
```

See also:

- [Entering Cleaning Cartridges](#)
- [Ejecting Used Cleaning Cartridges](#)

## Entering Cleaning Cartridges

When entering cleaning cartridges, make sure to complete the following procedure:

- Use cleaning cartridges whose media types are compatible with the drive types in your library. ACSLS automatically selects the correct type of cartridge for each cleaning operation.

To see which cleaning cartridges are compatible with a drive type, see the media-to-drive compatibility table in the *ACSL S Product Information* manual or use the `drive_media.sh` utility.

- Define at least a few cleaning cartridges for each drive type in your library. For most sites, having at least one cleaning cartridge for every four drives is reasonable.

To define cleaning cartridges to ACSLS:

1. Make the CAP ready for entry:  
See [Entering Cartridges](#) for more information.
2. Enter the cleaning cartridges.

`cmd_proc` displays messages with the cartridge IDs of the cartridge you entered.

As described in [Automatic Assignment of Cleaning Cartridge Attributes](#), ACSLS automatically defines cleaning cartridges when they are entered or added by audit, enter, or Cartridge Recover. This includes their maximum usage.

## Ejecting Used Cleaning Cartridges

ACSL S logs messages to the Event Log when a cleaning cartridge has reached its maximum usage or a drive reports that the cleaning cartridge is spent. ACSLS leaves the cartridge in the library, but no longer selects it for cleaning. You must eject the used cleaning cartridge and enter a replacement.

To eject used cleaning cartridges:

1. Use `query clean` and `display volume` to identify cleaning cartridges that are over their maximum usage or spent.

```
query clean all
display volume * -spent_clean
```

2. Eject the cleaning cartridges.

```
eject cap_id vol_id | volrange
```

Where:

`cap_id` specifies the CAP used to eject the cleaning cartridges.

`vol_id | volrange` specifies the IDs of the cleaning cartridges to eject.

3. Remove the spent cleaning cartridges.

See [Monitoring Cleaning Cartridges](#)

## Manually Cleaning a Drive

Use this procedure to clean a drive when auto-clean is disabled or is not working.

To manually clean a drive:

1. Determine which cleaning cartridge types are compatible with the drive to be cleaned.

Refer to the *Product Information Guide* and look in the Drive and Media Compatibility table for a list of the cleaning cartridges for each drive type.

2. Display the available cleaning cartridges:

```
query clean all
```

To display all compatible cleaning cartridges in the same ACS as the drive, use the `display` command:

```
display volume * -home acs ,*,*,*,* -media media_type
```

To display the cartridges' maximum cleaning usage and current usage:

```
display volume * -home acs ,*,*,*,* -media media_type -f vol_id acs lsm
media max_use access count
```

To display all cleaning cartridges in an ACS, with their maximum cleaning usage and current usage:

```
display volume CLN* -home acs ,*,*,*,* -f acs lsm type media max_use
access_count
```

3. Select a compatible cleaning cartridge from those listed and mount it on the drive:

```
mount vol_id drive_id
```

4. After the drive is cleaned and the cleaning cartridge is unloaded, dismount the cleaning cartridge:

```
dismount vol_id drive_id
```

## Cleaning Cartridges on Fibre-Attached Libraries

ACSLs automatic cleaning is not supported for drives in Fibre-attached libraries. You can only clean these drives using ACSLS by manually mounting a cleaning cartridge. However, Fibre-attached libraries let you enable auto-cleaning using the library GUI. For more information, see your library documentation.

## What To Do When Drives Are Not Being Cleaned

The following are some troubleshooting tips to try when drives are not being cleaned.

### Verify Auto-Cleaning is Enabled

If auto-cleaning is disabled, ACSLS logs a message in the event log and displays cleaning messages with `cmd_proc` when the drive requires cleaning. You must manually mount a cleaning cartridge.

Use `acsss_config` to enable or disable auto-cleaning. In addition, with `acsss_config` you can specify how cleaning cartridges are ordered for selections and queries.

Automatic cleaning is enabled when the `AUTO_CLEAN` dynamic variable is set to **TRUE** (on), which is the default setting. To view `AUTO_CLEAN`, enter:

```
dv_config -e AUTO_CLEAN
```

ACSLs does not perform automatic cleaning for Fibre-attached libraries.

### Verify there are Cleaning Cartridges for the Drives

If all cleaning cartridges are expired (the `max_usage` value is exceeded) or have been reports as spent by the drives, ACSLS performs the original mount request without cleaning the drive. For that mount and for each subsequent mount to the uncleaned



drive, ACSLS posts message 376 N "Drive *drive\_id*: No Cleaning cartridge available" to the event log. Add more cleaning cartridges that are compatible with the drive type as described in [Manually Defining Cleaning Cartridges](#).

If drives are not being cleaned, make sure that there are cleaning cartridges for the drives in the library and that they still have usage left.

From `cmd_proc`, you can use the display command to see:

- All of your cleaning cartridges and their usage:

```
display volume * -clean -f media access_count max_use
```

- All of the volumes of a specific media type.

For example, to display all LTO cleaning cartridges:

```
display volume * -media LTO-CLNU -f access_count max_use
```

- All used-up (spent) cleaning cartridges and their usage:

```
display volume * -spent_clean -f media access_count max_use
```

## Verify Auto-Cleaning is Disabled Using SL Console for an SL8500 or SL3000 Library

If you ever have an issue where automatic cleaning for an SL8500 or SL3000 is **not** working, check to make sure Automatic Cleaning is *not* enabled for the library using SL Console.

When automatic cleaning is enabled using ACSLS, it automatically mounts a cleaning cartridge before the next mount when it receives a "drive needs cleaning" message from the library after a dismount.

If automatic cleaning is enabled at the library level using SL Console, the library is responsible for doing automated cleaning. When library auto-cleaning is enabled, the library does *not* send a drive needs cleaning message to ACSLS. ACSLS never knows that the drive must be cleaned. The library then tries to mount a cleaning cartridge from one of its system cells to clean the drive before sending ACSLS the dismount response.

As a result, there can be a mix-up where the library is trying to do automated cleaning, but has no cleaning cartridges in the system cells. ACSLS can be managing cleaning cartridges in normal storage cells, but ACSLS does not receive a "drive needs cleaning" message. The result is that the drive is not cleaned.

To resolve this:

- If ACSLS automated cleaning is enabled, but drives are not being cleaned, check if the library also has auto cleaning enabled.
- If auto cleaning is enabled at the library, disable it using the SL Console.

Use the SL Console or the library operator panel.

1. Select the System Detail tab.
2. Select Library.
3. Select the Auto Clean tab.
4. Select the Configure tab.
5. Check if Auto Cleaning is enabled for this partition (or "Partition 1 or None").
6. Disable automated cleaning if it is enabled.

## Check if Cleaning Cartridges are Marked as Questionable

Automatic cleaning does not select questionable cartridges to avoid repeatedly selecting a faulty cleaning cartridge. Cartridges are marked as questionable if the library reports that the cartridge has an unreadable label.

You can identify cleaning cartridges that are marked as questionable with the `display` command. This also displays the cleaning cartridges' ACS, LSM, type, `max_use` and `access_count`.

```
display volume CLN* -f media_status acs lsm media_status type max_use  
access_count
```

To clear a questionable status:

- Eject the cartridge, examine it, and if it is good, enter it back into the library.
- Questionable status is cleared when a cartridge is entered.

## Managing Scratch Cartridges

Scratch cartridges either contain no data or data that can be overwritten. A user or application mounts a scratch cartridge to write new data on that cartridge.

To assign scratch status:

- Cartridges can be defined as scratch cartridges and assigned to scratch pools by the `set scratch` command.
- The `watch_vols` utility can automatically assign cartridges to a scratch pool based on the cartridges' `vol_id` or `volrange`. See "[watch\\_vols](#)".

Clearing a volume scratch status:

- A cartridge's scratch status is cleared when the cartridge is successfully mounted (either by a mount scratch or a normal mount request).

### Note:

The `set scratch` command can be used to clear scratch status. Although a volume's scratch status is cleared when a volume is mounted, the `pool_id` is not cleared. The result is a data volume assigned to the pool.

The `set scratch` command can also be used to assign a data volume to a scratch pool using:

```
set scratch off pool_id vol_id | volrange
```

You must ensure that the library has enough available scratch cartridges to satisfy scratch mount requests. For more information see:

- [Adding Scratch Cartridges to the Library](#)
- [Rebalancing Scratch Pools](#)

The following sections provide additional information about managing scratch cartridges and scratch pools:

- [Displaying Scratch Pool and Scratch Cartridge Information](#)
- [Deleting Scratch Pools](#)
- [Mounting Scratch Cartridges](#)
- [Unscratching Cartridges](#)

## Displaying Scratch Pool and Scratch Cartridge Information

To display scratch pool information, use the following ACSLS functions:

- `query pool`  
Displays scratch pool attributes. See [query pool](#)
- `query scratch`  
Displays scratch cartridge information. See [query scratch](#)
- `query mount *`  
Displays the status of media-compatible cartridges for a specified scratch pool (and, optionally, for a specific cartridge media type within the pool). See [query mount \\*](#)
- Customized volume report  
Reports selected scratch volume information. See [Creating a Logging Volume Statistics Report](#)

## Adding Scratch Cartridges to the Library

Use this procedure to add scratch cartridges to the library.

To add scratch cartridges to the library:

1. If necessary, create a new scratch pool:  
For more information, see [query scratch](#) .
2. Enter scratch cartridges into the library.  
For more information, see [Entering Cartridges](#) .
3. Define the cartridges you entered in Step 2 as scratch cartridges and assign them to a scratch pool.  
This can either be done using policies defined in `vol_attr.dat` for the `watch_vols` utility or using `set scratch` .

## Rebalancing Scratch Pools

Use this procedure to rebalance scratch pools by moving scratch cartridges from one pool to another.

To rebalance scratch pools:

1. Display the attributes of all scratch pools:  
`query pool all`

For more information see [query pool](#) .

2. Use the `query scratch` command to display the IDs of scratch cartridges in pools you want to rebalance.

For more information, see [query scratch](#) .

3. Use the `set scratch` command to move scratch cartridges from one pool to another.

For example, to reassign cartridges YUMA20 through YUMA80 (which currently reside in pool 5) to pool 10, enter:

```
set scratch 10 YUMA20-YUMA80
```

For more information, see [set scratch](#) .

## Deleting Scratch Pools

To manage scratch pools, you may want to delete any scratch pools that no longer contain scratch cartridges. You cannot delete the common pool (Pool 0). Notice that you can delete only empty scratch pools; you cannot delete a scratch pool if it contains either data or scratch cartridges. You can, however, use [Deleting All Empty Pools](#) to delete all empty pools (ACSLs will not delete any pools that contain scratch or data cartridges).

## Emptying a Scratch Pool

Use this procedure to empty a scratch pool before deleting it.

To empty a scratch pool:

1. To move data cartridges out of the pool, enter:

```
set scratch off 0 vol_id volrange ...
```

Where the `vol_id` or `volrange` specifies the data cartridges you want to move to the common pool (pool 0). For more information, see [set scratch](#) .

2. To move scratch cartridges out of the pool, do one of the following:
  - Move the cartridges to another pool.
  - See [Ejecting Cartridges](#) . If you eject scratch cartridges, however, ACSLS no longer manages these cartridges. If you later want to use these cartridges, you must reenter them and assign them to a scratch pool.

## Deleting a Single Pool

To delete a single pool:

```
delete pool pool_id
```

## Deleting All Empty Pools

The `delete pool all` command deletes only empty scratch pools, not pools that contain scratch or data cartridges.

To delete all empty pools:

```
delete pool all
```

## Mounting Scratch Cartridges

The `mount scratch (mount * using cmd_proc)` command selects a scratch cartridge that is compatible with, and as close as possible to, the specified drive and mounts it in the drive. If a pool is specified, the scratch cartridge must be assigned to that pool.

If the mount of the scratch cartridge fails because the scratch cartridge is not found in its home cell or another recoverable error, ACSLS automatically tries to select another scratch cartridge and retry the mount.

You can set a Volume Access Control policy for whenever a scratch cartridge is mounted, it is automatically owned by the ACSAPI user who mounted it. See [Establishing Ownership of Your Volumes](#).

Use the following procedures to mount scratch cartridges in single-media and mixed-media environments.

### Single-Media Environments

- To mount a cartridge from a specified pool:

```
mount * drive_id pool_id
```

If no cartridge is available from the specified pool and the pool has been set for "overflow," ACSLS selects a cartridge from the common pool (pool 0).

- To mount a cartridge from the common pool:

```
mount * drive_id
```

### Mixed-Media Environments

- To mount a scratch cartridge with a specified media type from a specified pool:

```
mount * drive_id pool_id media media_type
```

If no cartridge is available from the specified pool and the pool has been set for `overflow` ACSLS will select a cartridge with the specified media type from the common pool (pool 0).

- To mount a scratch cartridge from a specified pool with a media type determined by scratch preferences:

```
mount * drive_id pool_id media *
```

If no cartridge is available from the specified pool and the pool has been set for `overflow` ACSLS will select a cartridge from the common pool (pool 0) according to the defined scratch preferences.

- To mount a cartridge from the common pool with a specified media type:

```
mount * drive_id media media_type
```

- To mount a cartridge from the common pool with a media type determined by scratch preferences:

```
mount * drive_id media *
```

## Unscratching Cartridges

Scratch cartridges are automatically reassigned to data cartridge status when they are mounted.

Use this procedure to “unscratch” cartridges (return them to data cartridge status) that were scratched in error.

To unscratch cartridges:

1. Use the `query pool` and `query scratch` commands to display the cartridge and pool IDs of the cartridges to unscratch.

For more information see [query pool](#) and [query scratch](#) .

2. To unscratch the selected cartridges, enter:

```
set scratch off 0 vol_id volrange.
```

Where the `vol_id` or `volrange` specifies the cartridges you want to change from scratch mode and move them to the common pool (pool 0). For more information, see [set scratch](#) .

## Using Absent and Ejected Cartridge Support

Absent cartridge support in ACSLS marks cartridges that cannot be found in the library as absent, instead of deleting them. If these cartridges are later found in the library, ACSLS changes them to active status instead of re-adding them to the database. Reactivation preserves the access count and settings, such as pool, volume access control ownership, and locks.

Similarly, ejected cartridge support retains cartridge information when cartridges are ejected. The cartridges are reactivated when they are re-entered.

Absent and ejected volume support is enabled when the `ABSENT_VOLUME_RETENTION_PERIOD` is set to a nonzero number of days. The default value is 5 days.

Additional aspects of absent and ejected cartridge support include:

- The Manual Volume Delete (`del_vol`) utility will retain volumes as absent unless the `-d` option is specified. If this option is specified, the volume is deleted without waiting for the expiration of an absent or ejected status.
- ACSLS queries SL3000 and SL8500 libraries for the location of lost cartridges.
- ACSLS improves volume recovery by searching for volumes that are not found in their expected locations in the library. ACSLS searches all recorded locations instead of automatically deleting the volume.
- Clients can specify, through the `ENABLE_STATUS_VOLUME_ABSENT` and `ENABLE_STATUS_VOLUME_MISSING` configuration settings, whether they want absent, ejected, and missing statuses to be reported through the ACSAPI.
- The `volrpt` utility with the `-i` option will report volume records with a status of absent or ejected. By default, `volrpt` does not report absent or ejected volumes.

# Absent, Ejected and Missing Cartridges

ACSLs reports three cartridge (volume) statuses:

- `missing`

The cartridge cannot be located in the library, and at least one recorded location for the cartridge cannot be searched because the LSM is offline or a drive is not communicating. The information about the cartridge has been retained.

- `absent`

The cartridge cannot be located in the library. All recorded locations for the cartridge have been searched, and the cartridge is not in any of them. The information about the cartridge is retained. If the cartridge is found or re-entered into the library (before the retention period expires), it is reactivated.

- `ejected`

The cartridge was ejected. The information about the cartridge is retained, and if the cartridge is found or re-entered (before the retention period expires), it is reactivated.

## Cartridge (Volume) Status Reporting

ACSLs reports cartridges (volumes) with the status "missing", "absent," or "ejected," differently in response to ACSLS commands than it does in response to ACSAPI requests.

The information displayed in response to ACSLS commands identifies a cartridge as "missing", "absent", or "ejected."

However, the cartridge status information ACSLS displays in response to ACSAPI requests is governed by the following ACSLS dynamic variables:

- `missing`

- If the ACSLS dynamic variable `ENABLE_STATUS_VOLUME_MISSING` is TRUE, ACSLS reports: `STATUS_VOLUME_MISSING`.
- If the ACSLS dynamic variable `ENABLE_STATUS_VOLUME_MISSING` is FALSE, ACSLS reports: `STATUS_VOLUME_IN_TRANSIT`.

- `absent`

- If the ACSLS dynamic variable `ENABLE_STATUS_VOLUME_ABSENT` is TRUE, ACSLS reports: `STATUS_VOLUME_ABSENT`
- If the ACSLS dynamic variable `ENABLE_STATUS_VOLUME_ABSENT` is FALSE, ACSLS treats the volume as if it had been deleted from the ACSLS database and reports: `STATUS_VOLUME_NOT_IN_LIBRARY`.

- `ejected`

- If the ACSLS dynamic variable `ENABLE_STATUS_VOLUME_EJECTED` is TRUE, ACSLS reports: `STATUS_VOLUME_EJECTED`
- If the ACSLS dynamic variable `ENABLE_STATUS_VOLUME_EJECTED` is FALSE, ACSLS treats the volume as if it had been deleted from the ACSLS database and reports: `STATUS_VOLUME_NOT_IN_LIBRARY`.

The `ABSENT_VOLUME_RETENTION_PERIOD` dynamic variable controls how long absent and ejected volumes are retained in the ACSLS database and specifies the number of days these volumes are retained. There are two special values:

- The value 0 (zero) days specifies that volumes are deleted and will not be marked absent or ejected. (This is the behavior of ACSLS releases previous to ACSLS 6.1.)
- The value 999 days specifies that absent and ejected volumes are retained forever in the database.

## Cartridge Recovery

Cartridge Recovery (`acscr`) is an ACSLS internal process that is called to resolve discrepancies whenever the actual content of a storage cell or tape drive does not match the information saved in the ACSLS database. It does this by:

- Letting the library examine the volume's home cell and, possibly, the drive. Then, updating the ACSLS database with the results.
- Letting ACSLS (with the SL3000 and SL8500 libraries) to recover a cartridge by asking the library where a cartridge is located, and then using the library's reply to update the ACSLS database.

If Cartridge Recovery finds a discrepancy, such as a cartridge that is recorded at another location, it creates another recovery request and adds it to its request queue. (This is called a "cascade".)

Other processes pass recovery requests to Cartridge Recovery when they encounter a discrepancy between the ACSLS database and the actual contents of the library. Cartridge Recovery is then the central location where cartridges are marked as missing, changed to absent, and reactivated. Thus, what appears to be the behavior of many other ACSLS commands and utilities is actually done by Cartridge Recovery when it updates the database to match the information reported by the library.

When other processes pass recovery requests to Cartridge Recovery, you can:

1. Continue and let Cartridge Recovery continue asynchronously (Cartridge Recovery proceeds independently).
2. If you need a specific cartridge that is lost, wait for Cartridge Recovery to finish processing this recovery request and report what it found.

## Missing Cartridges

A cartridge is marked as missing when:

- Cartridge Recovery cannot find a cartridge in the library.
- It cannot examine all recorded locations for a cartridge (home cell and drive, if the cartridge has a recorded drive location).

For example, when Cartridge Recovery cannot examine the home cell in an offline LSM or an offline drive, and when it does not find the cartridge in other locations, it marks the cartridge missing.

Cartridge Recovery preserves the cartridge's home location unless it examines the cartridges' home cell and finds another cartridge there. In this situation, it marks the cartridge "homeless," with a minus one (-1) in the `home_lsm` field.



When Cartridge Recovery finds a cartridge that was missing, it changes that cartridges' status to "home" or "in drive" in the database, depending where it found the missing cartridge.

- If the cartridge is found in a cell other than its recorded home cell, Cartridge Recovery checks the cartridges' home cell to see whether it has found a duplicate cartridge.
- If the cartridge is not in its recorded home cell, Cartridge Recovery records the cell in which it was found as its new home cell.
- If the new cartridge is a duplicate, Cartridge Recovery reports this on the Event Log. The duplicate cartridge is *not* ejected.
- If Cartridge Recovery finds a "homeless" cartridge in a drive, it does not assign a new home cell. When the cartridge is dismantled, the dismant process assigns a new home cell.

## Absent and Ejected Cartridge

This section discusses absent and ejected cartridges.

### Cartridges Not Found

When Cartridge Recovery can examine all recorded locations and it cannot find a cartridge:

1. If the `ABSENT_CARTRIDGE_RETENTION_PERIOD` is 0, Cartridge Recovery:
  - Deletes the cartridge record from the database.
  - Marks the cell record in the database for the cell that was the cartridges' home cell as "empty."
2. If the `ABSENT_CARTRIDGE_RETENTION_PERIOD` is greater than 0, Cartridge Recovery:
  - Changes the status of the cartridge record in the database to "absent" if the cartridge has not already been marked absent or ejected.
  - Records the cartridge as "homeless" (with a minus one (-1) in the `home_lsm` field).
  - Marks the cell record in the database of the cartridges' former home cell as "empty."

### Cartridges Found

If Cartridge Recovery finds an ejected or absent cartridge, it reactivates the cartridge.

If the ejected or absent cartridge is found in a storage cell, this becomes its new home cell, and Cartridge Recovery changes the cartridges' status to "home" in the database.

If the cartridge is found in a drive, ACSLS assigns a new home cell when the cartridge is dismantled.

## Using the Manual Volume Delete Utility

The Manual Volume Delete utility, `del_vol`, allows you to access a volume that is in an offline, unavailable LSM. If you manually remove the cartridge from the LSM and try to re-enter it into another LSM, ACSLS will issue a `duplicate volume` message and will not enter the cartridge. Using the `del_vol` utility, you can first delete the volume from the database, then manually remove it from the offline LSM and successfully re-enter it into an online LSM.

The `del_vol` utility now retains volumes as absent, with the option to delete the volume. The volume can be deleted without waiting for the expiration of an absent or ejected status.

 **Note:**

To remove a cartridge from an online LSM, issue an `eject` command for the cartridge. If the cartridge is not actually in the LSM, you can run `del_vol` with the `-f` (force option). ACSLS and the database must be up and running to use this utility. Do not run `del_vol` while the system is in recovery; unpredictable results may occur.

For more information about this utility, see [del\\_vol](#).

To delete a cartridge using the `del_vol` utility:

1. Log in as `acsss`.
2. Delete the cartridge:

```
del_vol vol_id
```

## Identifying Expired Cartridges

When tape cartridges exceed their design life, the media can wear thin, and mechanical parts, like the cartridges' gate wearing out. When cartridges reach the end of their engineered life, you should consider migrating the data on them to new cartridges and retiring the old cartridges. This avoids the slight risk of either mechanical cartridge components failing or the data being unreadable.

A cartridges' chronological age and its usage are different. Some 9840 cartridges have been in use for a decade, but they have different usage patterns. Some have been used daily while others are used for deep archive and are rarely accessed. Identifying the cartridges that are past their design life is critical.

To identify the cartridges that need to be retired, you must determine their usage. A cartridge's usage is recorded in the cartridge's directory, and before a cartridge is dismounted, the drive updates the directory.

For cartridges in libraries controlled by ACSLS:

- For some libraries that ACSLS manages, cartridge usage is displayed as a percentage of "warranty life" and "end-of-life".
- For prior ACSLS releases and libraries, the ACSLS `access_count` can be displayed with the `display` command and the `volrpt` utility.

## Cartridge End-of-Life Percentages

With modern libraries running the latest firmware and StorageTek drives running the latest firmware, the tape drive reports the cartridge's "end-of-warranty-life" and "end-of-life percentages" to the library when the cartridge is dismounted. Then, the library reports this to ACSLS. ACSLS saves this information in its database, and you can see it by running the ACSLS `display volume` command. See [Using display Command Options](#).

**Example:** To display all T9840 cartridges, with ACS, LSM, media, and `end_of_life` information, sorted by `end_of_life`:

```
display volume * -media STK1R -f acs lsm media end_of_life warranty_life -s end_of_life
```

Specifically this information is reported to ACSLS for these libraries and drives:

Libraries:

- SL3000
- SL8500 (with the 4.10 firmware)

Tape Drives:

- All T10000 tape drives - with 1.38 firmware
- T9840C and T9840D – with 1.42 firmware

## Access Counts

In many cases, cartridge end-of-life reporting is not available. In these cases, the ACSLS `access_count` is the best information available. The ACSLS database records the number of times that volumes have been selected or accessed. This can be used to estimate the number of times that they have been mounted, if the cartridge has stayed in a group of connected libraries (within an ACS).

This information is collected regardless of library type, so it is maintained for SL8500s and SL3000s. ACSLS has saved this information for decades, so even if you are still on down-level releases, you still have this information. However, this data has limits. The biggest one is that when a cartridge is entered into a library, the count is set to zero (0).

Information is retained about volumes for the *retention period* you have set, so counts are preserved when a cartridge is ejected from an ACS and re-entered into the same or different ACS within the retention period. The default retention period is five days. However, if a volume is ejected from a library and remains off-site for longer than the retention period for volume information, the information about the volume is deleted from the ACSLS database.

For cartridges that have remained in a single library these ACSLS access counts are very useful. For a T9840 cartridge, if the ACSLS `access_count` is over 11,000, the cartridge in question is near the end of its life, if not already over its end of life. T10000 cartridges have an end-of-life value of 16,000 mounts.

Because ACSLS provides tools so you can preserve and migrate your database information when you install new releases of ACSLS, this information can go back over a decade. In the absence of data from the cartridge, this is the only option.

## ACSLs Cartridge Mount Count Details

The ACSLS field is called `access_count`. It counts:

- Mounts (dismounts are not counted)
- Enters and ejects (enters and ejects are often rare)
- Moves (although the `move` command using `cmd_proc` is rarely used, and it is not available to ACSAPI clients)

The `access_count` is mainly a count of the number of times the cartridge was mounted. ACSLS remembers ejected volumes for the `ABSENT_VOLUME_RETENTION_PERIOD` (default of 5

days). When cartridges are moved between ACSs and sent off-site and brought back on-site, ACSLS can remember the `access_count`.

You can see the ACSLS `access_count` using both:

- The ACSLS `display` command.

To see all 9840 data cartridges, sorted by access count, and also showing media type, ACS, and LSM:

```
display volume * -media STK1R -s access_count -f media access_count acs lsm
```

- The `volrpt` utility.

`volrpt` can be sorted by use (`access_count`), and can only include selected fields. For example, a custom `volrpt` containing `vol_id`, `media type`, `access_count`, and `location`, can be output to a flat file for further processing by a script.

## Cartridge Warranty and End-of-life Thresholds

The warranty and end-of-life thresholds are presented in the table below:

**Table 8-6 End-of-Life Thresholds**

Threshold	Mounts
9x40 (T9840 and T9940) Warranty	10,000
9x40 End-of-life	11,000
T10000 Warranty	15,000
T10000 End-of-life	16,000

## Moving Least-Recently Accessed Cartridges from Active LSMs

When ACSLS dismounts a cartridge from a tape drive, it attempts to avoid pass-thru by "floating" cartridges that came from other LSMs to a new home cell in the same LSM as the tape drive.

For example, if a cartridge is mounted from SL8500 library 3, rail 2 (LSM 9) to a drive in library 1, rail 4 (LSM 3), this requires two horizontal pass-thrus and an elevator pass-thru. When ACSLS dismounts the cartridge, it tries to find a new home cell in LSM 4 and avoid any pass-thru on the dismount.

**Problem:**

It is impossible to float cartridges to an LSM if there are no free (unassigned) storage cells in the LSM. If there are no free cells in the tape drives' LSM, ACSLS still tries to dismount the cartridge to the closest LSM to the drive, but this requires at least one pass-thru.

**Solution:**

Identify cartridges that have not been accessed in a long time, and move them out of full LSMs to provide free cells to which cartridges can float to on a dismount.

In the information that ACSLS records about cartridges, the `access_date` is updated whenever a cartridge is mounted or dismounted, entered, or moved. The `access_date` can be used to identify those cartridges that are not active

This whole process is safe, as you are only moving volumes within the same ACS and are not doing anything which would prevent future mounts of these cartridges, such as ejecting cartridges or changing their status.

To identify and move least-recently used cartridges, follow the procedure below:

- Identify the LSMs with too few free cells and the LSMs with empty cells.
- Select the cartridges in an LSM by earliest access date.
- Move the cartridges from full LSMs to LSMs with empty cells.

## Identify the LSMs with too Few Free Cells and LSMs with Empty Cells

Using `cmd_proc`:

```
query lsm all
```

The Free Cell Count column lets you identify both LSMs with few or no free cells, and LSMs that have empty cells where you can move inactive cartridges.

### Example:

```
ACSSA> query lsm all
2011-08-29 18:15:45          LSM Status
Identifier  State  Free Cell  Audit  Mount  Dismount  Enter  Eject
              Count    C/P    C/P    C/P    C/P    C/P
1,0         online    1      0/0    3/0    3/0      0/0    0/0
1,1         online    1      0/0    4/0    5/0      0/0    0/0
1,2         online    1      0/0    3/0    3/0      0/0    0/0
1,3         online    0      0/0    4/0    5/0      0/0    0/0
1,4         online   388     0/0   11/0    1/0      0/0    0/0
1,5         online   162     0/0    4/0    5/0      0/0    0/0
1,6         online   552     0,0    7/0    2/0      0/0    0/0
1,7         online   601     0/0    5/0    3/0      0/0    0/0
```

## Review Cartridges in an LSM by Access Date

You now need to identify inactive cartridges that you can move to other LSMs in the ACS.

## Ensure Access Date is Reported for Easy Sorting

Ensure that the access date is reported in a manner conducive to sorting. The format that dates are reported is controlled by the `TIME_FORMAT` dynamic variable.

- Use the default format: `TIME_FORMAT=%Y-%m-%d %H:%M:%S`. so cartridges can easily be sorted by access date. At the UNIX command prompt, enter:

```
dv_config -p TIME_FORMAT
```

Enter ? at the variable prompt to display help.

- If you have made changes, update the dynamic variable in the shared memory:

```
dv_config -p TIME_FORMAT -u
```

## Look at the Distribution of Last Access Dates for Cartridges in an LSM

For each LSM without sufficient free cells, list the cartridges sorted by the last access date. You need a custom `volrpt` that only selects the `VOLID` and `access_date`.

For complete details, see the comment header in:

```
$ACS_HOME/data/external/volrpt/owner_id.volrpt
```

The fields on a line are `field_name`, `field_length`, and `delimiter_length` (spaces after the field).

In the following example, there are two active lines. You probably have 6-character `VOLUME_IDS`. For the `ACCESS_DATE`, you want only the date portion, not the time.

```
VOLUME_ID      6      2
ACCESS_DATE    10     2
```

To create the report:

1. Type: `$cd ACS_HOME/data/external/volrpt`.
2. Copy `owner_id.volrpt` and save it to a file, such as `access_date.volrpt`. See [Creating a Logging Volume Statistics Report](#).
3. Use a text editor to edit for `ACCESS_DATE`.
4. Create a sorted list of cartridges for an LSM.

```
volrpt -l <lsm_id> -d -f access_date.volrpt | sort -k 2,2 -0
vols_sorted_lsm_##
```

Where `access_date.volrpt` is the name of the custom report and `##` is the LSM number.

Look at the `vols_sorted_lsm_##` file to see the distribution of last access dates in each LSM.

## Move the cartridges from Full LSMs to LSMs with Empty Cells

You now need to create lists of cartridges to move, and move them to LSMs with free space.

### Create Lists of Cartridges to Move

1. Take the file with the list of cartridges sorted by access date and remove the access date so you just have a list of cartridges.
2. Take each `vols_LSM_##` file and select the first 100 (or your number) of cartridges to move.

```
cat vols_sorted_LSM_## | cut -d" " -f1 > vols_LSM_##_tmp
```

```
head -100 vols_LSM_##_tmp > vols_LSM_##
```

You can combine both of the above operations:

```
cat vols_sorted_LSM_## | cut -d" " -f1 | head -100 > vols_LSM_##
```

## Move the Cartridges to an LSM with Free Space

For each LSM from which you are moving cartridges, pick a destination LSM that has free space for the cartridges.

1. Use the `moving.sh` utility to move the cartridges to a new LSM, specified by `-t <lsm_id>` (example: `-t 0,8`).

```
moving.sh -f vols_LSM_## -t <lsm_id>
```

2. Run a separate `moving.sh` for each LSM.

If your library is busy, you may want to only run one or two `moving.sh` utilities at a time.

## Manually Loading Cartridges Into Drives in a Disabled LSM

If your LSM fails and you take it off line, you can still manually load cartridges into the library drives if the data path is still operational.

To manually load cartridges into drives in a disabled LSM:

1. Open the LSM door.
2. Note the cartridge labels of any cartridges that are already in drives, and remove them. You will need to replace these cartridges at the end of this procedure.
3. Load the drives with the cartridges you want to read from or write to.

Repeat this step as often as necessary until the LSM is repaired, then continue with Step 4.

### Caution:

In this step, you can remove cartridges from library cells and load these cartridges in the drives. Note the cell locations of these cartridges and ensure that you return the cartridges to these locations in Step 4.

4. After the LSM is repaired, remove all cartridges from the drives, and replace them with the original cartridges you noted in Step 2.
5. Close the LSM door, vary the LSM back online, and resume normal operations.

# 9

## Database Administration

The database contains all information about the library configuration and the location of all library cartridges.

ACSL control files that are backed up and are recovered include the customer-configurable files located in `$ ACS_home` under `data/external` and some files located in `data/internal/client_config`.

This chapter discusses: importing and exporting the database; verifying the imported database and library configuration; backing up the database; and restoring and recovering the database.

- Exporting and Importing the database includes:
  - Exporting the database to a disk file or local tape device
  - Importing the database from a disk file or local tape device
  - Importing ACSLS control configuration files
  - Merging any customized dynamic variables
  - Verifying the imported database and library configuration
- Backing up the database and ACSLS control files includes:
  - Automatic database backup
  - Performing manual backups to a local tape device or to disk
  - Backing up to a UNIX File
  - Creating a backup that can be restored to a different server
- Recovering and restoring the database and ACSLS control files includes:
  - Restoring the database to the most recent backup
  - Recovering from a specific file
  - Restoring ACSLS control files
  - Restoring a backup created on a different server
  - Restarting the database

### Utilities Used

You will use the following utilities:

- The `bdb.acsss` utility for backups to a:
  - specified UNIX file
  - tape device
  - default file and location
- The `rdb.acsss` utility for:



- recovering the database from corruption
  - from changes that produce unintended results
  - from server failure
- The `db_export.sh` and `db_import.sh` utilities for migrating between versions of ACSLS. This includes going to a later release, or going to a previous release.

When you install ACSLS, you also automatically install the database management software. The ACSLS database is initialized after ACSLS is installed when you:

- configure the library hardware using `acsss_config`
- import a previous exported database using `db_import.sh`
- recover a database backup created on a different server using `rbd.acsss.sh`

## Exporting the Database

This section describes how to migrate the ACSLS database and its associated ACSLS control files from either a previous version of ACSLS, the same release level of ACSLS, or return to a prior release.



### Note:

You cannot run `db_export.sh` while ACSLS is running. This ensures that you get a consistent copy of the database. Use `acsss disable` to disable ACSLS.

The `db_export.sh` utility creates an ASCII representation of the database on tape or a specified file to disk. It is also responsible for gathering ACSLS control files. This utility can be used in two different ways.

- If it is executed without any options, the exported files are copied to the default tape device: `/dev/rmt/0n`

```
db_export.sh
```

- If you want to use a different tape device, use the `-f` option, followed by the desired tape device:

```
db_export.sh -f /dev/rmt/3n
```

- If you want to export to a local file on the same machine, specify the file path name using the `-f` option:

```
db_export.sh -f /export/save/acsls_export.03_Dec_2014
```

When saving to a file, the result is two separate files. The database tables is saved in the file name you specify. The miscellaneous control files have the identical path name with a `.misc` extension.

The files generated by `db_export.sh` are then used as input to the `db_import.sh` utility at the time of an upgrade or recovery.

**Note:**

This is the preferred method for migrating all previous ACSLS versions to the current version.

When executing the `db_export.sh` utility either with the `-f` option or without, you will be prompted to choose the version of ACSLS to which you are exporting.

```
$ db_export.sh
```

Exporting database to `/dev/tape`

Choose the release to which you are exporting by selecting from the options below:

If exporting to tape, a no-rewind device is required.

```
1: ACSLS 5.3.2 or 5.4
2: ACSLS 6.0 or 6.0.1
3: ACSLS 6.0.1 with L700e
4: ACSLS 6.0.1 with PUT0201
5: ACSLS 6.1, 7.0, or 7.1/7.1.1 before PUT0701
6: ACSLS 7.1/7.1.1 with PUT0701 or ACSLS 7.2 (any)
7: ACSLS 7.3 (any)
8: ACSLS 8.0, 8.01, 8.02, and 8.1
9: ACSLS 8.2 or later
E: Exit
```

## Removing Unsupported Tape Libraries, Drives, and Cartridges Before Exporting to an Earlier Release

If you are exporting your database to an earlier release of ACSLS that does not support some of your tape libraries, tape drives, or cartridge media types, remove the unsupported tape libraries from your configuration, tape drives and cartridges from your libraries before exporting your database. Otherwise, the following may occur:

- If you select an earlier ACSLS release that does not support a library, you are prompted to remove the library from your configuration before exporting your database.
- If you export tape drives to an earlier ACSLS release that does not support them, the drives are reported as "unknown," and you are unable to use them.
- If you export cartridges to an earlier ACSLS release that does not support their media type(s), the cartridges are marked as absent, and you must manually remove them from your libraries.

## Removing Logical Libraries before Exporting to Linux

ACSLs running on Linux does not support logical libraries accessed using a Fibre target. If you are exporting your database to ACSLS running on Linux, remove any logical libraries. Otherwise, you will not be able to use any logical libraries running on Linux.

## Exporting to a Disk File

You can export the ACSLS database and ACSLS control files to a disk file, as shown in the following procedure.

1. Log in as `acsss`.
2. Disable ACSLS:  

```
acsss disable
```

(from a UNIX command prompt)
3. Start the `db_export.sh` utility.  

```
db_export.sh -f /path/db_file
```
4. Select the desired option for the version to which you are migrating.
  - As it executes, the utility displays output indicating successful table data being exported.
  - When the export is complete, a message is displayed indicating that the export has been successful.
  - The `db_export.sh` utility creates two files: `db_file` and `db_file.misc` in the location specified with the `-f` option.
5. Ensure that these files are placed in or moved to a secure location where they will not be removed.

Do **not** put these files in or under the following directories, as the directories may be removed or deleted when ACSLS maintenance is installed:

- `$ACS_HOME`(the ACSLS home directory)
- `$ACSDB_BACKUP_DIR` (such as `/export/backup`) (directory where ACSLS backups are stored)
- `/tmp`

 **Note:**

If you plan to install a new release of the operating system, do **not** save the exported files on the ACSLS server.

6. To start ACSLS and the database, enter the following command:

```
acsss enable
```

## Exporting to Tape

You can export the ACSLS database and ACSLS control files to tape, as shown in the following procedure.

To export the database and ACSLS control files to tape:

1. Log in as `acsss`.
2. Disable ACSLS:  

```
acsss disable
```

(from a UNIX command prompt)
3. Insert a blank tape into the default tape device.
4. Start the `db_export.sh` utility;

```
db_export.sh -f tape_device
```

**Example:** `dbexport.sh -f /dev/rmt/0mn`

5. Select the desired option from which you are exporting.

As it executes, the utility displays output indicating successful table data being exported and successful ACSLS files being backed up. A message displays when the export is completed.

6. Remove the cartridge from the drive only when the program completes and the prompt re-appears.

**▲ Caution:**

You will lose files if you remove the cartridge before the program completes the export. Write "protect the cartridge" and clearly mark it to identify the contents as the exported database.

Do not leave the cartridge in the library.

7. To start ACSLS and the database, enter the following command:

```
acsss enable
```

## Importing the Database

The following attributes are imported into the new database when you use the `db_import.sh` utility.

- Volumes: These database tables include all of the information associated with each volume in the library, such as:
  - where the volume resides
  - type of cartridge (such as data, scratch, and cleaning)
  - last associated scratch pool
  - current status of the cartridge (home, mounted, and so on)
  - entry date and last accessed date
  - number of mounts since the entry date
  - maximum use (for cleaning cartridges)
  - associated lock ID and user ID (if the cartridge is locked)
- ACS and Library: database tables include the ACSs and library components, such as: LSMs, drives, panels, and cells
- ACSLS control files include all configuration updates since the initial installation, including:
  - access control information
  - fixed volume preferences
  - scratch media preferences
  - custom volrpt templates

- Dynamic and static variables: dynamic variables that have been customized in a previous release can be imported

This section describes how to use the `db_import.sh` utility to:

- recreate the ACSLS database
- recover important ACSLS control files
- recover customized dynamic variables from data exported using the `db_export.sh` utility.

## Importing From a Disk File

You can import the ACSLS database and ACSLS control files from a disk file, as shown in the following procedure.

To import the ACSLS database, ACSLS control files, or customized dynamic variables from a disk file:

1. Log in as `acsss`.
2. Disable ACSLS.

```
acsss disable
```

(from a UNIX command prompt)

3. Start the `db_import.sh` utility.

```
db_import.sh -f db_file
```

```
ACSLS Import Utility
```

```
If importing from tape, a no-rewind device is required.
```

```
What would you like to do:
```

- ```
1) Import data, control files, and dynamic variables from
   from a DIFFERENT release or platform version of ACSLS (upgrade)

2) Import data, control files, and dynamic variables from the SAME
   release (version and PUT level) and platform of ACSLS(Disaster
   Recovery)

3) Import database tables only (any level of ACSLS)

4) Import control files only (any level of ACSLS)

5. Merge customized dynamic variables only (any level of ACSLS)

E) Exit
```

```
Please select one of the above:
```

- Option 1 - importing data, control files, and dynamic variables from a different release or platform version.

Use this option to import database files, control files, and dynamic variables when moving to a different release or upgrading ACSLS.

 **Caution:**

Existing database and control files and dynamic variable settings, are destroyed, re-built, and populated with the data provided from the export. The results are final and there is no recovery without rebuilding the database. To preserve information in existing tables, do not continue unless you have exported the table data using `db_export.sh`.

This option also recovers customized dynamic variables from previous environments. This is useful when upgrading versions of ACSLS without having to record previous customized dynamic variables. All files in the directory `acs.home` under `data/external` including access control files are recovered. If access control is configured, it also recovers `data/internal/client_config`.

- Option 2 - importing data, control files, and dynamic variables from the same release or platform version

Use this option to recreate an ACSLS environment, including both the database and the control files. This would be used when:

- recovering from a hardware failure or during a hardware upgrade.
- you must rebuild the ACSLS server to be identical to the ACSLS server from which the data was exported.

- Option 3 - importing only database tables from any ACSLS release level

Use this option to import only database files from any ACSLS release level.

This option destroys the existing database tables and control files, rebuilds them, and then populates them with the data provided from the exported database. To preserve information in existing tables, do not continue unless you have exported the data using `db_export.sh`

- Option 4 - importing ACSLS control files from any ACSLS release level

Use this option to import only ACSLS control files either from any version of ACSLS. This imports all files in the directory `acs.home` under `data/external` including access control files. If access control is configured, it also imports `data/internal/client_config`.

This option recovers ACSLS database files, control files, and dynamic variable from the same version. This recovers all files in the directory `acs.home` under `data/external`, including access control files.

This option recovers customized dynamic variables from previous environments. This is a very useful option for upgrading versions of ACSLS without having to record previous customized dynamic variables.

Selecting this option gathers the settings from the database export, and then reconfigures shared memory with the new variable settings.

- Option 5 - merging only customized dynamic variables

This is a very useful option for upgrading versions of ACSLS without having to record previously customized dynamic variables. Selecting this option gathers the settings from the database export and reconfigures shared memory with the new variable settings.

 **WARNING:**

If you are importing from ACSLS 7.2.0 and if you start ACSLS before executing this option, certain data could be lost. If you are upgrading ACSLS from a previous version and had customized dynamic variables, you should import your customized variables BEFORE starting ACSLS.

4. Verify the install as described under [Verifying the Imported Database and Library Configuration](#).
5. To start ACSLS, enter the following command:

```
acsss enable
```

## Importing from Tape

Use the following procedure to import the ACSLS database, recover ACSLS control files, and rebuild customized dynamic variables from tape.

1. Log in as `acsss`.
2. Disable ACSLS.

```
acsss disable
```

(from a UNIX command prompt)

3. Insert the exported database tape that you exported with the `db_export.sh` command into the tape drive.
4. Run the database import utility by entering the following at a UNIX command prompt.

```
db_import.sh
```

The `db_import.sh` utility displays its main menu as shown in [Importing From a Disk File](#). It also provides more information.

 **Note:**

You receive an "unsuccessful" message if you run the `db_import` utility from one terminal while doing a tape rewind from a different terminal.

5. Refer to the step "3" for menu options.
6. Verify the install as described in [Verifying the Imported Database and Library Configuration](#).
7. Import from tape, other than the default tape device (no rewind).
8. To start ACSLS, enter the following command:

```
acsss enable
```

## Migrating mchangers for Fibre to a New Platform

The SCSI Media Changer (mchanger) is the Fibre attached library device driver that communicates between ACSLS and any Fibre attached library. A mchanger must be created for each Fibre attached library that is connected to ACSLS.

The numbers in `/dev/mchanger#` device driver links may change when importing ACSLS to another platform and/or release which can create problems. For example, an SL500 or SL150 library that was connected through `/dev/mchanger3` on the old ACSLS server may be connected through `/dev/mchanger4` on the new ACSLS server.

This is not an issue when going from one ACSLS Linux server to a new Linux server because the format of mchanger names is different on Linux. Instead of a number, on Linux servers, the mchanger name includes the library serial number.

The following procedure avoids problems when you have media changer drivers configured for Fibre-attached libraries and migrate to a new ACSLS release or server platform.

1. On a Solaris or AIX ACSLS server, record the mchanger number associated with each Fibre-attached library on the old ACSLS server. See [Record Details about Fibre-attached Libraries on the Old ACSLS Server](#) .
2. Update your configuration with the new mchanger names for the libraries. See [Reconfigure ACSLS to Change mchanger Names for Fibre-attached Libraries](#) .

### Record Details about Fibre-attached Libraries on the Old ACSLS Server

On a Solaris or AIX ACSLS server, before exporting the database from the old ACSLS server, record the mchanger number associated with each Fibre-attached library on the old ACSLS server. Save the output from `cmd_proc` and the `showDevs.sh` utility that shows the mchanger associated with each Fibre-attached ACS and the serial numbers of the libraries.

`cmd_proc:`

- `query lmu all`

This shows all ACSs controlled by ACSLS and their port connections. The port names for Fibre-attached libraries on Solaris and AIX systems will be `/dev/mchanger#`, where # is a number

- `display lsm * -f type serial_num`

This displays the library type and serial number of all LSMs managed by ACSLS. Use library type, such as SL500 or SL150, to identify the Fibre-attached libraries. Use the serial number to identify the specific library.

#### Utilities:

`showDevs.sh -s`

The `showDevs.sh` utility with the `-s` option shows the mchanger device link, the library type, the library serial number, and details which identify the Fibre-attached library.



## Reconfigure ACSLS to Change mchanger Names for Fibre-attached Libraries

After importing your database, if you are either migrating to or from Linux, or the same mchanger numbers were not configured on Solaris, you must update your configuration with the new mchanger names for these libraries.

Using `acsss_config`:

1. Log in as `acsss`.
2. Use `showDevs.sh` to display all of your Fibre-attached libraries.  
The `showDevs.sh` utility with the `-s` option shows the mchanger device link, the library type, the library serial number, and details which identify the Fibre-attached library.
3. Save the output from `showDevs.sh` in a file so you can copy and paste it into prompts from `acsss_config`.
4. With the output from `showDevs.sh -s` displayed in one terminal window, open a second terminal window and log in as `acsss`.
5. Run `acsss_config` in the second terminal window.
6. Select Option 8: Define or Change Library Configuration.
7. Reply `y` to Configure library communications? (y/n):
8. Reply `y` to Library server data base exists and will be overwritten, continue (y or n)?
9. Referencing the saved output from `query lmu all`, reconfigure all of the ACSs that were configured on your old ACSLS server.
  - a. Configure all of the ACSs in the same order, and with the same ACS numbers as they were on the old ACSLS server.
  - b. Configure the non Fibre-attached libraries as partitioned or not partitioned, and with the same port connections as they had on the old ACSLS server.
10. When configuring the port connections for your Fibre-attached libraries, specify the new mchanger link names used on the new ACSLS server. For mchanger link names on Linux, the easy way to do this is copy them from the `showDevs.sh` output and paste them after the `acsss_config` prompt.
11. Finish reconfiguring your ACSLS library hardware.

## Verifying the Imported Database and Library Configuration

Use the following procedure to mount or dismount a cartridge to verify ACSLS.

1. Verify that you are logged in as `acsss`.
2. If ACSLS is not running, start it by entering the following command:

```
acsss enable
```
3. Query the server from the `cmd_proc` by entering the following command:

```
query server
```

If messages are displayed indicating that the server is in recovery mode, wait for a message indicating that the server is running.

4. Verify that the following are online. If not, bring them online with the `vary` command.

```
query port all
query acs all
query lsm all
query drive all
```

5. Do you have at least one cartridge in an LSM:
  - YES - Continue with the procedure.
  - NO - Enter a cartridge into an LSM.
6. Mount a cartridge by entering the following command:

```
mount vol_id drive_id
```

Use the `query drive` command to get the ID of an available drive and the `query volume` command to get the ID of a library cartridge.

7. Did you see a message indicating a successful mount?

A successful mount message is:

```
Mount: vol_id mounted on drive_id
```

- YES - Procedure is complete.
  - NO - If an error message appears, run this verification procedure again to ensure that you specified a valid, available drive and a library cartridge. If the mount or dismount still fails, call Support for assistance.
8. Dismount the cartridge by entering the following command:

```
dismount vol_id drive_id force
```

where *vol\_id* is the volume and *drive\_id* is the drive you specified in Step 6.

## Automatic Database Backup

ACSLs automatically creates a backup file of the database-to-disk every 24 hours at midnight, or, the time of day and days of the week you specified in the backup options in `acsss_config`.

## Performing Manual Backups to Tape

In addition to the automatic database backups that ACSLS creates, you should periodically run the `bdb.acsss` utility to manually create tape backups that can be stored off-site and used, if needed, for disaster recovery of the database.

Regular backups transferred to an off-site device can enable rapid restoration if disaster to the ACSLS server occurs.

Use `bdb.acsss` to manually back up the database to tape after:

- Running `acsss_config`.
- Importing the database.
- An audit of the entire library.

- Any database recovery.

## Backing up to a Specified Tape Device Attached to the ACSLS Server

To backup the ACSLS database to a specified tape device attached to the ACSLS server, do the following:

1. Log in as `acsss`.
2. Insert a blank tape into the tape device.
3. From a terminal window, enter the following command:

```
bdb.acsss -f tape_device
```

Where *tape\_device* specifies a tape device attached to the ACSLS server.

4. Messages reporting the progress of the backup appear.

Wait for the following message to appear:

```
Check tape device (/dev/rmt/0mn) to make sure you have a tape in the  
tape drive.
```

```
[Hit RETURN to continue or Ctrl-C to exit]
```

```
Press RETURN.
```

5. Wait for the following message to appear:

```
ACSL database backup successfully completed.
```

**Example:** To backup the ACSLS database to tape device `/dev/rmt/0mn`, enter the following command:

```
bdb.acsss -f /dev/rmt/0mn
```

## Backing up to a UNIX File

In the interest of disaster recovery, Oracle does not recommend that you backup to a UNIX file unless the file is on a remote disk. See [bdb.acsss](#) .

To back up the ACSLS database to a UNIX file, do the following:

1. Log in as `acsss`.
2. From a terminal window enter the following command:

```
bdb.acsss -f db_file
```

Where *db\_file* specifies a UNIX file to contain the ACSLS database. You must have write permissions to the file.

3. Wait for the following message to appear:

```
ACSL database backup successfully completed.
```

## Recovering and Restoring

This section describes the following restoration/recovery procedures:

- Restoring a corrupted or lost database to the most recent backup
- Restoring a corrupted or lost database to a specified date and time
- Recovering from a disk failure
- Disaster recovery for a failed server
- Recovering from a specific backup file
- Restoring non-database, ACSLS control files

Most of these procedures use the `rdb.acsss` utility, which provides options for restoring a database from the most recent backup, or from a specified date and time; disaster recovery using a backup created by `bdb.acsss`; and restoring ACSLS control files created by `bdb.acsss`. For more information about these options, see [rdb.acsss](#).

 **Note:**

If the home cell of a cartridge changes from its last location after a backup, then the restored database will not be up-to-date. To avoid cartridge movement on dismounts: each LSM must be the only LSM in its ACS (true in most SCSI libraries), or the Extended Store Feature must be enabled for all LSMs that are connected to other LSMs through a pass-thru-port.

For more information, see [Using the Extended Store Feature](#). If the Extended Store Feature is not enabled for all connected LSMs or cartridges have been entered or ejected, audit the library after the restoration to make the database current, and enable all LSMs that are connected to other LSMs through a pass-thru-port.

 **Note:**

Do not specify the `-f` option as a general option for the `rdb.acsss` utility. If you backed up your database to an external network file or to an alternate tape device, use the `-f` option only after entering `rdb.acsss`. Choose the third recovery option. When prompted, enter `-f` and the path name to your external network file or an alternate tape device. See "6" for more information.

## Restoring the Database to the Most Recent Backup

In this procedure, you restore the database to the most recent backup created on the local disk by automatic backups. ACSLS control files are also restored.

To restore a corrupted or lost database to the most recent backup, complete the following steps:

1. Log in as `acsss`.
2. Disable ACSLS.

```
acsss disable
```

3. Enter the following command:

```
rdb.acsss
```

4. Select option 1:

1. Restore from a current local disk backup
5. Refer to [rdb.acsss](#) for procedures.
6. To start ACSLS, enter the following command:

```
acsss enable
```

## Recovering from a Failed Server

Use this procedure for a disaster recovery when you have lost or corrupted both primary and secondary disks.

To recover from a failed server, complete the following:

1. Install the operating system.
2. Install ACSLS.

### Caution:

You *must* install ACSLS in the same directory you used before the disk failure.

3. Log in as `acsss`.
4. Disable ACSLS:  

```
acsss disable
```
5. Enter the following command:  

```
rdb.acsss
```
6. Select option 2:  

```
2. Restore from a previous tape or network file backup
```
7. Refer to [rdb.acsss](#) for procedures.
8. To start ACSLS, enter the following command:  

```
acsss enable
```
9. You must run `acsss_config` to re-specify automated backup date and time and retention periods *unless* you want to accept the default settings.

## Restoring ACSLS Control Files

In this procedure, you restore ACSLS control files. These are non-database files that include all files in the `data/external` directory such as access control files, the fixed volume file, the scratch preferences file, and custom `volrpt` files. These files are restored from a `bdb.acsss` backup-to-tape or an external network file.

To restore ACSLS control files, complete the following:

1. Log in as `acsss`.
2. Disable ACSLS:

```
acsss disable
```

3. Enter the following command:

```
rdb.acsss
```

4. Select option 4:

```
Restore only ACSLS non-database control files
```

5. Refer to [rdb.acsss](#) for procedures.

6. To start ACSLS and the database, enter the following command:

```
acsss enable
```

# 10

## Reporting and Logging

This chapter describes how to:

- Create a custom volume report
- Create a report for volume movement statistics

### Creating a Custom Volume Report

You can use the `volrpt` utility to create a volume report; for more information, see [volrpt](#). `$ACS_HOME/data/external/volrpt/owner_id.volrpt` is a sample input file that you can run or use as a template to create customized volume reports. You can also save your customized volume reports in the `$ACS_HOME/data/external/volrpt` directory. Use this procedure to create an input file to the `volrpt` utility to create a custom volume report.

To create a custom volume report, complete the following:

1. Log in as `acsss`.
2. Open a UNIX command tool.
3. Change to the custom volume report directory:  

```
cd /home/ACSSS/data/external/volrpt
```
4. Copy the sample volume report file to a new, custom file.  

```
cp owner_id.volrpt my.volrpt
```

`my.volrpt` is the name you want to assign to the new file.
5. Using a text editor, such as `vi`, edit the `my.volrpt` file to specify the fields and formats you want to see on the custom report.
  - You can specify any of the fields listed in the sample file.
  - The format for each entry is: `field_name field_length delimiter_length`
  - You can make the field lengths and delimiters any size you want. Just be sure that all of the fields you specify will fit on one line when the report prints.
  - Detailed editing instructions are given in the sample file.
6. When you are finished editing the file, save it.

### Custom Volume Report

You create a customized report by designating fields, field length, and delimiter lengths in an input file, such as the one shown in the following example, below.

```
#####  
#  
# File name: owner_id.volrpt  
#  
# This file describes the report layout for volrpt invoked with # the -f option. #
```

```

volrpt -f <filename>
#
# The format of a line is:
# field_namefield_lengthdelimiter_length
#
# The field length is the number of characters which will be printed for
# the field. The delimiter length is the number of spaces that will be
# printed after the field. If you leave out the lengths, or specify a
# value of -1, the default values will be used. Default delimiters are
# always 2. Here are the fields and their default lengths.
#
# ACCESS_COUNT      5      2
# ACCESS_DATE      15     2
# CELL_ID          14     2
# DRIVE_ID         10     2
# ENTRY_DATE      15     2
# LABEL_ATTR       5      2
# LOCK_ID          5      2
# LOCK_TIME       15     2
# MAX_USE          5      2
# MEDIA_TYPE       7      2
# OWNER_ID        20     2
# POOL_ID          5      2
# VOLUME_ID        6      2
# VOL_STATUS      17     2
# VOLUME_TYPE      4      2
#
# Revision History:
# xx/xx/xx Name      Changes
#
#####
VOLUME_ID            6      2
MEDIA_TYPE           7      2
DRIVE_ID            12     2
CELL_ID             14     2
OWNER ID            -1     0

```

From this input file, generate the customized report shown. Among other uses, a customized report, as shown in the following example, allows you to report the owners of the volumes.

An example of a customized volume report using the input file:

```

2014-06-30 13:22:07

TOTAL    VOLUMES:2    SEQUENCE:    sort by volume identifier

Volume   Media                Home                Owner

Label    Type                Drive ID            Location            ID

RB1400   3480                Not-in-drv         0, 1, 1, 0, 0      cray
RB1401   DD3A                0, 0, 1, 0         0, 1, 2, 0, 0      cray

```

## Custom Volume Report Example

The following `volrpt` includes ACSs 0 and 1, absent and ejected volumes, and shows the status of volumes.

The input `volrpt` option::



```
volrpt -f my.volrpt -a 0 1 -i
```

These control statements are used to select and format the output.

Example of a custom volume report control statements:

```
CELL_ID          14          2
VOLUME_ID        6          2
VOL_STATUS       17          0
POOL_ID          5          2
ACCESS_COUNT     5          1
LOCK_ID          5          1
OWNER_ID         20         0
```

Example of custom volrpt showing absent and ejected volumes:

```
VOLUME REPORT UTILITY

2014-06-03 15:27:48

TOTAL VOLUMES: 61      SEQUENCE: sort by volume identifier

Home      Volume  Volume      Pool  Times  Lock  Owner
Location--- Label  Status      ID    Mount ID   ID-----

1, 0, 0, 0, 0 ABC001 VOLUME_HOME  0    2    0    presc
0,-1, 0, 0, 0 ABC002 VOLUME_ABSENT 0    0    0
0,-1, 0, 0, 0 ABC003 VOLUME_ABSENT 0    0    0
1, 3, 0, 0, 3 ABC004 VOLUME_MISSING 0    0    0
1, 3, 0, 0, 4 ABC005 VOLUME_MISSING 4    0    28001 tom
1, 3, 0, 0, 5 ABC006 VOLUME_MISSING 0    0    0
0,-1, 0, 0, 0 ABC007 VOLUME_ABSENT 0    0    0
1, 0, 0, 0, 7 ABC008 VOLUME_HOME  0    0    0
0,-1, 0, 0, 0 ABC009 VOLUME_ABSENT 0    0    0
0,-1, 0, 0, 0 ABC010 VOLUME_ABSENT 0    0    0    presc
1, 0, 0, 0,10 ABC011 VOLUME_HOME  0    0    0
1, 0, 0, 0,12 ABC012 VOLUME_HOME  0    0    2371 abc012
1, 0, 0, 0,13 ABC013 VOLUME_HOME  0    0    28001
1, 0, 0, 0,14 ABC014 VOLUME_HOME  0    0    28001
0,-1, 0, 0, 0 ABC015 VOLUME_ABSENT 1    0    29712
0,-1, 0, 0, 0 ABC016 VOLUME_EJECTED 1    0    29712
0,-1, 0, 0, 0 ABC017 VOLUME_ABSENT 1    0    29712
0,-1, 0, 0, 0 ABC018 VOLUME_ABSENT 1    0    29712
1, 0, 0, 0,19 ABC019 VOLUME_HOME  1    0    0
1, 0, 0, 0,20 ABC020 VOLUME_HOME  1    0    0
0,-1, 0, 0, 0 ABC021 VOLUME_ABSENT 0    0    0
0,-1, 0, 0, 0 ABC022 VOLUME_ABSENT 4    0    0
```

## Creating a Logging Volume Statistics Report

You can use the Volume Statistics Log file (`acsss_stats.log`) to log volume movement statistics. These statistics consist of entries for each time ACSLS detects that a volumes' location changed. ACSLS logs entries for enters, ejects, mounts, dismounts, and for each time an audit detects that a volumes' location has changed (typically, by being manually moved).

Use the `acsss_config` configuration program to perform the following:

- Enable or disable volume statistics logging using the `LIB_VOL_STATS` variable.

- Specify the maximum size of the Volume Statistics Log file.
- Specify the number of rollover files for the Volume Statistics Log file.

The `stats_report` utility uses the `acsss_stats.log` to report all mounts and tape drive usage.

The Volume Statistics Log file contains collection mode entries that tell if volume statistics logging is enabled or disabled and volume statistics entries.

Example of Volume Statistics Log file entries:

```
2014-06-30 08:53:00 CONFIG
Library volume statistics on.

2014-06-30 09:23:08 EJECT
U01120 Home 0,0,1,3,5 Cap 1,0,0 Client Host Id 129.81.15.25

2014-06-30 10:36:05 ENTER
PB0444 Home 0,0,4,3,5 Cap 0,0,0 Client Host Id 129.81.15.25

2014-06-30 10:42:48 MOUNT
PB0478 Home 0,0,1,35,1 Drive 0,0,1,0 Client Host Id Local

2014-06-30 10:43:19 DISMOUNT
PB0478 Home 0,0,1,35,1 Drive 0,0,1,0 Client Host Id Local

2014-06-30 10:43:19 AUDIT
RB0478 0,0,1,35,1 STATUS_VOLUME_NOT_FOUND Client Host Id JBHUTTO

2014-06-30 10:43:19 AUDIT
PB0444 0,0,1,32,1 STATUS_VOLUME_FOUND Client Host Id JBHUTTO

2014-06-30 10:45:00 CONFIG
Library volume statistics off.

2015-01-16 09:51:07 ACSCR
0A1235 Home 0,0,5,14,14 STATUS_VOLUME_NOT_FOUND Client Host Id Local

2015-01-16 09:40:13 ACSCR
0A123A Home 0,0,5,14,15 STATUS_VOLUME_FOUND Client Host Id Local
```

In the above example, the collection mode entries show that statistics collection started at 8:53 a.m on June 30, 2014 and ended at 10:45 a.m. the same day. These collection start and stop times bracket the volume statistics entries for this collection period.

The format of the volume statistics entries, is:

- `yyyy-mm-dd hh:mm:ss command`  
`vol_id home_loc function_loc client_host_ID`

Where:

- `yyyy-mm-dd` is the year, month, and day of the entry. Four-digit year formats are supported. You specify the date format using `acsss_config`.
- `hh:mm:ss` is the hour, minute, and second of the entry.
- `command` is the ACSLS command or client request that either moved the volume or (for audits) detected that the volume was moved.

- MOUNT is the mount request.
- DISMOUNT is the dismount request.
- ENTER is the manual or automatic mode `enter` request.
- EJECT is the eject request.
- AUDIT is the audit request.
- ACSMV is the move request.
- ACSCR is the Volume Recovery activity. This activity is generated automatically by ACSLS processing.
- `vol_id` is the volume identifier.
- `home_loc` is the volumes' home (storage cell) location.
- `function_loc` is the volumes' location for the requests that used the volume as follows:
  - `mount` or `dismount` requests  
   location is a transport ID.
  - `enter` or `eject` requests  
   location is a CAP ID.
- `audit` requests specifies that an audit detected one of the following errors:
  - `STATUS_VOLUME_FOUND`  
   The audit found a volume in a location that did not match the location specified in the database.
  - `STATUS_VOLUME_NOT_FOUND`  
   The audit did not find a volume in the location specified in the database.
  - Volume Recovery activity  
   Location may be a cell ID or a transport ID, indicating that Volume Recovery detected one of the following situations:
    - \* `STATUS_VOLUME_FOUND`  
 A volume was found which was not recorded in the database, and is being added.
    - \* `STATUS_VOLUME_NOT_FOUND`  
 A volume in the database was not found in any recorded location, and is being deleted.
- `client_host_ID` is one of the following;
  - For client application requests it is the host IP address.
  - For `cmd_proc` commands, if the environment variable `LIBVOLSTATS_CMD_PROC_ID` is set (ASCII characters only) in the environment of the shell that started the `cmd_proc`, the entry is the first 12 characters of the value of the environment variable.
  - For `cmd_proc` commands, if the environment variable `LIBVOLSTATS_CMD_PROC_ID` is not set or contains non-ASCII characters, the entry is Local.

# 11

## Utility Reference

This chapter discusses the following ACSLS utilities:

- [acs\\_partition.sh](#) enables you to change the library partition identifier for a given ACS in your library complex without having to reconfigure the attached library.
- [acs\\_renumber.sh](#) enables you to change the identifier of a given ACS in your library complex without having to reconfigure the attached libraries.
- [acs\\_rewallet.sh](#) allows you to change the user name and password information stored in the ACS wallet for any SL4000 ACS.
- [The acsss Macro](#) starts and stops ACSLS, as well controls and monitors for maintenance and troubleshooting.
- [bdb.acsss](#) backs up the ACSLS database and ACSLS control files.
- [Dynamic Configuration \(config\) utilities](#) dynamically implements configuration changes to ACSLS libraries (and components) while ACSLS remains online and running. These configuration utilities are:
  - [config acs](#) dynamically adds an ACS or reconfigures an existing ACS and its components.
  - [config drives](#) - on existing drive panels, it dynamically adds drives, changes drive types, and deletes drives.
  - [config lsm](#) dynamically reconfigures an existing LSM and all of its components. These components include CAPs, panels, and drives.
  - [config ports](#) dynamically reconfigures the port connections to an ACS.
- [db\\_export.sh](#) exports the ACSLS database information and ACSLS control files in preparation for an upgrade installation or reinstallation of ACSLS.
- [db\\_import.sh](#) imports the ACSLS database information and ACSLS control files exported when you used the [db\\_export.sh](#) utility.
- [del\\_vol](#) deletes a volume from an offline LSM.
- [drives\\_media.sh](#) displays all drive types, media types and the drive-to-media compatibilities that are supported by the current release of ACSLS.
- [ejecting.sh](#) conducts mass eject operations quickly and efficiently.
- [free\\_cells.sh](#) allows you to monitor and manage the free cells in libraries controlled by ACSLS.
- [getHba.sh](#) manages Fibre Channel HBA ports.
- [greplog](#) filters the `acsss_event` log to include or exclude messages containing specific keywords.
- [install\\_scsi\\_Linux.sh](#) creates `/dev/mchanger` links which can be used when configuring libraries to ACSLS.
- [lib\\_type.sh](#) returns the LSM type of the LSMs attached to the specified ACS ID.
- [moving.sh](#) moves multiple cartridges to one or more LSMs.

- [probeFibre.sh](#) displays the model number, revision level, and Target-LUN address of each device connected behind an Emulex (LP10000) or QLogic (QLA2300) fibre-channel HBA.
- [rdb.acsss](#) restores the ACSLS database and ACSLS control files.
- [showDevs.sh](#) shows detail for every mchanger device configured on Solaris.
- [showDrives.sh](#) presents a list of all configured drives attached to ACSLS.
- [stats\\_report](#) gathers library volume statistical information.
- [userAdmin.sh](#) administers ACSLS GUI user passwords. You can add users, remove users, list users, and change user passwords.
- [volrpt](#) creates a volume report.
- [watch\\_vols](#) automatically assigns ownership and pool association to volumes as they are entered through the CAP.

## Overview

Follow these general guidelines for using the ACSLS utilities:

- Generally, the utilities described in this chapter are intended for execution by user `acsss`. To inherit the privileges and environmental dependencies required to run them you should login as user `acsss`.  
If you prefer to use `su`, be sure to use `su - acsss`.
- It is recommended that you use `bdb.acsss` to manually back up the database to tape after:
  - Configuring your library hardware.
  - Importing the database. After you upgrade to a new version of ACSLS, do not use database backups created with previous versions. Create a new backup as soon as you have upgraded.
  - Any database recovery.
- To ensure that you recover an accurate and consistent database, always use the most current database backup.

If a utility fails, retain all event logs. These logs aid in the support of resolving any problems.

## Legacy Start/Stop Scripts

The start and stop scripts that were used in ACSLS 7.x are not supported in ACSLS 8.x.

ACSL 8.x provided a new mechanism for starting and stopping the library management application, which is integrated with the Solaris Service Management Facility (SMF). This replaces `rc.acsss` and `kill.acsss` used in ACSLS. This mechanism also provides the ability to monitor application status.

You can start and stop ACSLS 8.x with the `acsss` command. The single command `acsss` provides ACSLS startup, shutdown, and monitoring functions. The utility resides in the `$ACS_HOME` directory and is accessible to any user.

## Utility Commands

The following section describes the ACSLS utilities.

### acs\_partition.sh

This utility enables you to change the library partition identifier for a given ACS in your library complex without having to re-run ACSLS configuration utilities. You can update more than one ACS at a time, as the utility prompts you for each configured ACS.

#### Note:

- This utility is not used to switch an ACS to a different partition. It reflects an update to partition numbers on the library side. The new partition identifier must refer to the same library.
- You must disable, but not shut-down ACSLS before running this utility script. The `acsdb` service must be online. The command `acsdb status` indicates the state.

To change the library partition id associated with an ACS, run the `acs_partition.sh` script. In an interactive session, you are first reminded about the note above and you are then prompted whether to continue.

```
$ acs_partition.sh

***** N O T I C E *****
* This utility can update the partition id for a configured ACS, *
* but the new partition id MUST refer to the same actual library *
* partition. To manage a different partition, re-configure ACSLS. *
*****
Continue...? (y or n): y
```

If you respond `y`, the routine automatically backs up the existing database before any changes are made. This allows you to restore to the previous configuration, should it be necessary to back out of the change. You can also reverse the change if needed, by repeating the `acs_partition.sh` routine.

The routine displays a list of the currently configured ACSs and, for each one, it asks whether to change the partition id for that ACS. If you do, it asks what new value to assign.

```
Current ACS list:
ACS=0 PORT=0 TYPE=SL4000 PARTITION=1 LIB=https://keystone13
ACS=1 PORT=0 TYPE=SL4000 PARTITION=21 LIB=https://keystone13
ACS=2 PORT=0 TYPE=SL4000 PARTITION=1 LIB=https://keystone35

Do you wish to change the partition ID for ACS-0 from 1? (y or n): n
Do you wish to change the partition ID for ACS-1 from 21? (y or n): y
  What is the new partition_id for ACS-1? 12345
Do you wish to change the partition ID for ACS-2 from 1? (y or n): n
```

Having accepted your input (in this example, your response was to update only ACS-1), the routine asks you to confirm the pending changes.

```
Fri Aug 9 13:22:34 MDT 2019
  Changing ACS-1 from partition 21 to partition 12345.

Correct? (y or n): y
```

If you answer `y`, the routine carries out all necessary updates to and automatically backs up the database in order to checkpoint the changes that you have made.

```
Updating table:   Changing ACS-1 to partition 12345
1 records

/export/home/ACSSS/utils/acs_partition.sh: ACS-1 was changed from partition 21
to partition 12345.

Complete!
Current ACS list:
ACS=0 PORT=0 TYPE=SL4000 PARTITION=1 LIB=https://keystone13
ACS=1 PORT=0 TYPE=SL4000 PARTITION=12345 LIB=https://keystone13
ACS=2 PORT=0 TYPE=SL4000 PARTITION=1 LIB=https://keystone35

Now backing up the database changes ...
```

## acs\_renumber.sh

This utility enables you to change the identifier of a given ACS in your library complex without having to reconfigure the attached libraries. Since every LSM, CAP, drive and volume in the library is identified in relation to an ACS, this utility updates all of the various database tables so that each library resource aligns with the new ACS ID that you assign.

New logical libraries would use the currently active pattern. For example, if you renumber ACS 0 to 1, then 1001 and 1002 would stay as they are, but a new logical library in ACS 1 would be 2001. If you then renumber ACS 6 to ACS 0, 7001 would stay as it is, but a new logical library in ACS 0 would be 1003. There is no real correspondence anymore, although newly-added ones would be predictable based on ACS.

### Note:

Changes made by this utility apply only to the ACSLS server and not to the client applications that use these resources. Consequently, it may be necessary to reconfigure any client databases after having changed the ACS i.d. on the server.

### Note:

ACSLs must be disabled before running this script.

To change the assigned number of an ACS, run `acs_renumber.sh`. In an interactive session, you are first warned that the changes made impact any client applications and you are then prompted whether to continue.

```
$ acs_renumber.sh
```

```
      N O T I C E
```

```
Changes made by this script will
impact client applications that
use ACSLS. Specifically, drive
i.d. mappings and LSM id's will change.
```

```
Continue...? (y or n):
```

If you respond `y`, the routine automatically backs up the existing database before any changes are made. This allows you to restore to the previous configuration, should it be necessary to back out of the change. (You can also reverse the change by repeating the `acs_renumber.sh` routine.)

The routine displays a list of the currently configured ACSs and, for each one, it asks whether to renumber that ACS. If you do, it asks what new value to assign.

```
Current ACS list:
```

```
ACS-0 (SL8500)
```

```
Do you wish to renumber ACS-0? (y or n):
```

```
What is the new value for ACS-0? 5
```

Having accepted your input (in this example, your response was 5), the routine asks you to confirm the pending change.

```
Change ACS-0 to ACS-5.
```

```
Correct? (y or n):
```

If you answer `y`, the routine begins updating all of the related database tables and automatically backs-up the database in order to checkpoint the changes that you have made.

```
Updating tables:    Changing ACS-0 to ACS-5
```

```
acstable: 1 records
captable: 4 records
celltable: 13424 records
drivetable: 128 records
handtable: 16 records
lmutable: 0 records
lshtable: 8 records
paneltable: 280 records
porttable: 1 records
ptptable: 16 records
scr_distr_table: 0 records
volumetable: 0 records
```

```
Complete!
```

```
Current ACS list:
```

```
ACS-5 (SL8500)
```



Now backing up the database changes...

## acs\_rewallet.sh

The `acs_rewallet.sh` utility re-creates the wallet for an SL4000 ACS, allowing you to change the user name and password information stored in the ACS wallet. This is helpful when the ACS wallet becomes corrupt, is accidentally removed, or when library changes the authentication profiles of the partitions as per their compliance or security policies.

### Note:

ACSLs must be disabled before running this script. If an active ACSLS is detected, the utility directs you to stop ACSLS with `acs sls disable`. The utility then exits.

To change the user name and password information stored in the ACS wallet, run `acs_rewallet.sh`. In an interactive session, you are first warned that the changes made impact any client applications and you are then prompted whether to continue.

```
$ acs_rewallet.sh

***** N O T I C E *****
* Changes made by this script will NOT impact client          *
* applications that use ACSLS.                                *
*****
```

Continue...? (y or n):

If you respond `y`, the routine displays a list of the currently configured SL4000 ACSs:

```
Only SL4000 ACSs require secured access, and hence need authentication profile.
Current SL4000 ACS list:
  ACS-0 (SL4000)
  ACS-1 (SL4000)
```

For each ACS listed, the routine asks whether to regenerate the wallet for that ACS. If you respond `y`, the routine prompts you to enter a username once and password twice (to verify your new password), and then recreates the wallet with the same name:

Do you wish to regenerate wallet for ACS-0? (y or n): y

ACS #0 : Interview for Authentication Profile

```
Enter User Name for ACS #0 : admin
Enter Password for ACS #0 (1st Time) : *****
Enter Password for ACS #0 (2nd Time) : *****
```

ACS #0 : Wallet creation successful.

```
-----
Recreate authentication profile for ACS-0...DONE
Regeneration of ACS wallet was successful.
```

Do you wish to regenerate wallet for ACS-1? (y or n): y

This is a critical operation. Make sure that all wallets are regenerated correctly. A single wallet with an incorrect authentication profile will prohibit ACSLS from starting, as shown in the following example:

```
$ acsss enable

Enabling ACSSES:
  WebLogic starting...
  acsdb enabled.
  acsls starting: [!!!!!!!!!!!!!!!!!!!!]
  acsls terminated!
+ exit 1

Problems starting ACSLES!
Details are logged toward the bottom of
the ACSLES event log and the SMF start log:
logfile      /export/home/ACSSES/log/acsess_event.log
logfile      /export/home/ACSSES/log/acsles_start.log

Disabling ACSSES:
  acsls disabled.
  surrogate disabled.
  rmi-registry disabled.
  acsdb disabled.
  weblogic disabled.
```

## The acsss Macro

The `acsss` macro is the primary start, stop, and status command that launches and brings down the various services associated with ACSLS. Depending on the installation, an ACSLS application is an aggregate consisting of up to seven services that are installed on the Solaris or Linux system.

- `acsdb` - manages the ACSLS library database.
- `acsls` - library control software that executes library operations.
- `weblogic` - web server for the ACSLS GUI.
- `surrogate` - communication link between java services and `acsls`.
- `rmi-registry` - lookup service for named java objects and methods.
- `smce` - SCSI Media Changer Emulation of logical libraries.
- `stmf` - target mode framework for logical libraries.

The `acsls` and `acsdb` services are common to all installations. The `weblogic`, `surrogate`, and `rmi-registry` services are present where the ACSLS GUI support has been configured. The `smce` and `stmf` services apply only if logical libraries have been configured (on Solaris).

All of the services are manipulated by the ACSLS user with the single macro, `acsss`, which starts and stops these services in a defined sequence required by dependencies between the different components. The macro issues commands to the Service Management Facility (SMF) on Solaris and the `init.d` service utility on Linux to undertake the actual work.

## Format

```
acsss <command>
```

Entering `acsss` without a command displays the list of options.

## Options

| Command                    | Function                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>enable</code>        | This is the default method to launch all the services associated with ACSLS. Once enabled, the various services remain enabled and will be re-enabled automatically after a system reboot.                                                                                                                                                                                                                                                                                                           |
| <code>temp-enable</code>   | Same as <code>acsss enable</code> but services are not restarted after a system reboot.                                                                                                                                                                                                                                                                                                                                                                                                              |
| <code>maint-enable</code>  | Intended for general maintenance operations not involving the ACSLS database. This option enables the GUI infrastructure allowing GUI users to remain logged in while ACSLS is disabled. This method is used in contexts of minor software patch installations. Neither the <code>acsls</code> nor the <code>smce</code> service is enabled.                                                                                                                                                         |
| <code>db</code>            | This is the preferred control mode to use for database maintenance operations including <code>db_export</code> , <code>db_import</code> and <code>acsss_config</code> . It enables the ACSLS database engine and disables all other ACSLS services including the ACSLS GUI.                                                                                                                                                                                                                          |
| <code>disable</code>       | This is the default method used to halt ACSLS operation. It is not a complete shutdown and allows for the database and any GUI login sessions to remain active for maintenance operations after the <code>acsls</code> and <code>smce</code> services have been disabled. The resulting state is identical to that of <code>acsss maint-enable</code> . This is the safest method to bring down the server since ACSLS and the library are placed in an idle state before the services are disabled. |
| <code>force-disable</code> | Same as <code>acsss disable</code> but the operation does not wait for an idle state before disabling <code>acsls</code> and <code>smce</code> .                                                                                                                                                                                                                                                                                                                                                     |
| <code>shutdown</code>      | This renders a complete shutdown of all ACSLS services. It is intended for contexts of software installation and de-installation, and other maintenance contexts that require the database ( <code>acsdb</code> ) or the GUI infrastructure ( <code>rmi-registry</code> and <code>surrogate</code> ) to be shutdown.                                                                                                                                                                                 |
| <code>status</code>        | Provides a quick status report of the various ACSLS services.                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| <code>a-status</code>      | Returns the operational status of the <code>acsdb</code> service.                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>d-status</code>      | Returns the operational status of the <code>acsls</code> service.                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <code>g-status</code>      | Displays the status of the ACSLS GUI.                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>l-status</code>      | Provides a verbose status summary of the various ACSLS services and includes pointers to log data for further analysis in troubleshooting contexts. The logs to which it points are helpful in contexts where the service failed to start up or shutdown.                                                                                                                                                                                                                                            |
| <code>p-status</code>      | Similar to <code>acsss status</code> , this report includes a listing of the various process IDs that are monitored by each respective service contract.                                                                                                                                                                                                                                                                                                                                             |
| <code>w-status</code>      | Shows the status of the <code>weblogic</code> service.                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>timeout</code>       | Reports the SMF start-up <code>timeout</code> for the <code>acsls</code> service on Solaris.                                                                                                                                                                                                                                                                                                                                                                                                         |
| <code>legal</code>         | Displays the ACSLS Legal Notice in English or French.                                                                                                                                                                                                                                                                                                                                                                                                                                                |

In most cases, you use only the top three commands: `enable`, `disable`, and `status`. The remaining commands are for convenience in contexts of servicing the software.

## bdb.acsss

The `bdb.acsss` utility backs up the ACSLS database contents and ACSLS control files that you must rebuild the ACSLS environment. The backup is either placed in a tar file or tape device you have named, or in the directory defined as the default.

This utility performs ACSLS database backups without requiring ACSLS to be down (hot backup).

Without the `-f` option, a `/export/backup/<time_stamp>.tar` file is created. `time_stamp` is the time the `bdb.acsss` command was executed. To verify the contents of the tape after running `bdb.acsss`, modify the following examples for your specific tape devices.

For Solaris:

```
tar tvbf 2048 /dev/rmt/0mn
```

After running the `tar tvbf` command, the tape advances to the next block. Make sure you have rewound or re-positioned the tape if you intend to run `rdb.acsss` after running the `tar tvbf` command.

## Format

```
bdb.acsss [-f backup_file | tape_device]
```

If you enter `bdb.acsss` with no options, a default backup is performed that provides you the ability to restore the database back to the time when this backup was performed.

## Options

- `-f backup_file`  
specifies a UNIX file to contain the ACSLS database backup. You must have write permissions to the file and directory.
- `-f tape_device`  
specifies any tape device attached and configured to the ACSLS server.

## Usage

Use the `bdb.acsss` utility to back up the ACSLS database to tape or to an external network file to create backups that can be used, if needed, for recovery of the database.

It is recommended that you use `bdb.acsss` to manually back up the database after:

- Running `acsss_config`.
- Importing the database. After you upgrade to a new version of ACSLS, *do not* use database backups created with previous versions.
- An audit of the entire library.
- Any database recovery.

### Example 1:

```
$ bdb.acsss -f /export/backup/my_backup
```

In this example, a file named `my_backup` was created in the directory `/export/backup`. You can now choose whether to keep the file, where it is located, or if to move it to another file system, another server, or a writable CD device.

This file can then be used to restore the database to the state it was in when the backup was performed. For example, if the backup was performed at 1:00 pm on Friday and a restore is

performed at 6:00 am on Monday, the database will revert back to its state as of 1:00 pm on Friday.

Within this same `-f` option, you can give a tape device instead of a file name and the backup will go to the named tape device.

**Example 2:**

```
$bdb.acsss -f /dev/rmt/0mn
```

In this example, a tape archive on the tape device `/dev/rmt/0mn` was created. This can be stored for later use in an offsite location.

See also:

- ["rdb.acsss "](#)
- ["Database Administration "](#)

## Dynamic Configuration (`config`) utilities

The dynamic configuration (`config`) utility allows you to implement configuration changes to ACSLS libraries (and components) while ACSLS remains online and running. These configuration changes are recorded in the `acsss_config.log` file.

The following dynamic configuration utilities are supported:

- `config acs`
- `config drives`
- `config lsm`
- `config ports`

Using the `config` utility provides the following benefits:

- ACSLS can continue running, allowing you to perform mount requests to unaffected library components.
- Allows you to reconfigure specified library components while all other configuration information remains unchanged. For example, when specifying:
  - an ACS, the configurations of other ACSs are not affected.
  - an LSM, the configurations of other LSMs are not affected.
  - a drive panel, the drives on a panel, mounts and dismounts to all existing drives are not affected.

It is important to recognize the following points:

- ACSLS must be running to use the dynamic `config` utility.
- You must use `acsss_config` to create your initial ACSLS configuration. Refer to [Setting Variables that Control ACSLS Behavior](#).
- Event notification reports all dynamic configuration changes.
- Before running dynamic configuration, ensure that all components being added or reconfigured are ready.
- The `acsss_config.log` file provides details regarding messages displayed.

- If you have not confirmed the configuration change, you can `cancel` the operation with `[[CTRL]]+C`.
- Dynamic configuration performs an automatic backup before and after the configuration change.
- After a configuration change is confirmed, it cannot be canceled. If you want to reverse a configuration change, shut down ACSLS and restore the backup that was taken immediately before the configuration change was made.  
You have 10 minutes to confirm a configuration change before it times out.
- You cannot remove the only (or last) ACS.
- Do not remove the last CAP in an ACS or the last drive defined to ACSLS.

## Limitations of Dynamic Configuration

The dynamic configuration utility has two significant limitations:

- You cannot delete an ACS, delete, or change a port (connection) to a library.
- With an existing SCSI/fibre-attached library, you can **only update** drive configurations in `config acs` and `config lsm` utilities. Panel or CAP configurations are not updated. The `config drives` and `config acs new` works with SCSI/fibre-attached libraries without limitation. The `config ports` utility is not supported on a SCSI/fibre-attached library.

Solution:

For these configuration changes that are not supported through dynamic configuration, you must bring down ACSLS and use `acsss_config`.

## Things You Should Not Do

- **Do not** use dynamic configuration to display status information for a library and its components because it involves extensive I/O to the library  
Use the `query` or `display` commands instead.
- **Do not** attempt to perform more than one configuration task at a time.  
Only one dynamic configuration task can be performed at a time. This:
  - Minimizes performance problems caused by the I/O between ACSLS and the library it is configuring
  - Avoids complex interactions between multiple configuration tasks.

## `config acs`

The `config acs` utility allows you to:

- Add an ACS or reconfigure an existing ACS and its components
- You can configure or reconfigure libraries without assigning all ACS numbers in sequence.

Example: You want to migrate from an SL8500 library to an SL3000 and then remove the SL8500 library. The SL8500 library is numbered ACS 0 and the SL3000 library is ACS 1. Using `config acs`, you can now migrate all of the cartridges and drives to the SL3000 library and later remove the SL8500 library without having to renumber your SL3000.

- Add the SL3000 library with `config acs acs_id new` without shutting down ACSLS.
- Move drives from the SL8500 library to the SL3000 library and update the drive configurations in both ACSs with `config acs acs_id`.
- Remove cartridges from the SL8500 and enter them into the SL3000 library.
- Finally, schedule an outage to shutdown ACSLS and remove the SL8500 from the configuration using `acsss_config`. Do not configure the SL8500. Remember to specify that the SL3000 is ACS 1 (not the default of zero).
- Configure a partitioned ACS.
- Add or remove LSMs, since the `config lsm` utility only allows you to reconfigure an existing LSM.

Each ACS must have at least one CAP. This can be a CAP that is shared with another partition. There must be at least one drive configured to the entire ACSLS system.

For example, if ACSLS supports four libraries three of the libraries can contain zero (0) drives. However, the fourth library must contain at least one drive.

## Format

- To add a new ACS, enter the following command:  
`config acs new`
- To reconfigure an existing ACS, enter the following command:  
`config acs acs_id`

## Adding a new ACS

To add a new ACS:

1. Enter the following command:

```
config acs new
```

2. Specify the ACS number for the new ACS.  
ACSLS lets you configure or reconfigure libraries without assigning all ACS numbers in sequence.

The ACS numbers already used and the first five available ACS numbers are displayed.

Enter the ACS number for your new ACS.

3. Select whether the ACS is in a partitioned SL8500 or SL3000.
  - If you enter `y`, you are asked for the partition ID for the ACS. This partition ID must match the partition ID on the SL Console.
  - If this is not a partitioned library, or is SCSI/Fibre-attached, enter `n`.

ACSLS does not support partitioned SCSI/Fibre-attached libraries, such as the SL150. Also, partitioned SCSI/Fibre-attached libraries do not have partition IDs.

4. Enter the number of connections to the ACS, followed by the device or host ID(s). You can have a maximum of fifteen connections.

 **Note:**

Ensure that all ports are connected to the same ACS.

The new ACS configuration is displayed.

5. Confirm the addition of the new ACS.  
After confirmation, the configuration information is displayed and the database is updated.

## Reconfiguring an existing ACS

The ACS should, if possible, be online or in diagnostic mode when you reconfigure the ACS.

To reconfigure the ACS:

1. Enter the following command:

```
config acs acs_id
```

The old and new configurations are displayed.

2. Confirm the new configuration.  
After confirmation, the database is updated.

If the configuration is unchanged, the configuration is displayed without asking for confirmation, and the utility exits. Examples of this are:

- Only drive types and/or serial numbers change
- LSM serial number changes
- Number of hands, such as SL8500 robots, change.

However, if these changes occur with other changes requiring confirmation, confirm the new configuration. The database is then updated.

The ACS and its components are removed from the database.

## Limitation of config acs

- With SCSI-attached libraries, `config acs` updates only the drive configuration. A SCSI-attached library must be IPLed to recognize drives that are added, removed, or changed. All drives must be ready when the library is IPLed.
- For a SCSI-attached library, `config acs` will not update panel or CAP configurations. In order to update them, you must bring down ACSLS and use `acsss_config`.

## config drives

The `config drives` utility allows you to reconfigure all drives on an existing drive panel. This includes, adding drives, updating drive types and serial numbers for existing drives, and deleting drives that were removed from the database.

Use the `config drives` utility for dynamic changes to drive configurations, which include installation, replacement, or removal of drives on an existing drive panel. Other changes to tape library hardware configurations, such as changes in the number and/or location of



storage cells, number or size of CAPs, or replacement of a storage cell panel with a drive panel must be made using the `config lsm` or `config acs` utility.

Notes:

- The LSM containing the panel with the changed drive configuration must be online or in diagnostic mode.
- On the drive panel being reconfigured, all drives must be ready.
- A SCSI-attached library must be IPLed to recognize drives that are added, removed, or changed. All drives must be ready when the library is IPLed.
- When new drives replace existing drives, varying the LSMs, drive types online or running `config drives` will automatically update the drive types and drive serial numbers.

## Format

`config drive panel_id` or `config drives panel_id`

## Usage

To reconfigure all drives on an existing drive panel:

1. Enter the following command:

```
config drive panel_id or config drives panel_id
```

The old and new drive configurations are displayed for the panel.

2. Confirm the configuration change.

After confirmation, the database is updated.

- If the configuration has not changed, the configuration is displayed without asking for confirmation, and the utility exits.
- If only drive types or serial numbers changed, the ACSLS database is updated without requesting confirmation.

## config lsm

This utility allows you to reconfigure an existing LSM and all of its components. These components include CAPs, panels and drives.

If you want to add or delete an LSM in an ACS, you must use the `config acs` utility.

Procedures for when Panels Change:

- If there are panels that are being removed or changed and have been emptied of cartridges, the LSM can remain online.
- If there are panels that are being removed or changed and contain cartridges, it is recommended that you `vary` the affected LSM to diagnostic until you reconfigure the LSM and audit the panels affected. If you do not do this, mounts and dismounts may fail.
- If you have added panels and have manually placed cartridges in these panels, run an audit to reconcile the database.

## Format

```
config lsm lsm_id
```

## Usage

To reconfigure the LSM:

1. Enter the following command:

```
config lsm lsm_id
```

The old and new configurations are displayed.

A "Y" next to the panel notifies you that the panel type(s) have changed. Look in the `acsss_config.log` file for details.

2. Confirm the new configuration.

After confirmation, the database is updated.

If the configuration is unchanged, the configuration is displayed without asking for confirmation, and the utility exits.

Minor changes are made automatically without confirmation. Examples are:

- Only drive types and/or serial numbers change
- LSM serial number changes
- Only LSM type changes between the 4410 and 9310
- Number of hands, such as SL8500 robots, change.

However, if these changes occur with other changes requiring confirmation, confirm the new configuration. The database is then updated.

## Limitation of config lsm

- With SCSI-attached libraries, `config lsm` updates only the drive configuration. A SCSI-attached library must be IPLed to recognize drives that are added, removed, or changed. All drives must be ready when the library is IPLed.
- It will not update panel or CAP configurations. In order to update them, bring down ACSLS and use `acsss_config`.

## config ports

The `config ports` utility allows you to dynamically add port connections to an ACS.

### Note:

All new ports must be connected to the same ACS as existing ports.  
Run `config acs acs_id` and then `config ports acs_id`.

To replace one ACS with another ACS or change port connection addresses, bring down ACSLS and use `acsss_config`.

## Format

```
config ports acs_id or config port acs_id
```

## Usage

To add ports:

1. Enter the following command:

```
config port acs_id or config ports acs_id
```

The current port connection for the specified ACS are displayed.

2. Enter the number of port connections for the specified ACS.  
You can have a maximum of fifteen connections.
3. Specify the device or host ID (s).

 **Note:**

Ensure that new ports are connected to the same ACS as existing ports.

The old and new configurations are displayed.

A change in the order of the ports is not a configuration change. The connections are displayed in the order they are currently recorded in the database.

4. Confirm the new configuration.

After confirmation, the database is updated.

If the configuration is unchanged, the configuration is displayed, and the utility exits.

## Limitations of config ports

The `config ports acs_id` utility is not supported on a SCSI/Fibre-attached library.

The `config ports` utility will not delete or change a port (connection) to a library. You must bring down ACSLS and use `acsss_config`.

## db\_export.sh

The `db_export.sh` utility exports the ACSLS database table data and ACSLS control files in preparation for an upgrade installation or a reinstallation of ACSLS.

**Note:**

The `db_export.sh` utility cannot run if ACSLS is running. Run `acsss disable` before running `db_export.sh`.

## Format

```
db_export.sh -f [ db_file | tape_device ]
```

## Options

- `-f db_file`  
specifies a UNIX file to contain a backup of the ACSLS database. Write permissions to both the file and directory.

**Notes:**

- If you export the database to a file, the file must reside in a non-volatile directory. If your intention is to re-install ACSLS, the re-installation will destroy the `$ACS_HOME` or `$ACSDB_BKUP` (such as `/export/backup`) directories. Place the exported file elsewhere in your file system.
- If you specify the filename without a path, `db_export` saves the database file under that filename in the current working directory. It saves the ACSLS control files in a file named `<filename>.misc` in the same directory.
- If you are exporting your database to an earlier release of ACSLS that does not support some of your tape libraries, tape drives, or cartridge media types, remove the unsupported tape libraries from your configuration, also the tape drives and cartridges from your libraries before exporting your database.
- `-f tape_device`  
specifies any tape device attached and configured to the ACSLS server.

If no options are specified, the system defaults to the tape device.

## Usage

Use the `db_export.sh` utility to prepare for a reinstallation of ACSLS or an upgrade installation of ACSLS.

See also:

- ["db\\_import.sh "](#)
- ["rdb.acsss "](#)
- ["Exporting the Database "](#)
- ["Migrating mchangers for Fibre to a New Platform "](#)

## db\_import.sh

The `db_import.sh` utility imports the ACSLS database table data and the ACSLS control files that you exported when you used the `db_export.sh` utility. The ACSLS control files are

located in `$ACS_HOME/data/external` and consist of user definable variables and configuration for ACSLS. They specify Access Control settings, scratch preferences, Extended Store LSMs, custom `volrpt` settings, volume attributes (for `watch_vols` utility), and so on. The `db_import.sh` utility also provides disaster recovery capabilities and the retention of your customized dynamic variables when moving to a different operating system or from an earlier release.

## Format

```
db_import.sh -f [ db_file | tape_device ]
```

## Options

- `-f db_file`  
specifies a UNIX file created by `db_export.sh`.
- `-f tape_device`  
specifies any tape device attached and configured to the ACSLS server.

If no options are specified, the system defaults to the tape device.

## Usage

Use the `db_import.sh` utility to import the ACSLS database that you exported using the `db_export.sh` utility.



### Note:

The `db_import` utility will not run if ACSLS is running. Run `acsss disable` before running `db_import.sh`.

See also:

- ["db\\_export.sh "](#)
- ["rdb.acsss "](#)
- ["Exporting the Database "](#)
- ["Migrating mchangers for Fibre to a New Platform "](#)

## db\_restore.sh

The `db_restore.sh` utility restores the ACSLS database using an existing database backup file.

## Format

```
db_restore.sh [latest | backup_filename]
```

## Options

- `latest`  
indicates that the database be restored from the most recent backup stored in `$ACSDB_BACKUP_DIR`.
- `backup_filename`  
specifies the path to a valid backup file to be used for the database restore. To see a list of valid backup files, list the contents of the directory `/export/backup`.

If no options are specified, the backup will appear in the backup directory which was defined when ACSLS was installed on the system.

## Usage

Use the `db_restore.sh` utility to restore the ACSLS database using an existing database backup file.

### Note:

- The `db_restore` utility will not run if ACSLS is running. Run `acsss disable` before running `db_restore.sh`.
- If you do not want to use an existing database, and would rather create a new one, then run `acsss_config`.

## del\_vol

The `del_vol` utility looks for a volume in the library. If it cannot find the volume, `del_vol` either marks the volume missing or absent, or deletes it from the database, depending on your request.

If all of the referenced locations for the volume cannot be verified (such as the LSM is offline or the drive is not ready), you are prompted to confirm marking the volume absent or deleting it, unless the `-n` (`no_confirm_flag`) is on. If the volume is deleted, the volume and information associated with it, such as scratch pool membership and current and pending locks, are removed from the database.

Volume-related information is displayed unless the `-q` (`quiet_flag`) has been entered. If multiple options are used, they can be formatted either as separate options, or as a contiguous string.

Before marking a volume absent or deleting it, `del_vol` asks you to confirm, unless the `-n` option is specified.

- If the volume is in the library, it remains an active volume in the database.
- If the volume cannot be located in the library, it is marked absent, unless you specify that the volume should be deleted.

- If cells or drives where the volume may be located are inaccessible (because libraries or drives are offline or inoperative), the volume is marked missing, unless you specify that the volume should be deleted.

Notes:

- This utility does not delete a volume that is found in the library.
- You can use the `del_vol` utility to remove a volume record without waiting for the expiration of an absent or ejected status.
- ACSLS must be running (enabled) to support the `del_vol` utility.

## Format

```
del_vol [-n] [-d] [-f] [-q] vol_id
```

## Options

- `-n`  
no-confirm mode; marks absent or deletes a volume that cannot be located without prompting the user for confirmation.
- `-q`  
quiet mode; do not print out all information extracted from the database.
- `-d`  
deletes the specified volume without waiting for the expiration of an absent or ejected status.
- `-f`  
permits you to re-enter cartridges that were stuck in a tape drive. With the `-f` option, you can delete a volume or mark it absent without verifying if the volume is actually in the library. This permits you to delete from the database a volume that was in a faulty tape drive. Using this option, you can remove the volume from the drive, delete it from the database, then re-enter the volume for normal library use while the drive is being repaired.
- `vol_id`  
The volume serial number to be deleted.

 **Note:**

If the volume serial number contains a dollar (\$) sign, enclose it in single quotes. For example: `del_vol 'AB$001'`

## Usage

You can use `del_vol` to remove a cartridge from an offline LSM and then re-enter it in an online LSM, so it can be automatically mounted.

- Remove the cartridge from the offline LSM.

- Use `del_vol` to mark the cartridge as absent.
- Enter the cartridge into the online LSM.

ACSL S and the database must be up and running (*not* idle) to use this utility.

**Note:**

If you mistakenly delete a cartridge from the database, you should audit the subpanel containing the home cell of the deleted cartridge to re-enter the cartridge into the database. Running `del_vol` while the system is in recovery can produce unpredictable results. The recovery sequence also happens during a `vary LSM online`.

## Example

To delete cartridge `U01102` without a printout of the cartridge information:

```
del_vol -q U01102
```

## drives\_media.sh

This routine displays all drive types, media types and drive-to-media compatibility that is supported by the current release of ACSLS. The information is normally displayed to standard output.

Refer to the *ACSL S Product Information Guide* for the current list of libraries, drive types, media types and drive-to-media compatibility supported.

## Format

```
drives_media.sh [ -f, -h ]
```

## Options

- `-f`  
the information is written to three files:
  - `/tmp/drive_types.txt`
  - `/tmp/media_types.txt`
  - `/tmp/media_compatibility.txt`.
- `-h`  
displays the syntax message.

## ejecting.sh

The `ejecting.sh` utility facilitates mass eject vaulting operations. Working from a specified list of CAPs and volumes, this utility executes multiple eject operations in parallel until the overall job is complete. Unless the user requests volumes to be ejected in sorted order, this utility ejects each volume to its nearest specified CAP.



Where possible, nearby CAPs with free space are used instead of the closest CAP, if the operation can reduce unnecessary CAP manipulation by the operator. The general approach reduces cartridge movement, eliminates unnecessary LSM pass-thru migration, and reduces the operators' overall workload.

All eject jobs are monitored and summarized in the single shell window where the operation was launched. See the `-x` option below for use with multiple XTERM windows. Users are advised each time a specific CAP is full and ready for the operator to Remove cartridges from CAP. The operator is notified when the overall operation is complete.

A set of logs showing the results of all eject operations for the past ten days is kept in the `$ACS_HOME/log/ejectingLogs` directory. Each individual log is identified by a time stamp taken when the operation completes.

## Format

Standard: `ejecting.sh [-dmox] -c <CAP list> -v <volume list file>`

Policy Specification: `ejecting.sh [dmox] -p <policy file>`

Legacy Format: `ejecting.sh <CAP ID> <volume list file>`

## Options

- `-c <CAP list>`

A list of CAP ID's separated on the command line by spaces. All CAPs in the CAP list must be in the same ACS.

**Example:** `-c 0,1,0 0,1,1 0,5,0 0,5,1 0,9,0`

Wildcard expressions using an asterisk (\*) are valid to specify all LSMs or all CAPs in an LSM, or both.

**Examples:**

- `-c 0,1,*` (All CAPs in LSM-1)
- `-c 0,*,0` (CAP-0 in every LSM)
- `-c 0,*,*` (All CAPs in every LSM)

Only CAPs that are online and available are selected for the operation. When wildcard expressions are specified, only nonzero priority CAPs are selected.

The ACS must be a numeric expression and cannot be referenced by a wildcard.

In larger library contexts, CAP selection can make a big difference with regard to the speed and efficiency of vaulting operations. Specifying too many CAPs may add unnecessary work for the operator to service multiple partially-filled CAPs. Specifying too few CAPs for a given work load can introduce bottlenecks and thereby increase robotics wait time. As a general rule of thumb for large volume lists, the CAPs you choose should be dispersed across the library complex on different rails and in different library modules. When selecting a few CAPs for a large number of volumes, divide the library into regions and choose a CAP located in the center of each region.

- `-v <volume list file>`

This is a file specification using full or relative path name to a simple text file. The file should contain a list of VOL-IDs (VOLSEERS) representing the volumes to be ejected. Only the volumes that are contained in the same ACS as the specified CAPs will be affected. Mounted volumes will not be ejected.

- `-p <policy file>`

The policy file is specification using a full or relative path name. This text file contains a defined policy for CAPs and volumes. The format of the file includes the word "caps:" followed by a list of CAPs to use for the operation; and the word "vols:" followed by the full path name of the volume list file.

**Example:**

```
caps: 0,1,0 0,1,1 0,5,0 0,5,1 0,9,0 0,9,1
vols: /export/backup/volumes_to_eject.txt
```

- `-d`

The `display` option instructs `ejecting.sh` to show the volume-to-cap assignments before executing the `eject` operation. You may choose to display the full list of volumes and the CAPs to which they move, or simply display a summary showing how many volumes migrate from each LSM to each CAP.

After viewing the display, you can choose to continue or to abort the operation.

- `-m`

Label the job with an operator message code between "04" and "99". On supported libraries, this numeric code will be displayed on the operator console.

- `-o`

There are times when vaulting operations require volumes to be stacked in a sorted order. With this option, the routine moves the specified volumes to the listed CAPs according to the order they are found in the volume list, and according to the specified CAP order. The CAP order is repeated from the first CAP to the last, until all volumes have been ejected.

 **Note:**

Since volume to CAP order takes precedence, this option makes no attempt to optimize volume movement by limiting LSM pass-thru routes.

- `-X`

Use a dedicated XTERM window for each discrete CAP `eject` payload. You may find this option useful to keep track of individual eject jobs during the overall mass-eject operation. An XTERM session pops up as each CAP `eject` is launched and disappears when the corresponding `eject` has completed.

When operating from a Windows terminal, make sure that X11-capable software is installed. X11 is standard on Solaris or Linux. You must open display access control to the ACSLS server from the local machine.

**Example:** `xhost + <acsls_server_hostname>`

This utility looks at your login identity (`who am i`) to determine where to send the DISPLAY. You must login directly from your local console or desktop machine to the ACSLS server in order to see the display.

### Legacy Format

```
ejecting.sh <CAP ID> <volume file>
```

The legacy form of this utility has been preserved. It takes a single CAP ID and a path name to a file containing a simple list of volumes. It then composes a series of `eject` commands optimized to the size of the CAP, and displays the resulting commands to standard out. The display includes as many `eject` commands as are necessary to eject the full volume list through the specified CAP.

The operator can pipe the output of legacy `ejecting.sh` to directly to `cmd_proc` to execute the operation.

### Example:

```
ejecting.sh 0,1,0 /export/backup/myVolumeList | cmd_proc -lq
```

Alternatively, the output can be redirected to a file which can later be redirected to `cmd_proc`.

### Example:

```
ejecting.sh 0,1,0 /export/backup/myVolList > /tmp/eject.dat cmd_proc -lq  
< /tmp/eject.dat
```

## ejecting.sh Logs

Every instance of `ejecting.sh` is logged in the directory, `$ACS_HOME/log/ejectingLogs/`. Each log file is named with a date-time stamp. For example:

```
ejecting.log.14-Oct_13:13:10
```

Each `ejecting.log` summarizes the overall ejecting operation as it was seen from the operator shell. Any errors that were encountered are seen here.

An `ejecting.log` may contain a listing of volumes that are ignored by the utility because:

- The volume ID is invalid.
- The volume is not contained in the specified ACS.
- The volume is in use.

The accumulating files in the `ejectingLogs/` directory are purged after ten days. Logs older than ten days are removed with each new invocation of `ejecting.sh`.

## free\_cells.sh

The `free_cells.sh` utility allows you to monitor and manage the free cells in libraries managed by ACSLS. This utility reports the free cell count for LSM, ACS, and the ACSLS server.

This utility is located in the `$ACS_HOME/utlils` directory.

## Format

```
free_cells.sh
```

## Option

- -a

displays the free cells, allocated cells and the total number of cells in each ACS and LSM and those managed by the ACSLS server.

For example:

```
LSM 1,3
Total free cells = 3,345
Allocated cells = 3,155
Total cells = 6,500
```

## getHba.sh

The `getHba.sh` utility manages Fibre Channel HBA ports.

## Format

```
getHba.sh
```

The `getHba.sh` utility is run at install time and is called by `install_acsss.sh` which is called by `install.sh`. The utility can be run directly any time a new HBA has been added to the system or any time the HBA ports are re-arranged. This utility identifies an appropriate HBA port to change from initiator to target mode, in order to reveal a client access point to ACSLS logical libraries.

### Note:

`getHba.sh` can only be run if a previous logical library was installed, or the correct target mode package was installed using `install.sh`. Refer to the *ACSLS Installation Guide* for information about running `install.sh`.

The best way to use this utility to set up your FC connections before `getHba.sh` is run. This enables `getHba.sh` to show you useful information about the existing connections.

The utility first assesses whether a target mode adapter has already been configured. If no target ports exist, the flow of the utility continues as discussed below. If the utility senses any existing target port, it displays the following menu of choices.

Select a desired action:

1. Keep the existing HBA port configuration.
2. Configure an additional target-mode port.
3. Restore an existing target port to initiator mode.

| Option | Description                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1      | Exits this utility.                                                                                                                                                                                                                                                                                                                                                                                               |
| 2      | The utility lists the ports that are currently operating in initiator mode. When a port shows "Connected to a remote HBA", it's means there is an initiator at the other end, making it a potential candidate to become an ACSLS target port. When a port shows "Connected to a target device", there is probably a tape library or disk attached, so that port would be a bad choice for target mode operations. |
| 3      | Identifies each port configured for target mode operations and prompts you for confirmation to restore that port to initiator mode.                                                                                                                                                                                                                                                                               |

### Example of Option 2

Select which local HBA port is to be changed to Target mode. Select from the following list:

1. HBA Port WWN 2100001b32055d85 Not connected.
2. HBA Port WWN 2101001b32255d85 Connected to a remote HBA.
3. HBA Port WWN 2102001b32055d85 Connected to a target device.
4. None of these.



#### Note:

If you have no intention of using the logical library feature, select "none of these".

Once you make the selection, you are asked for confirmation.

```
2
HBA Port WWN 2101001b32055d85 /pci@0,0/pci10de,377@f/pci1077,143@0
Is this correct? (y or n):
```

You are given the opportunity to change your mind. If you respond "n," the list of available ports are again displayed with a prompt for your selection. If you respond "y," you are asked if there are additional ports you want to reconfigure.

The utility will proceed to add a target group and a target group member, and you are instructed to reboot the server for the target-mode changes to take effect.

Creating Target Group: 2101001b32255d85

### Example of Option 3

This option enables you to unconfigure an existing target and restore the HBA to its native mode as an initiator.

```
# cd $ACS_HOME/install
# ./getHba.sh
A Target-mode port has already been configured:
  Target: wwn.2100001B32050A28
    Connected to ...
      Initiator: wwn.210100E08BA61A29
Please select a desired action:
1) Keep the HBA port configuration as it is.
```

```
2) Configure an additional target-mode port.
3) Restore a target port to initiator mode.
3
Target: wwn.2100001b32050a28
Do you wish to restore this port to initiator mode? (y or n): y
Removing 'qlt' binding in /etc/driver_aliases
Are there additional ports you wish to reconfigure? (y or n): n
A reboot will be necessary for these changes to take effect.
```

## greplog

Use the `greplog` utility to filter the `acsss_event` log to include or to exclude messages containing specific keywords. The syntax of this routine is similar to the UNIX `'grep'` function. `greplog` is specifically designed for use with the `acsss_event.log`, but it may function with any type of message file where the records are separated by an empty line.

## Format

```
greplog -[v|i] <keyword> <logfile>
```

## Options

- `-v`  
Optional. The option displays all the messages in the log except those which include the keyword.
- `-i`  
Optional. This option ignores the case of characters in the specified keyword.
- `-keyword`  
returns the complete multi-line message containing the keyword.
- `-logfile`  
list of log files.

## Usage

Since the utility is specifically designed for log files, `greplog` returns the complete multi-line message containing the keyword rather than a single line containing that word. Using the `-i` option, `greplog` ignores the case of characters in a specified keyword. Using the `-v` option, `greplog` displays all of the messages in the log except those which include the keyword. `greplog` is specifically designed for use with the `acsss_event.log`, but it may function with any type of message file where the records are separated by an empty line.

## install\_scsi\_Linux.sh

The `install_scsi_Linux.sh` utility creates `/dev/mchanger*` links which can be used when configuring libraries to ACCLS. Those `mchanger` names are now constructed using a serial number reported by the library, providing ACCLS with a reliable identifier that persists across changes in the SAN fabric or server reboots (both of which can change the underlying device paths for a library).

Information about the resulting `/dev/mchanger` links and the associated libraries is displayed as part of the script output, using the `showDevs.sh` utility. That utility can also be run as a stand-alone operation (after the `mchanger` links have been created) to display library information.

## Format

```
install_scsi_Linux.sh
```

### Sample Output:

```
=====
# install/install_scsi_Linux.sh
[root@acslsdevx1 install]# ./install_scsi_Linux.sh
Installing SCSI device(s) for Oracle StorageTek ACSLS.
Adding ACSLS rules for udev ...
Starting udev:                                     [ OK ]
Successfully built the following...
  /dev/mchanger-3500104f00079f9d2: STK SL500 V-1485 336-cells 10-drives
  /dev/mchanger-3500104f0007a8532: STK SL500 V-1485 205-cells 6-drives
  /dev/mchanger-3500104f000cc6a67: STK SL150 V-0182 59-cells 4-drives
Installation of SCSI device(s) successfully completed.
#
=====
# utils/showDevs.sh
  /dev/mchanger-3500104f00079f9d2: STK SL500 V-1485 336-cells 10-drives
  /dev/mchanger-3500104f0007a8532: STK SL500 V-1485 205-cells 6-drives
  /dev/mchanger-3500104f000cc6a67: STK SL150 V-0182 59-cells 4-drive
#
=====
```

## lib\_type.sh

This routine returns the LSM type of the LSMs attached to the specified ACS ID. If multiple LSMs of a common type exist in the configuration, then only a single type is returned for the multiple LSMs.

## Format

```
lib_type.sh <ACS ID>
```

## moving.sh

The `moving.sh` utility moves multiple cartridges to one or more LSMs. This utility reads a file that lists the cartridges to be moved. These cartridges can be:

- Cartridges in one or more LSMs
  - Cartridges on a panel being moved to other panels in the same LSM or other LSMs
  - Any group of cartridges you select

The limitations of `moving.sh` are:

- All destination LSMs and cartridges in the `vol_list_file` must be in same ACS

- If any destination LSM is offline or does not contain any free cells, no cartridges are moved to that LSM

**Notes:**

- The `moving.sh` utility runs only if ACSLS is running
- Internally, `moving.sh` moves only one cartridge at a time to avoid impacting library performance (mounts and dismounts)
- You can run multiple move utilities in parallel after creating separate lists of volumes. Ensure that:
  - the destination LSM is same. Make sure that there are enough free cells in the LSM to accommodate all cartridges
  - you are moving within one SL8500 library - there are only two elevators, so running more than two move utilities at a time will not increase performance

## Format

```
moving.sh -f vol_list_file -t lsm_id or list of lsm_ids
```

**Where:**

- `-f vol_list_file`

The name of the file containing the list of volumes to be moved.

 **Note:**

The volume IDs must follow these rules: One cartridge-id per line; The `vol_ids` must be valid ACSLS volume IDs; If the `vol_ids` include trailing or leading spaces, they must be enclosed within single quotes or double quotes.

- `-t lsm_ids`

Specifies one or more LSM IDs to which the cartridges are moved. Each LSM ID should be separated by a space and belong to the same ACS.

Use the `moving.sh` utility to move a list of cartridges to other LSMs or from one panel to other panels in the same LSM.

You can use either a custom volume report or the `display volume` command to create a file containing the list of volumes to be moved from an LSM.

You want to use the `moving.sh` utility:

- When a SL8500 is initially partitioned or re-partitioned, and one or more rails (LSMs) are removed from an existing partition (ACS), `moving.sh` can move cartridges from the LSM being removed from the partition to the LSM(s) that remains in the partition.
- When any LSM(s) is removed from an ACS, `moving.sh` can move cartridges to the LSMs that remain in the ACS.

For example, if SL8500s are removed from a library complex (ACS), `moving.sh` moves cartridges from the SL8500s that are being removed to the LSMs that will remain in the library.



- When a storage expansion module(s) is removed from an SL8500, cartridges can be moved from the panels being removed to the panels that remain in the library.
- To optimize library performance, move inactive cartridges to an LSM with few or no drives that are used to archive cartridges. This frees up space in LSMs with drives for new, active cartridges.

## Creating the Volume List File

Before you begin, you must create a file that contains the list of volumes to be moved from an LSM. You can use either the *volrpt* (custom volume report) or the *display volume* command.

- Create a *vol\_list\_file*

```
volrpt -d -f custom_volrpt_input -l lsm_id > vol_list_file
```

Where the *custom\_volrpt\_input* file is:

```
VOLUME_ID 6
```

Sample output:

```
$ volrpt -d -f my_custom -l 0,2 > my_file_list
$ cat my_file_list
ABC744
ABC748
ABC756
ACS151
EN0823
000373
```

- Using the *display volume* command to create the *vol\_list\_file*

1. Display the list of volumes.

**Example:**

```
display volume * -home acs,lsm,panel,*,* -f vol_id
```

This example selects all volumes on the panel identified by the *-home* parameter. Row and column are wild-carded. Only the *vol\_id* is output.

Sample output:

```
ACSSA> display volume * -home 0,3,5,*,* -f vol_id
2007-02-12 15:31:45          Display Volume
Vol_id
PG0350
PG0353
PG0356
PG0358
PQ0616
```

2. Create and name your *vol\_list\_file*.
3. Cut and paste the list of volumes (created from the *display* command) into this file.
4. Edit the output.  
The *vol\_list\_file* cannot contain any blank lines and leading spaces. Use the following *vi* command to eliminate them.

```
:%s/^[ ]*//g
```

If you do not do this, you get an error message as shown in the following example.

```
$ moving.sh -f my_file_list -t 0,2
Error in file my_file_list.
Invalid entry
ABC748
ABC756
ACS151
EN0823
```

This error message was generated because there was an extra space before volumes ABC748 and 756.

## Moving Cartridges Before Removing an LSM from an ACS

After a library is reconfigured or re-partitioned, and if an LSM is removed from an ACS, all cartridges in the LSM become inaccessible. Therefore, before the LSM is removed, all of its cartridges should be moved to LSM(s) that will remain in the ACS. Use the following procedure:

- When a rail (LSM) is removed from a legacy partition in a partitioned SL8500.
  - When an LSM(s) is removed from an ACS. The ACS can include 9310s or an SL8500 library.
1. Plan your new configuration.
    - Organize the cartridges and drives for performance.
    - Empty an LSM shortly before you change the library configuration.
    - Determine how many cartridges you have in the LSMs that you are emptying, and how many free cells in the LSMs to which you are moving cartridges.

Use `free_cells.sh -a` to find out the number of cartridges in these LSMs (allocated cells) and free cells.
  2. Schedule the move and re-configuration.
    - Schedule the move to minimize the impact on your system.

Moving the cartridges takes time, and reconfiguring a library or re-partitioning an SL8500 is disruptive.

    - Make sure there are enough free cells in the target LSM(s) for the cartridges being moved. If needed, eject cartridges to free up space.
  3. Vary all of the drives in the LSM being removed offline.  
This prevents the following:
    - Contention for robots in the LSM.
    - Mounts to this LSM.

Otherwise, cartridges mounted to this LSM can float to new home cells in the LSM, filling up the LSM that you are trying to empty.
  4. Vary the LSM being emptied to diagnostic mode to restrict access to only the `cmd_proc` using the following command:

```
vary lsm lsm_id diagnostic
```

**Example:** `vary lsm 0,1 diagnostic`

5. Run a custom `volrpt` to output all of the cartridges in the LSM being emptied to a file, using the following command:

```
volrpt -f custom_volrpt_input -l from_lsm_id > move_vols_list
```

Where the `custom-volrpt_input` is:

```
VOLUME_ID 6
```

Example:

```
volrpt -f volrpt_input -l 0,1 > move_vols_list
```

See [Creating a Logging Volume Statistics Report](#) for more information.

6. Move the cartridges out of the LSM being emptied, using the following command:

```
moving.sh -f move_vols_list -t dest_lsm_id(s)
```

7. Check that the LSM is empty using `volrpt` since the cartridges may have been entered into the LSM or may have “floated” into it.

```
volrpt -l from_lsm_id
```

If it is not empty, run the custom `volrpt` again to select the volumes that are now in the LSM. Then, run `moving.sh` again (steps 3 and 4).

 **Note:**

**Do not** run `moving.sh` again with the original list of volumes.

8. Vary the LSM being emptied offline to prevent volumes from being moved to it.

```
vary lsm lsm_id offline
```

 **Note:**

Remove the LSM from the partition and/or ACS.

9. Reconfigure the ACS using either `config acs acs_id` or `acsss_config`.

## Moving Cartridges Before Removing Cells from a Partition

### Note:

The SL3000 can partition down to the drive and cell level, and an SL8500 can partition to the drive and cell array level with enhanced partitioning. If cells are reassigned from one partition to another, the cartridges in those cells will be orphaned, and will no longer be accessible by the partition that they were in before. The host managing the other partition could write over the data on the cartridges.

To prevent cartridges from being orphaned when partition boundaries change: before re-partitioning the library move them to cells that will remain in the partition. Since SL3000 is a single LSM, the existing ACSLS move command does not work. You would just move them somewhere else in the library. You might also move them to another cell that will also be removed from the partition.

Use one of the following methods to move your cartridges:

- Use the StorageTek Library Console (SL Console).  
Audit the library to audit your volume's locations.  
Refer to the *SL8500 or SL3000 User's Guide* for detailed information and procedures.
- Use the following ACSLS procedure:
  1. Use "`volrpt`" or "[Using display Command Options](#)" to display volume locations.
  2. Display a list of available (empty) cells in a specific panel using the following display command:
 

```
display cell a,l,p,*,* -status empty -f status
```

 For more information, refer to "[Using display Command Options](#)".
  3. Move the cartridges to a specific cell by specifying a free cell instead of an LSM ID. For a cell move, use the move command:
 

```
move AAAAAA a,l,p,r,c
```

## Moving Cartridges Before Changing or Removing Panels

You must move the cartridges before changing a cell panel to a drive panel in a 9310 or removing a storage expansion module in an SL8500.

Steps 1 - 4 as procedures as [Moving Cartridges Before Removing an LSM from an ACS](#) .

Step 5: Select the cartridges in the panel being emptied, and output them to a file.

1. Run a custom `volrpt` to output all of the cartridges in the LSM being emptied to a file. Include the panel number (in the home cell ID).

```
volrpt -f custom_volrpt_input -l from_lsm_id > move_vols_list_1
```

Where the `custom-volrpt_input` is:

```
VOLUME_ID 6
CELL_ID    14
```

Select the volumes in the panel(s) being emptied and output these `vol_ids` to your `move_vols_list_2`.

2. Select the cartridges in a panel being emptied using the `display volume` command.

```
display volume * -home acs,lsm,panel,*,* -f volume > move_vols_list_2
```

This selects all volumes on the panel identified by the `-home` parameter. The row and column are wild-carded. Only the `vol_id` is output, and the output writes to the file.

Edit the output, removing any leading spaces and the trailing blank line.

### Note:

If the destination or "to" LSM is the same as the source or "from" LSM and more than one panel is being emptied, some volumes are moved back to the panels being emptied. You will have to select the volumes off the panels and move them repeatedly to clear out the panels.

Steps 6 - 9 same procedures as [Moving Cartridges Before Removing an LSM from an ACS](#).

Step 10. Reconfigure the LSM, using either `config lsm lsm_id` or `acsss_config`.

## Examples

- Moving cartridges from LSM 0,4 to LSM 0,0 and 0,1

To move cartridges from LSM 0,4 to LSM 0,0 and 0,1, create a file containing the list of cartridges in LSM 0,4 using `volrpt`, and then run the `moving.sh` utility as below:

### Sample output:

```
$ moving.sh -f vol_list.txt -t 0,0 0,1
Number of free cells in LSM 0,0 : 308
Number of free cells in LSM 0,1 : 362
-----
Total number of free cells : 670
Total number of volumes to move : 7

Cartridge CAB001 moved to 0,0,3,0,0
Cartridge CAB002 moved to 0,0,4,0,0
Cartridge CAB003 moved to 0,0,5,0,0
Cartridge CAB004 moved to 0,0,6,0,0
Cartridge CAB005 moved to 0,0,7,0,0
Cartridge CAB006 moved to 0,0,8,0,0
Cartridge CAB007 moved to 0,0,9,0,0

Summary
=====
Number of free cells remaining in LSM 0,0 : 301
Number of free cells remaining in LSM 0,1 : 362
```

```
-----
Total number of free cells remaining : 663
Number of cartridges moved : 7
Number of cartridges not moved : 0
```

- Moving cartridges from LSMs 0,4 0,5 0,6 and 0,7 to LSMs 0,0 0,1 0,2 and 0,3,

To optimize performance by moving each LSM to the adjacent LSM:

- Prepare files containing the list of cartridges in LSM 0,4 0,5 0,6 and 0,7 using `volrpt`.
- Run four `moving.sh` utilities at the same time but in separate UNIX command terminals.

There is no contention between the separate instances of `moving.sh` because the source and destination LSMs and the pass-thru ports used are all different:

Sample Output:

```
moving.sh -f vol_list_0-4.txt -t 0,0
moving.sh -f vol_list_0-5.txt -t 0,1
moving.sh -f vol_list_0-6.txt -t 0,2
moving.sh -f vol_list_0-7.txt -t 0,3
```

The `moving.sh` utility can be used to move inactive cartridges to archival LSMs. An archival LSM is an LSM with few or no drives that stores cartridges that have a low probability of being mounted. The top rail in an SL8500 is a good choice for an archival LSM because it does not have direct access to the CAP.

Inactive cartridges that do not need to be in a library can be ejected, while inactive cartridges that still need to be available for automated mounts should be moved to archival LSMs.

To move inactive cartridges to an archival LSM, complete the following procedure:

1. Identify the inactive cartridges. For example, to select cartridges that have not been accessed in the last three months:
2. Run a custom `volrpt` to output all of the cartridges in the LSM being examined, and output the results to a file.

```
volrpt -f custom_volrpt_input -l from_lsm_id > move_vols_list_1
```

Where the `custom-volrpt_input` is:

```
VOLUME_ID 6
ACCESS_DATE 15
```

3. Select the cartridges where the `access_date` is earlier than three months ago and output these `vol_ids` to a file with the list of volumes to be moved.
4. Move the inactive cartridges to the archival LSM.

```
moving.sh move_vols_list_2 archival_lsm_id
```

See:

- [Using display Command Options](#)
- `volrpt`

## probeFibre.sh

This utility displays all direct-attached or SAN-attached libraries behind a contemporary fibre-channel HBA.

The `probeFibre.sh` utility displays the model number, LUN ID and World Wide Port Name (WWPN) of each fibre-attached library. The `probeFibre.sh` utility can be run even before the `mchanger` devices are created for each library.

This utility requires `root` access.

### Format

```
probeFibre.sh [-v] [-p]
```

### Options

No argument.

displays vendor, model, LUN ID and WWPN for each library device.

- `-v`  
produces a structured output that includes the model number of the host bus adapter (HBA) and the WWPN of each initiator port, along with the library devices detected on each port (including WWNN).
- `-p`  
produces an output that includes  
`vendor:model:version:driver:target:lun:wwpn` with each field delimited by a colon.

## rdb.acsss

The `rdb.acsss` utility restores the ACSLS database and the ACSLS Control files using a backup created by either the automatic backup function or the `bdb.acsss` utility. The ACSLS Control files are located in `$ACS_HOME/data`, and define several different environmental variables for ACSLS. They specify Access Control settings, scratch preferences, Extended Store LSMs, custom `volrpt` settings, and volume attributes (for `watch_vols` utility), and so forth.

If you are restoring from a tape backup, be sure to rewind or position the tape device before restoring the ACSLS database and control files from tape. Use one of the following commands to rewind or position the tape to the exact location where the backup files resided before running `rdb.acsss`.

```
mt -f /dev/rmt/0mn rewind  
mt -f /dev/rmt/0mn nbsf 1
```

### Format

```
rdb.acsss
```

## Menu Options

When you run `rdb.acsss`, a menu displays four options, as shown in the example below.

Please enter the number followed by Return for your choice from the following menu.

Press? followed by the Return key for help.

- 1: Restore from a list of current local disk backup files
- 2: Restore from a previous tape or file backup
- 3: Restore database only (do not include ACSLS control files)
- 4: Restore only ACSLS non-database control files
- E: Exit

1. Restore from a current local disk backup.  
All of the current ACSLS backup files on the local disk are listed.

**Explanation:** The database is restored to the backup. ACSLS Control files are restored from the backup only. The backups are saved in the default backup directory (`$ACSD_BKUP`). The database is restored to any database backup listed and selected. Usually there are 8 different dates listed, but this varies depending on the database retention period set in `acsss_config`.

**Usage:** Use this option to restore a corrupted database. With this option, all backups are displayed and you can restore to any displayed database backup.

**Example:**

```
Menu choice: 1
rcvr_previous.sh 2642: ACSLS database recovery started.
You have taken backups on the following days. Please enter the corresponding date
and time to the backup that you wish to recover from. ACSLS database and control
files will be restored to that time.
2011-10-02 04:38:48
2011-10-03 00:00:01
2011-10-04 00:00:01
2011-10-05 00:00:01
2011-10-05 11:49:06
Please enter the recovery date and time (YYYY-MM-DD HH:MM:SS):
HINT: You may copy and paste to enter the date and time.
```

You must enter the desired date and time from the relevant backup and the database is restored to that point.

2. Restore from previous tape or file backup  
**Explanation:** Select this option to recover a database that was copied to a different file system (such as NFS) or to a backup device (such as tape). ACSLS Control files are restored.

**Usage:** Use for a catastrophic event such as hardware failure when the database must be restored to either the server or even an entirely different server. The platform (OS version/update and ACSLS release/PUT level) must be the same.

```
Option 2:
Menu choice: 2.
rcvr_manual.sh 2635: ACSLS recovery started
To recover the ACSLS environment either:
- Mount a ACSLS backup tape in a tape device and
  specify this tape device with '-f tape_device', or
- Specify a file name containing a ACSLS backup with '-f backup_file'.
```



The ACSLS database will be recovered from the file specified.

Please enter `-f [ backup_file | tape_device ]`:

**Example 1: Specifying a file with `-f backup_file`**

Please mount tape (if used) and enter backup source: `-f /export/backup/my_backup.bak`

This would restore a backup called `my_backup.bak`. Both database and ACSLS Control files would be restored and ACSLS would be put back in the state that it was when the backup was run.

**Example 2: Restoring a backup created on a tape device**

Restoring a backup created on a tape device uses the same option but works a little differently. When a backup is created to a tape device, the tar archive is created on the tape but it does not have a name. When restoring a backup from a tape, only the tape device is given.

**HINT:** You should use a **no rewind** tape device.

Please mount tape (if used) and enter backup source: `-f /dev/rmt/0mn`

This goes to the device `/dev/rmt/0mn` and verifies that there is a valid database backup. If it does exist and is valid, it is restored.

Procedure to Rewind a Tape:

The tape must be **REWOUND** or **POSITIONED** at the correct location where the backup files reside before `rdb.acsss` is attempted.

 **Note:**

After running `tar tvbf` command, the tape is advanced to the next block. Make sure you have rewind/repositioned the tape if you intend to run `rdb.acsss` after running the `tar tvbf` command.

- a. The tape can be rewind/positioned using the following command:

```
mt -f /dev/rmt/0mn rewind or mt -f /dev/rmt/0mn nbsf 1 ---> SOLARIS
```

- b. To verify the contents of tape after `bdb.acsss`, use the following commands:

```
tar tvbf 2048 /dev/rmt/0mn ---> SOLARIS
```

3. Restore database only (do not include ACSLS control files)

**Explanation:** The option provides to ability to restore data only. In some environments, you may need to restore the ACSLS database including its data, but you do not need to restore the ACSLS non-database control files.

Option 3:

Menu choice: 3

To recover the ACSLS database data only, either:

- Mount an ACSLS backup tape in a tape device and specify this tape device with `'-f tape_device'`, or

- Specify a file name containing an ACSLS backup with `'-f backup_file'`.

The ACSLS database data will be recovered from the file specified.

\*\*\*\*This option does not include the ACSLS control files\*\*\*\*

Please enter `-f [ backup_file | tape_device ]`:

4. Restore only ACSLS non-database control files

**Explanation:** Restores only the ACSLS control files. Before restoring any file located in the `$ACS_HOME/data/internal` directory, backups are made of the existing files, appending the end with a “.bak” extension.

```
$ACS_HOME/data/internal/dynamic_variables/dv_config.dat.bak
$ACS_HOME/data/internal/dynamic_variables/dv_trace.dat
$ACS_HOME/data/internal/release.vars.bak
```

This is not the case for files located in `$ACS_HOME/data/external`. No backups are performed of the ACSLS control files before recovery.

```
Option 4:
Menu choice: 4
To recover the ACSLS non-database control files either:
- Mount an ACSLS backup tape in a tape device and
  specify this tape device with '-f tape_device', or
- Specify a file name containing an ACSLS backup with '-f backup_file'.
ACSLs non-database control files will be recovered from the file specified.
Please enter -f [ backup_file | tape_device ]:
```

#### Example:

```
Please enter -f [ backup_file | tape_device ]: -f $ACSDB_BKUP/my_file.bak
```

- `-f $ACSDB_BKUP/my_file.bak` recovers the ACSLS control files from the specified file
- `-f /dev/rmt/0mn` recovers ACSLS control files from the specified tape device

#### 5. Exit

When you exit the `rdb.acsss` utility, a backup is initiated to the default directory, `$ACSDB_BKUP`.

See Recovery procedures for:

- [Exporting the Database](#)
- [Exporting the Database](#)
- [bdb.acsss](#)

## showDevs.sh

The `showDevs.sh` utility displays the critical device attributes associated with each mchanger instance in the `/dev` directory. Critical attributes include the library model number and revision level, cell capacity and number of attached drives. Additional attributes can be displayed using the options below.

### Format

```
showDevs.sh [-w] [-s]
```

### Options

This utility can be run with several options.

No argument.

This option displays each mchanger, library model and code level, and number of cells and drives.

- `-w`  
World Wide Name - along with the basic information, displays WWPN of the attached libraries.
- `-s`  
Serial number - along with the basic information, displays library serial number.

 **Note:**

To display server-side HBA information (including WWPN of HBA ports) and the WWPN of all attached libraries, use the `probeFibre.sh` utility as user `root`.

## showDrives.sh

This utility presents a list of all configured drives attached to ACSLS. The simple list of drive locations is sorted by drive type. If the verbose (`-v`) option is used, the utility displays a summary showing drive state, drive status, and assigned logical status of each drive.

### Format

```
showDrives.sh [-v]
```

## stats\_report

The `stats_report` utility generates library volume statistics reports. To run this utility, you must be logged in as `acsss`.

### Format

```
stats_report [vol_statsX.log ...]
```

Where:

```
vol_statsX.log -
```

1. Using this optional argument, you can specify one or more archived Volume Statistics Log file names.

(The archived files have the format `vol_statsX.log` (where  $0 \leq X \leq 8$ )).

Using one archived file as input:

```
$stats_report vol_stats0.log
```

The time centric and drive centric reports are generated with the name of the user input file pended to (and shown before) the report file name.

For example, if you specify `vol_stats0.log` then the reports will be generated in the `$ACS_HOME/log` directory as the following:

```
vol_stats0_drive_centric.txt and vol_stats0_time_centric.txt
```

2. To generate a report for all of the archived volume statistic files at once, follow the procedure below:
  - a. Generate the full log from individual files

```
$cd $ACS_HOME/log
$cat vol_stats8.log .... vol_stats0.log
acsss_stats.log > vol_statsXXXX.log
  where vol_statsXXXX.log
```

(The string `vol_stats` is necessary, but `XXXX` can be anything like `FULL`, and so on) is the concatenated file of all `vol_statsX.log` (where  $0 \leq X \leq 8$ ) and `acsss_stats.log` in reverse order.

- b. Run `stats_report`

```
$stats_report vol_statsXXXX.log
```

Reports are generated as `vol_statsXXXX_drive_centric.txt` and `vol_statsXXXX_time_centric.txt`.

If no filename is given as an argument, then the time centric and drive centric reports are generated from `$ACS_HOME/log/acsss_stats.log`.

## Usage

- The `stats_report` uses the current `acsss_stats.log` to prepare two reports of volume statistics. Enable library volume statistics gathering by setting the variable `LIB_VOL_STATS`. This can be done through the `acsss_config` (option 3) process or through the command line command `dv_config -p LIB_VOL_STATS.ACCLS` then automatically rolls and maintains 9 `acsss_stats.log` files when the log reaches the default size of 500 KB.
- The size of the log files and the number of files to retain is controlled through the variables `LIB_STATS_FILE_NUM` and `VOL_STATS_FILE_SIZE`. These variables are set using the same method as `LIB_VOL_STATS` discussed above.
- The two types of reports are:
  - `drive_centric.txt`

This report contains an ordered list of drives. Each drive record contains all cartridges mounted to the drive, the requestor, the time of the request and the duration of the mount.

  - `time_centric.txt`

### Note:

This report contains the usage of drive resources listed on an hourly time scale. Each record in a time period includes the requestor, the specific drive, the number of mounts during that period for that drive, and the duration of drive usage during the hour.

If the drive usage exceeds 60 minutes for a time period, it is an indication that the mount spanned two time periods and thus, will not be listed in the second time period. The first report created by `stats_report` is a drive view.

**Notes:**

- If there is a `DISMOUNT` record in the log but there is no corresponding `MOUNT` record, the reason could be that:
  - The log was rolled over, or
  - The operation was logged due to some unknown logging problem.  
In this case, the record is omitted from the report generated.
- If there is a `MOUNT` record in the log but there is no corresponding `DISMOUNT` record, the reason could be that:
  - The `DISMOUNT` has not yet happened, or
  - The operation was not logged due to some unknown logging problem.  
In this case, the mount duration is set to `-1`, which is an indication of cases mentioned above. These records are omitted from calculating the total mount duration in the time centric report.
- In the cases of going from day light savings time to standard time, scenarios where the calculated mount duration is negative can arise. To suppress them, the absolute value of the mount duration is taken.

## userAdmin.sh

The `userAdmin.sh` menu-driven utility administers ACSLS GUI user passwords. It is found in the `$ACS_HOME/install` directory. You can add users, remove users, list users, and change user passwords. WebLogic must be running to use this utility. If it is not up, this utility starts WebLogic and confirms that it is online before displaying the menu.

This utility is run by `root`, and requires `acsls_admin` authentication. The `acsls_admin` user account is configured during ACSLS 8.5 installation.

When adding a user or changing a password for any user, you are prompted for the user name and assigning password. The password is verified against WebLogic criteria of size and legal characters.

When a user is removed, that account could still have an active GUI session. Once the user logs out or terminates the session, the user will be unable to log back in. Restarting the GUI is the only way to force the session to terminate immediately. An option is provided to restart the ACSLS GUI (which terminates all sessions).

You cannot use this utility to change the password for the `acsls_admin` user. When it is necessary to change or reset the password for `acsls_admin`, you should:

1. Run the `winstall.sh` utility.  

```
$installDir/winstall/winstall.sh
```
2. Run `userAdmin.sh` to re-establish the remaining user accounts.

## Format

```
userAdmin.sh
```

## Examples

```
# ./userAdmin.sh
  ACSLS GUI User Administration
  Weblogic is online.
Please enter the acsls_admin password:
Authenticating.....Connected!
Menu:
1) Add a user account.
2) Remove a user account.
3) Change a user password.
4) List users.
5) Restart ACSLS GUI.
6) Exit.
Please select by number: 1
--- Add a User ---
Please enter the id of the user you wish to add: acsss
Do you wish to add a GUI account for user 'acsss'? (y/n) y
Please assign a password for 'acsss'.
  Passwd: Please confirm password:
  Passwd:
Connecting.....
User accounts has been added.
Please select by number: 2
--- Remove a User ---
Please enter the name of the user you wish to remove: accounts Do you wish to remove
the ACSLS GUI account for user 'accounts'? (y/n) y
Connecting.....
The account for accounts has been removed for future logins.>
To disable any current login session for accounts, you must restart the ACSLS GUI.
Please select by number: 3
--- Change Password ---
Enter the user name: acsss
Passwd: Please confirm password:
Passwd:
Connecting.....
Password changed for acsss!
Please select by number: 4
--- List Users ---
Connecting.....

Configured WebLogic users:
  OracleSystemUser
  acsls_admin
  acsss
Please select by number: 5
Do you wish to restart the ACSLS GUI (affects all users)? (y/n) y Restarting:
  Disabling WebLogic .....
  Enabling WebLogic .....
Please select by number: e
#
```

## volrpt

The volrpt utility creates a volume report.

## Format

```
volrpt [-s vol|loc|use] [-d] [-f filename] [-z] [-a|-l|-v identifier_list] [-i]
```

## Options

- `-s`  
specifies the sort order. If you do not specify this option, the default is to sort by volume ID. If you specify this option, you must specify one of the following values:
  - `vol`  
sort by volume ID.
  - `loc`  
sort by volume home location.
  - `use`  
sort by volume use (number of mounts).
- `-d`  
specifies that the output contains no page breaks or header information. The output can be used as input to other programs such as `pr`.
- `-f filename`  
*filename* specifies a custom volrpt template.
- `-z`  
zero fills identifier fields.
- `-a`  
restricts the report to the specified ACS. You can specify multiple ACSs (use blanks to separate the *acs\_ids*).
- `-l`  
restricts the report to the specified LSM. You can specify multiple LSMs (use blanks to separate the *lsm\_ids*).
- `-v`  
restricts the report to the specified volumes (or volume ranges). You can specify a single *vol\_id*, a list of *vol\_ids* separated by blanks, or a volume range indicated by *vol\_id-vol\_id*.
- `identifier_list`  
described by the `-v`, `-a`, and `-l` options. This is a list of ACSs, LSMs, and volumes (or volume ranges).
- `-i`  
reports all volumes, including absent and ejected cartridges.  
If this option is not specified, absent and ejected cartridges are not reported.

## Usage

Use the `volrpt` utility to create a report of library cartridges, including their physical location, history, attributes, and use. You can also use `volrpt` to verify the database after you restore it. You can use the `-a`, `-l`, or `-v` options to specify the ACSs, LSMs, or cartridges for the report. If you do not specify any of these options, `volrpt` reports on only ACS 0.

### Note:

Special consideration for leading and trailing spaces. When specifying arguments for volumes that contain leading or trailing spaces, you should enclose the arguments in single quotes. To assure that the single quote will be passed from one shell component to another, the quote must be tagged with an escape character. In UNIX, the standard escape character is the backslash (`\`).

### Examples:

To formulate a `volrpt` command on the local machine where you want to reference volume id's with a leading space, you would submit the command, as follows:

```
volrpt -v \'0000\'-\'9999\'
```

To submit the same command through a remote shell (`rsh`) you would enclose the entire argument inside double quotes:

```
rsh <acsls_hostname> -l acsss bin/volrpt -v \"\' 0000\'-\' 9999\'\"
```

The following example shows a standard volume report, which contains fields for volume id, location, label type, media type, and history of usage.

```
VOLUME REPORT UTILITY
2002-06-30 14:01:21
TOTAL VOLUMES: 400 SEQUENCE: sort by volume identifier
Volume Home      LabelVolume      Times|---Entered---||--Last Used--| Label Location
AttrType/Media MountedDateTime      DateTime
CLN000 0,0,1,0,3 ExtC/STK1U    108/22/0109:30    10/04/01 14:26
RB0000 0,1,2,1,10Ext.D/STK1R    310/01/0108:16    10/01/01 08:18
RB1400 0,0,10,1,3Ext.S/STK1R    24310/01/0109:30    10/06/01 11:04
RB1401 0,0,10,3,5Virt.D/STK1R    1210/01/0103:29    10/05/01 23:11
"      "      "      "      "      ""
"      "      "      "      "      ""
"      "      "      "      "      ""
TB1440 0,1,3,1,9 Ext.D/STK2P    4308/12/0109:1109/28/0117:52
"      "      "      "      "      ""
"      "      "      "      "      ""
"      "      "      "      "      ""
```

In the `Volume Type/Media` column: C denotes cleaning cartridges; D denotes data cartridges; P denotes a cleaning cartridge that was reported as spent (used-up) by a tape drive; and S denotes scratch cartridges.

Use the `-f filename` option to create a customized report; see "[Creating a Logging Volume Statistics Report](#)", for more information.



`$ACS_HOME/data/external/volrpt/owner_id.volrpt` is a sample input file that you can run or use as a template to create customized volume reports. You can also save your customized volume reports in the `$ACS_HOME/data/external/volrpt` directory.

You can redirect the volume report to a file with standard UNIX redirection:

```
volrpt > file
```

## Examples

By default, `volrpt` reports only the first ACS in the list. To report the cartridges in both ACS 0 and ACS 1, enter the following command:

```
volrpt -a 0 1
```

To report the cartridges in LSMs 0,1 and 2,1 sorted by home cell location, enter the following command:

```
volrpt -s loc -l 0,1 2,1
```

### Notes:

- `volrpt` displays the specified volume report if it completes successfully. `volrpt` prints a message to `stderr` and exits if you specify the `-f` option and `volrpt` cannot find the specified file or you specify more than one input file. For field errors within the input file, `volrpt` prints a message to `stderr` and ignores the line in error but does not exit.
- If cartridges are not found in the volume ID list, range or library component specified, `volrpt` returns a `no volumes found` message.
- When a parameter is not specified, it uses the default of ACS 0.
- If a library component(s) is specified through the `-a`, `-l`, or `-v` option, but no volumes are found, messages such as the following are displayed:

- `-a` option (ACS)

Messages:

when a single `acs_id` is provided and no volumes are present, the following error displays: `No Volumes found for ACS: (<acsid>)`

**Example:**

```
$ volrpt -a 2
No Volumes found for ACS: (2)
```

when multiple `acs_ids` are provided and none of them have any volumes, the following error displays:

```
No Volumes found for ACS: (<acsid1>)(acsid2>)
```

**Example:**

```
$ volrpt -a 0 1
No Volumes found for LSM: (0) (1)
```

- `-l` option (LSM)

Messages:

when a single `lsm_id` is provided and no volumes are present, the following error displays:

```
No Volumes found for LSM: (<lsmid>)
```

**Example:**

```
$ volrpt -l 1,1  
No Volumes found for LSM: (1,1)
```

when multiple `lsm_ids` are provided and none of them have any volumes, the following error displays:

```
No Volumes found for LSM: (<lsmid1>)<lsmid2>  
> (<lsmid2>)
```

**Example:**

```
$ volrpt -l 1,1 1,2  
No Volumes found for LSM: (1,1) (1,2)
```

– **-v option (VOLUME)**

Messages:

when a single `valid` is provided and no volumes are present, the following error displays:

```
Volume(s) not: (<valid>)
```

**Example:**

```
$ volrpt -v BBB112  
No Volumes found: (BBB112)
```

when multiple `valids` are provided and none of them have any volumes, the following error displays:

```
Volume(s) not found: (<valid1>)<valid2>
```

**Example:**

```
$ volrpt -v BBB112 BBB114  
No Volumes found: (BBB112) (BBB114)
```

The `-v` option can also be used for volume range, and produces similar messages when no volumes are present.

when a single volume range is provided and no volumes are present, the following error displays:

```
Volume(s) not: (<volrange>)
```

**Example:**

```
$ volrpt -v BBB112-BBB116  
No Volumes found: (BBB112-BBB116)
```

when multiple volume range is provided and no volumes are present, the following error displays:

```
Volume(s) not: (<volrange1>)<volrange2>
```

**Example:**

```
$ volrpt -v BBB112-BBB116 BBB220-BBB224
No Volumes found: (BBB112-BBB116) (BBB220-BBB224)
```

When an ACS or LSM has not been configured

When `volrpt` is used with an `acs_id` or `lsm_id` that does not exist, it displays a message according to the identifier.

- `-a` (ACS)  
ACS identifier (<*acsid*>) not configured
- `-l` (LSM)  
LSM identifier (<*lsmid*>) not configured

See [Creating a Logging Volume Statistics Report](#) .

## watch\_vols

This utility applies pre-defined policies for volumes that are:

- newly entered
- discovered by audit or cartridge recovery
- re-activated by audit, cartridge recovery, or an `enter`

These policies are defined in the file:

```
$ACS_HOME/data/external/vol_attr.dat
```

This file contains a list of user-defined volume IDs or volume ranges and a user-specified policy for each volume recorded. For each volume or volume range listed in that file, you can define volume ownership, pool association, preferred LSM location, and/or logical library assignment when a volume is entered. Specific instructions for defining policies are explained in detail in the `vol_attr.dat` file.

The `watch_vols` utility uses the `acsss_stats.log` to identify the existence of newly entered volumes or volumes discovered or re-activated during an audit or by cartridge recovery. To enable this capability, you must enable volume statistics with `acsss_config` (option 3). With volume statistics enabled, `watch_vols` monitors the tail of the `acsss_stats.log`, looking for matching volumes with the entries defined in `vol_attr.dat`. Wherever a match is found, the defined policy for that volume is automatically applied.

The volume IDs must follow these rules:

- One `vol_id` or volume range per line.
- The `vol_ids` must be valid ACSLS volume IDs.
- If the `vol_ids` include trailing or leading spaces, they must be represented as underscores (`_`). For example: `_V234_`.

## Format

```
watch_vols [start|stop]
```

## Usage

You can check the running status of the utility by invoking `watch_vols` with no parameter. If you are unsure of the status of `watch_vols` (running or stopped), the command `watch_vols` with no argument displays the current status.

There are two options for `watch_vols`, `start` and `stop`.

- `watch_vols start`

When the `start` parameter is invoked, `watch_vols` reviews the policies defined in `vol_attr.dat`. If there are errors in format or syntax, `watch_vols` displays the error and prompts you to make the necessary correction to `vol_attr.dat`. Once the defined policy is accepted by `watch_vols`, the utility invokes a daemon to run in the background. The daemon continues to run if ACSLS is running. It starts automatically whenever ACSLS is restarted.

The policy table in `vol_attr.dat` can be updated at any time. You need not stop `watch_vols` in order to update the policy. Just run `watch_vols start` to commit the updates to the running program.

- `watch_vols stop`

This command stops any further policy enforcement for the specified volumes.

A log of all `watch_vols` activities is maintained in the log file:

```
$ACS_HOME/log/watch_vols_event.log
```

Each change of volume ownership, `pool_id`, or LSM home location is logged in this file.

## Example

You are performing an enter operation and you want to move specific volumes to a target LSM when they are entered.

1. Audit the target LSM with `watch_vols` disabled.
2. Once the target LSM has been audited, start `watch_vols`.
3. Enter the volumes that have policies defined in `vol_attr.dat`.  
`watch_vols` then moves the specified volumes to the destination LSM after they are entered.

# 12

## Command Reference

In this chapter, you will learn how to use the ACSLS commands, including general command syntax and reference information.

- [audit](#)  
Creates or updates the database inventory of the volumes in a library component.
- [cancel](#)  
Cancels a current or pending request.
- [clear lock](#)  
Removes all active and pending locks on drives or cartridges.
- [define pool](#)  
Creates or modifies scratch pools.
- [delete pool](#)  
Deletes empty scratch pools.
- [dismount](#)  
Dismounts a cartridge.
- [eject](#)  
Ejects one or more cartridges from the ACS.
- [enter](#)  
Sets a CAP to enter mode.
- [idle](#)  
Stops ACSLS from processing new requests.
- [lock](#)  
Locks (dedicates) a cartridge or drive to a user.
- [logoff](#)  
Exits from cmd\_proc.
- [move](#)  
moves a specified cartridge to an available storage cell in a specified LSM.
- [mount](#)  
Mounts a data or scratch cartridge.
- [query commands](#)  
Displays the status of a library component.
- [set commands](#)  
Sets various attributes of different library components.
- [show](#)  
Displays your lock ID or user ID.
- [start](#)  
Starts ACSLS request processing.
- [switch lmu](#)  
Manually switches ACS management from the ACS's active LMU to the standby LMU.

- **unlock**  
Removes active locks on cartridges or drives.
- **uptime**  
Indicates the amount of elapsed time ACSLS has been enabled and/or the last restart of the ACSLS library manager.
- **vary**  
Changes the state of an ACS, LSM, CAP, drive, or port.

## General Command Syntax

This section describes the general syntax of the ACSLS commands. The following sections fully describe each command, including its syntax.

ACSLs commands use the following general syntax:

```
command type identifier state options
```

Where:

```
type identifier
```

is the ACS component and its identifier; see [Component Types and Identifiers](#) for more information.

- *state*  
is a device state for the `vary` command only.
- *options*  
are command options; see the description of the command you want to run.  
Note the following syntax rules:
  - Enter commands in the order shown above (command name, followed by the component and its identifier, the state and any options).
  - This chapter shows commands in lowercase, but you can enter commands in any combination of lowercase and uppercase letters.
  - Underlines show minimum abbreviations for commands and keywords. For example, **query server** and `q ser` are both valid forms of the `query server` command.
  - Ellipses (...) indicate that you can repeat an identifier.
  - Brackets [ ] enclose optional options.

## Component Types and Identifiers

The following table describes the valid ranges of values for each of the ACS component identifiers. You can specify a maximum of 42 identifiers for each type. See specific command descriptions for the valid component types of each command.

 **Note:**

The identifiers specified in the following table represent valid ranges of values supported by the software. Your LSM type and library configuration determine what identifier values are valid for your particular site.

**Table 12-1 ACSLs Component Types and Identifiers**

| Component        | Type            | Identifier         | Valid Values                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------|-----------------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| entire library   | <u>server</u>   | none               | none                                                                                                                                                                                                                                                                                                                                                                                                    |
| ACS              | <u>acs</u>      | <i>acs_id</i>      | acs(0-31),<br>lsm(0-99)                                                                                                                                                                                                                                                                                                                                                                                 |
| LSM              | <u>lsm</u>      | <i>lsm_id</i>      | acs(0-31),<br>lsm(0-99)                                                                                                                                                                                                                                                                                                                                                                                 |
| LSM panel        | <u>panel</u>    | <i>panel_id</i>    | acs(0-31),<br>lsm(0-99),<br>panel(0-50),                                                                                                                                                                                                                                                                                                                                                                |
| LSM subpanel     | <u>subpanel</u> | <i>subpanel_id</i> | acs(0-31),<br>lsm(0-99),<br>panel(0-50),<br>startrow(0-51),<br>startcolumn(0-23),<br>endrow(0-51),<br>endcolumn(0-23)<br><br>The ending row (and column) must be greater than or equal to the beginning row (and column). Only cells within the beginning and ending matrix are audited: the matrix starts with the beginning row and beginning column and extends to the ending row and ending column. |
| LSM storage cell | <u>subpanel</u> | <i>cell_id</i>     | acs(0-31),<br>lsm(0-99),<br>panel(0-50),<br>row(0-51),<br>column(0-23)                                                                                                                                                                                                                                                                                                                                  |

**Table 12-1 (Cont.) ACSLS Component Types and Identifiers**

| Component                                                                   | Type                                   | Identifier        | Valid Values                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------------------------------------------------------------------------|----------------------------------------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| CAP                                                                         | <u>cap</u>                             | <i>cap_id</i>     | acs(0-31),<br>lsm(0-99),<br>cap(0-11)<br><br>An asterisk (*) in the <i>cap_id</i> does the following:<br>acs,lsm,* - causes ACSLS to select the highest priority available CAP in the LSM.<br>acs,* - causes ACSLS to select the highest priority available CAP in the ACS.<br>* - for an enter request causes ACSLS to select the CAP in the LSM with the most free cells.<br>* - for an eject request causes ACSLS to select the highest priority CAP in each ACS with a cartridge designated for ejection.                                                                                                                                                                                                                                                                                                                         |
| drive                                                                       | <u>drive</u>                           | <i>drive_id</i>   | acs(0-31),<br>lsm(0-99),<br>panel(0-50),<br>drive(0-31)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| drive type                                                                  | <u>drive</u>                           | <i>drive_type</i> | Up to a 10-character drive type identifier; can be any combination of numbers (0-9) or letters (A-Z). Spaces are not allowed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| port                                                                        | <u>port</u>                            | <i>port_id</i>    | acs(0-31),<br>port(0-3)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| volume serial number of a data or scratch cartridge or a cleaning cartridge | <u>volume,scrat</u><br><u>ch,clean</u> | <i>vol_id</i>     | Six-character identifier consisting of any combination of numbers (0-9), letters (A-Z), dollar sign (\$), pound sign (#), and spaces<br><br>Use single or double quotes to enclose volsers with leading or trailing spaces.<br><br><i>Do not</i> specify volsers with embedded spaces.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| range of volume serial numbers                                              | <u>volume</u>                          | <i>volrange</i>   | Specifies an ascending range of volumes separated by a dash (-)<br><br>If the ALPHANUM_VOL_RANGES dynamic variable is set to FALSE (default): ACSLS supports the starting and ending <i>vol_ids</i> of the range that are specified as complete <i>vol_ids</i> (Ex: AAA000-AAA999). Specify only the right-most numeric portions of the volsers as the range. All preceding characters must be identical. ***<br><br>If the ALPHANUM_VOL_RANGES dynamic variable is set to TRUE: ACSLS supports volume ranges consisting of any combination of numbers (0-9), letters (A-Z), dollar sign (\$), pound sign (#), and spaces (leading and trailing). ***<br><br>*** Use single or double quotes to enclose volsers in a range with leading or trailing spaces. The range must be in ascending sequence per the ACSII collating sequence. |



**Table 12-1 (Cont.) ACSLS Component Types and Identifiers**

| Component            | Type           | Identifier        | Valid Values                                                                                                                                |
|----------------------|----------------|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| volume media type    | <u>media</u>   | <i>media_type</i> | Up to a 10-character media type identifier; can be any combination of numbers (0-9), letters (A-Z), and dashes (-). Spaces are not allowed. |
| volume owner         | <u>owner</u>   | <i>owner_id</i>   | -                                                                                                                                           |
| scratch pool         | <u>pool</u>    | <i>pool_id</i>    | Decimal number (0-65534). Specifying an asterisk (*) for the <i>pool_id</i> reassigns the volume to its current <i>pool_id</i> .            |
| ACSLS request        | <u>request</u> | <i>request_id</i> | Unique decimal number (0-65535) assigned by ACSLS.                                                                                          |
| drive or volume lock | <u>lock</u>    | <i>lock_id</i>    | Decimal number (0-32767)                                                                                                                    |
| ACSLS request        | <u>request</u> | <i>request_id</i> | unique numeric (0-65535) request identifier assigned by ACSLS.                                                                              |

## Common Command Error Messages

A common error message appears if ACSLS rejects a command because of a syntax error, invalid identifier, type, option, process failure, database error, and so forth. For more information on common error messages, refer to the publication *ACSL Messages*.

If you enter a command that you cannot use, the following message appears:

```
Command access denied.
```

If you specify a volume that you cannot access, the following message appears:

```
Volume access denied.
```

## Commands

The following section describes the ACSLS commands.

### audit

The `audit` command updates the ACSLS database to match the actual inventory of library cartridges.

You must always perform an audit, if:

- This is a new library.
- To re-synchronize the ACSLS database with the contents of the library.
- One or more modules have been added, removed or swapped.
- Cartridges have been manually added or removed through the door.
- After library configuration settings have changed.
- If ACSLS does not know the location of one or more cartridges in the library.

- If a library has been expanded and you are auditing it for the first time. When a SL8500 has been expanded, see the procedures described in [Understanding SL8500 Internal Addresses and ACSLS Addresses](#), to audit it for the first time.

## Format

```
audit cap_id type identifier...
```

- *cap\_id*  
specifies the CAP that ACSLS uses to eject any errant volumes. You can specify a particular cap or enter an asterisk (\*), which causes ACSLS to select the highest priority CAP.  
  
You can specify the server or multiple ACSs only if CAP priority has been set in each ACS. You must specify an asterisk for the *cap\_id* to allow automatic selection of a CAP in each ACS.  
  
A single LSM *audit* waits until ACSLS updates the database, then reserves the CAP if required to eject errant volumes.
- *type identifier*  
specifies a library component. The following table lists the components you can audit.

**Table 12-2 Valid Components for Audit**

| Library Component | <i>type</i> | <i>identifier</i>  |
|-------------------|-------------|--------------------|
| all libraries     | server      | <i>none</i>        |
| ACS               | acs         | <i>acs_id</i>      |
| LSM               | lsm         | <i>lsm_id</i>      |
| LSM panel         | panel       | <i>panel_id</i>    |
| LSM subpanel      | subpanel    | <i>subpanel_id</i> |

You can specify multiple ACSs, LSMs, panels, or subpanels in a single *audit* request. You cannot specify overlapping subpanels.

You can specify the server or multiple ACSs only if CAP priority has been set in each ACS. You must specify an asterisk for the *cap\_id* to allow automatic selection of a CAP in each ACS. If this is not done, *audit* cannot eject any invalid cartridges found.

When you audit the server or an ACS, the parent audit process creates a separate audit process for each LSM. These LSM audits run in parallel and process panel-by-panel in ascending *panel\_id* sequence. To audit all LSMs in all libraries or an ACS, just audit the server or ACS. There is no need to run multiple, separate LSM audits.

However, within an LSM, regardless of the order in which you specify multiple components the audit processes these components in ascending order by *acs\_id*, *lsm\_id*, *panel\_id*, *subpanel\_id*. For example, the audit will process panel 0,0,9 before subpanel 0,0,10,1,7 even if you specify the subpanel first.

## Usage

An `audit` updates the ACSLS database to match the actual inventory of library cartridges. You run an `audit` to:

- Create volume information in the database if you do not enter cartridges through the CAP. For example, run an `audit` if you add an LSM to your library, open the LSM door and manually add cartridges to the LSM.
- Resolve discrepancies between the library and the database. For example, run an `audit` if you open an LSM door and manually remove cartridges instead of ejecting them through the CAP. The `audit` deletes information for the removed cartridges from the database.
- View the contents of cells specified by the `audit` (you must have a display monitor connected to the LSM robots' vision system).

 **Note:**

`Audit` recognizes particular models of cleaning cartridges and will record them in the database as cleaning cartridges. The `audit` command also sets the `max_uses` for new cleaning cartridges it encounters appropriately for each type of cleaning cartridge.

 **Note:**

When `audit` cannot find a cartridge in any recorded location (cell and possibly drive), `audit` either marks the cartridge as absent (if absent volume retention is enabled) or deletes the volume. When `audit` deletes the volume, it removes all information about the volume from the database, including volume information and customer-supplied information about access control, scratch status, lock IDs, and pool IDs. If `audit` later finds the cartridge in another location, it re-adds the volume information, but the customer-supplied information is lost.

An `audit` ejects errant volumes and deletes their information from the database. An errant volume has:

- An external label that duplicates one already scanned.
- A missing or unreadable external label and no virtual label.
- An invalid media type.

Audits only apply to LSM storage cells, not to drives or CAPs. ACSLS displays `cmd_proc` messages during the audit and records any database changes from the `audit` in the Event Log. If volume statistics logging is enabled, ACSLS records additional volume found and volume not found messages in the `acsss_stats.log`. See "[Creating a Logging Volume Statistics Report](#)" for more information.

**HINT:** Use the following guidelines for running an `audit`:

- The ACS or LSM being audited must be either online or in diagnostic state. Normal library processing (including mounts and dismounts) can occur during an `audit`, although library processing slows down the audit.

The difference between online and diagnostic states, is: an `audit` in diagnostic state can only be performed through the `cmd_proc`. Client requests are rejected while the LSM is in the diagnostic state. This state is faster because the `audit` now has exclusive run of the library without competing `mount/dismount` requests that could otherwise be submitted from the client.

- After you start an `audit` on an entire LSM, you cannot start another `audit` on the same LSM. You must `cancel` and rerun the audit.
- You can `cancel` any audit request, but ACSLS will always finish auditing the current panel or subpanel. When you `cancel` an audit, some or all of the cartridges marked for ejection may not be ejected. When you `cancel` an audit, cartridges already ejected are not re-entered.

### **Caution:**

If you `cancel` an audit, or if there is a library or ACSLS hardware or software failure during the audit, you must rerun the same audit. Cartridges marked for ejection but not actually ejected during the first audit are no longer in the database and are not under ACSLS control.

## Examples

- To audit the entire library and specify the highest priority CAP in each ACS for ejections:

```
audit * server
```

- To audit LSM 0,1 and specify CAP 0,1,1 for ejections:

```
audit 0,1,1 lsm 0,x
```

- To audit panel 10 of LSM 0,1 and specify the highest priority CAP in LSM 0,1 for ejections:

```
audit 0,1,* panel 0,1,10
```

### **Note:**

A cell cannot be audited if it is reserved by another process. If a cell is reserved, ACSLS rechecks the database until the cell becomes available up to a maximum of 60 retries. If the cell is still unavailable, the `audit` skips the cell and logs a message to the Event Log.

See also:

| For information about...            | See...                                 |
|-------------------------------------|----------------------------------------|
| Guidelines for auditing the library | <a href="#">Specifying ACS Numbers</a> |
| Cancelling a command                | <a href="#">cancel</a>                 |

| For information about...                  | See...                           |
|-------------------------------------------|----------------------------------|
| Displaying CAP status                     | <a href="#">query cap</a>        |
| Displaying ACSLS and library status       | <a href="#">query server</a>     |
| Displaying ACS status                     | <a href="#">query acs</a>        |
| Displaying LSM status                     | <a href="#">query lsm</a>        |
| Setting CAP selection priority            | <a href="#">set cap priority</a> |
| Setting CAP mode (manual or automatic)    | <a href="#">set cap mode</a>     |
| Changing the state of a library component | <a href="#">vary</a>             |

## Command Area Messages

The following section discusses audit messages.

### Success Messages:

- The following message appears when the `audit` completes successfully.  
Audit: Audit completed, Success.
- In addition, one of the following messages appears to confirm which component was audited:

```
Audit: Audit of storage server, valid
```

```
Audit: Audit of ACS, acs_id, status valid
```

```
Audit: Audit of LSM, lsm_id, panel_id, valid
```

```
Audit: Audit of panel, panel_id, valid
```

```
Audit: Audit of subpanel, subpanel_id, valid
```

**HINT:** If you `audit` an invalid cell location (that is just above or below a drive or on an extra column), ACSLS returns a success message to allow audits to work with PTPs.

### Intermediate Messages:

Intermediate messages consist of a two-line display with the following first line:

```
Audit: Intermediate response: Audit activity.
```

One of the following messages appears on the second line:

- Audit: Volume ejected, unreadable label.  
Explanation: ACSLS ejected a cartridge that had:  
No external label  
No virtual label, or  
An unreadable label
- Audit: Volume *vol\_id* will be ejected, duplicate label.  
Explanation: ACSLS ejected with a duplicate external label within the range of cells being audited.  
Variable: *vol\_id* is the volume with the duplicate label.

- Audit: Volume *vol\_id* found.  
Explanation: The audit found a volume in the ACS that is not in the ACSLS database. The audit added the volume to the database.  
Variable: *vol\_id* is the volume added to the database.
- Audit: Volume *vol\_id* not found.  
Explanation: A volume listed in the ACSLS database is not in the ACS. The volume is deleted from the database.  
Variable: *vol\_id* is the volume deleted from the database.
- Audit: Volume will be ejected, invalid media type  
Explanation: ACSLS ejected a volume with an invalid media type.

**Error Messages:**

- Audit in progress.  
Explanation: ACSLS did not start the `audit` because another audit for the same LSM is in progress.
- CAP *cap\_id* in use.  
Explanation: The CAP specified for the audit is in use.  
Variable: *cap\_id* is the CAP in use.
- Multiple ACS audit.  
Explanation: The audit failed because the `audit` command specified multiple ACSs without specifying an `*` for the *cap\_id*.
- Not in same ACS.  
Explanation: The `audit` failed because the *cap\_id* and the *identifier* specified are not in the same ACS.  
Variable:  
*cap\_id* is the CAP specified for the audit.  
*identifier* is the library component specified for the audit.

**Display Area Messages:**

- *cap\_id* Remove cartridges from CAP.  
Explanation: The audit filled the CAP with ejected cartridges. Empty the CAP, then close it to continue the audit.  
Variable: *cap\_id* is the CAP that contains the ejected cartridges.
- CAP *cap\_id*: Place magazines in CAP.  
Explanation: The CAP requires magazines for the audit. Open the CAP, place magazines inside, then close the CAP.  
Variable: *cap\_id* is the CAP that requires magazines.
- CAP *cap\_id*: No CAP available, waiting...  
Explanation: No CAP is available to eject cartridges.  
Variable: The *cap\_id* appears as it was specified in the audit command:

- `acs, lsm, cap` if the `audit` command explicitly specified the CAP.
- `acs, lsm, *` if the `audit` command specified the CAP as `acs, lsm, *`.
- `acs, *, *` if the `audit` command specified the CAP as `acs, *` or `*`.

## cancel

The `cancel` command cancels a current or pending request.

## Format

```
cancel request_id
```

## Options

- `request_id`

specifies the identifier of the request to cancel.

Use the `cancel` command to cancel current or pending request issued by an `audit`, `define pool`, `delete pool`, `eject`, `enter`, `lock`, `query`, `set`, or `venter` command or client application. Use the `query request` command to display the ID of the request you want to cancel.

You can `cancel` an audit of a server, ACS, or LSM. Because ACSLS internally translates server, ACS, or LSM audits into a series of panel audits, ACSLS completes the `audit` of the current panel before cancelling the remainder of the audit. You cannot `cancel` an audit of a panel or subpanel. When you `cancel` an audit, cartridges already ejected are not re-entered.

### Caution:

If you `cancel` an audit or if there is a library or ACSLS hardware or software failure during the audit, you must rerun the same audit. Cartridges marked for ejection but not actually ejected during the first audit are no longer in the database and are not under ACSLS control.

**HINT:** Enter a `cancel` command from a different `cmd_proc` than the `cmd_proc` that issued the request you want to cancel.

The `cancel` command immediately cancels any pending requests and handles current requests as follows:

- `audit`

Because ACSLS internally translates server, ACS, or LSM audits into a series of panel audits, ACSLS completes the `audit` of the current panel before cancelling the remainder of the audit.

 **Note:**

If you `cancel` an audit or if there is a library or ACSLS hardware or software failure during the audit, you must re-run the same audit. Cartridges marked for ejection but not actually ejected during the first audit are no longer in the database and are not under ACSLS control.

- `define pool`  
ACSL S stops defining scratch pools but does not delete any scratch pools already defined.
- `delete pool`  
ACSL S stops deleting scratch pools but does not redefine any scratch pools already deleted.
- `eject`  
ACSL S stops the ejection and the `cmd_proc` displays a message to remove any cartridges already ejected, which are not re-entered.  
The `eject` does not terminate until the cartridges are removed, the CAP is closed, and ACSLS verifies that the CAP is empty.
- `enter`  
ACSL S stops the `enter` and, if any cartridges remain in the CAP, `cmd_proc` displays a message to remove these cartridges. Any cartridges already entered into the LSM are not ejected.  
The `enter` does not terminate until the cartridges are removed, the CAP is closed, and ACSLS verifies that the CAP is empty.

 **Note:**

Automatic `enter`. You cannot `cancel` an automatic `enter` operation that is in progress using the `cancel` command. To terminate an automatic `enter` in progress:

- If the CAP door is open, remove all of the cartridges and close the door.
- If the CAP door is closed and the cartridges are being moved into the library, allow the remaining cartridges to be entered into the library. The `enter` then terminates.
- `lock`  
Resource locking by the specified request is stopped. If the request has not yet acquired all specified resources, none of the resources are locked.
- `query`  
ACSL S cancels the query.
- `set`



For `set cap`, ACSLS stops setting CAP attributes, but does not change any attributes already set.

For `set scratch` or `set clean` requests, ACSLS stops setting scratch cartridge or cleaning cartridge attributes, but does not change any attributes already set.

- `venter`

ACSLs stops the enter, and, if any cartridges remain in the CAP, `cmd_proc` displays a message to remove these cartridges. Any cartridges already entered into the LSM are not ejected.

## Examples

- To display request IDs for all current and pending requests:

```
query request all
```

Example output of `query request all`:

| Identifier | Command | Status  |
|------------|---------|---------|
| 13         | enter   | Current |
| 15         | query   | Pending |

- To cancel request 13 (current enter request) in the example above:

```
cancel 13
```

See also:

| For information about...                                                        | See...                         |
|---------------------------------------------------------------------------------|--------------------------------|
| Updating the ACSLS database to match the actual inventory of library cartridges | <a href="#">audit</a>          |
| Creating or modifying scratch pools                                             | <a href="#">define pool</a>    |
| Deleting empty scratch pools                                                    | <a href="#">delete pool</a>    |
| Ejecting cartridges from the library                                            | <a href="#">eject</a>          |
| Making a CAP (manual mode) ready to enter labelled cartridges into the library  | <a href="#">enter</a>          |
| Locking (dedicating) drives and cartridges to your current lock ID              | <a href="#">lock</a>           |
| Displaying the status of a library component                                    | <a href="#">query commands</a> |
| Setting various attributes of different library components                      | <a href="#">set commands</a>   |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- `Request request_id canceled.`

Explanation: ACSLS canceled the requested command.

Variable: `request_id` is the request identifier of the canceled command.

### Intermediate Messages:

None.

#### Error Messages:

- Request `request_id` can not be canceled: `status`.  
Explanation: ACSLS cannot cancel the specified command.  
Variable:
  - `request_id` is the request identifier of the command that ACSLS cannot cancel.
  - `status` is one of the following:
- Request identifier `request_id` invalid.  
The `cancel` command specified an invalid request identifier.
- Request identifier `request_id` not found.  
The `cancel` command specified an request identifier for a request that is not current or pending.

## Display Area Messages

None.

## clear lock

The `clear lock` command removes all active and pending locks on a specified drive or cartridge.

## Format

```
clear lock type identifier
```

## Options

```
type identifier
```

specifies a library component. The following table lists the components whose resource locks you can clear.

**Table 12-3 Valid Components for Clear Lock**

| Library Component | Type   | Identifier |
|-------------------|--------|------------|
| drive             | drive  | drive_id   |
| volume            | volume | vol_id     |

## Usage

Use the `clear lock` command to remove all active and pending locks on a specified drive or cartridge. Your current lock ID must either be 0 or it must match the lock ID of the driver or cartridge.

The `unlock` command removes only active locks on drives or cartridges. You can, however, use the `unlock` command to remove active locks on all drives or all cartridges.

**Note:**

The `clear lock` command always resets your lock ID to 0.

## Examples

- To clear all locks for drive 1,1,5,2:  
`clear lock drive 1,1,5,2`
- To clear all locks for volume NN0108:  
`clear lock volume NN0108`

**Note:**

You cannot cancel a `clear lock` command.

See also:

| For information about...                           | See...                     |
|----------------------------------------------------|----------------------------|
| Locking drives and cartridges                      | <a href="#">lock</a>       |
| Displaying the lock status of a drive or cartridge | <a href="#">query lock</a> |
| Setting your lock ID                               | <a href="#">set lock</a>   |
| Displaying your lock or user ID                    | <a href="#">show</a>       |
| Removing active locks for drives or cartridges     | <a href="#">unlock</a>     |

## Command Area Messages

This section discusses the command area messages.

**Success Messages:**

- The following message appears when a clear request succeeds:  
`Clear: Completed, Success.`
- In addition, for each identifier in the request, one of the following messages appears depending on the library component (*type*):
  - `Clear: Drive drive_id all locks cleared.`
  - `Clear: Volume vol_id all locks cleared.`

**Intermediate Messages:**

None.

**Error Messages:**

- Clear: Clear lock of drive `drive_id` failed,  
Drive identifier `drive_id` available.  
**Explanation:** ACSLS cannot clear locks because the specified drive is not locked.  
**Variable:** `drive_id` is the identifier of the specified drive.
- Clear: Clear lock of volume `vol_id` failed,  
Volume identifier `vol_id` available.  
**Explanation:** ACSLS cannot clear locks because the specified volume is not locked.  
**Variable:** `vol_id` is the identifier of the specified volume.

## Display Area Messages

None.

## define pool

The `define pool` command creates or modifies scratch pools.

## Format

```
define pool low_water_mark high_water_mark pool_id...[overflow]
```

## Options

- `low_water_mark`  
is the low volume warning threshold. If the scratch cartridge count falls below this threshold, ACSLS logs a warning message in the Event Log. Valid values are 0 to  $2^{31}-1$ . The default is 0.
- `high_water_mark`  
is the high volume warning threshold. If the scratch cartridge count reaches or exceeds this threshold, ACSLS logs a warning message in the Event Log. This value must be greater than the value for `low_water_mark`.
- `pool_id`  
specifies the pool identifier. Pool 0 is the common scratch pool, which always exists. You can modify the common scratch pool attributes.
- `overflow`  
specifies that if this pool cannot satisfy `mount scratch` requests, ACSLS will select cartridges from the common pool (Pool 0).

## Usage

Use the `define pool` command to create or modify scratch pools.

## Examples

- To define a low threshold of 0, a high threshold of 600, and overflow for new pool 1:  

```
define pool 0 600 1 overflow
```
- To define a low threshold of 0, a high threshold of 600, and no overflow for existing pool 5:  

```
define pool 0 600 5
```

### Note:

Scratch pools are not owned by one client application or user ID. You can, however, use volume access control to restrict access for specific scratch cartridges.

See also:

| For information about...                                  | See...                             |
|-----------------------------------------------------------|------------------------------------|
| Cancelling a command                                      | <a href="#">cancel</a>             |
| Deleting empty scratch pools                              | <a href="#">delete pool</a>        |
| Guidelines and procedures for managing scratch cartridges | <a href="#">Populating the LSM</a> |
| Mounting a scratch cartridge on a drive                   | <a href="#">mount *</a>            |
| Displaying scratch pool attributes                        | <a href="#">query pool</a>         |
| Setting or clearing cartridge scratch attributes          | <a href="#">set scratch</a>        |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- The following message appears when a `define pool` command succeeds:  

```
Define: Define completed, Success.
```
- The following message appears for each created or modified pool:  

```
Define: Pool pool_id created.
```

### Intermediate Messages:

None.

### Error Messages:

None.

## Display Area Messages

- Pool `pool_id`: low water mark warning.

Explanation: The number of volumes in the specified scratch pool is less than or equal to the low volume threshold.

Variable: `low_water_mark` is the low volume threshold of the specified scratch pool.

- Pool `pool_id`: high water mark warning.

Explanation: The number of cartridges in the specified scratch pool is greater than or equal to the high volume threshold.

Variable: `high_water_mark` is the high volume threshold of the specified scratch pool.

## delete pool

The `delete pool` command deletes empty scratch pools.

## Format

```
delete pool pool_id...|all
```

## Options

- `pool_id`  
specifies the pool ID.  
Pool 0 is the common pool, which you cannot delete.
- `all`  
specifies all empty scratch pools.

## Usage

Use the `delete pool` command to delete empty scratch pools. If a pool contains scratch cartridges, you must reassign these cartridges to another pool before deleting the first pool. If a scratch cartridge is mounted, it becomes a data cartridge but remains in its scratch pool. Use the `set scratch off` command to reassign the data cartridge to the common pool.

## Examples

- To delete all empty scratch pools:

```
delete pool all
```

Only the empty pools will be deleted; pools with cartridges assigned are not affected.

- To delete scratch pool 1, follow the procedure below:

1. Query scratch pool 1:  
`query scratch 1`

```
1998-0630>09:35:30>Scratch Status
Scratch Pool> Identifier> Homer location> Status> Type
1) 34813> 0,0,1,8,8> home> 3480 3480
1) 34815> 0,0,1,8,1> home> 3480 3480
```

Pool 1 has two cartridges, 34813 and 34815.

2. Empty pool 1 by reassigning its cartridges to pool 5:

```
set scratch 5 348013 348015
```

3. Delete pool 1:

```
delete pool 1
```

See also:

| For information about...                                  | See...                             |
|-----------------------------------------------------------|------------------------------------|
| Cancelling a command                                      | <a href="#">cancel</a>             |
| Creating or modifying scratch pools                       | <a href="#">define pool</a>        |
| Guidelines and procedures for managing scratch cartridges | <a href="#">Populating the LSM</a> |
| Displaying scratch pool attributes                        | <a href="#">query pool</a>         |
| Displaying the location and media type of a cartridge     | <a href="#">query volume</a>       |
| Setting or clearing cartridge scratch attributes          | <a href="#">set scratch</a>        |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- The following message appears when a `delete pool` command completes successfully:  
Delete: Delete completed, Success.
- The following message appears for each deleted pool:  
Delete: Pool pool\_id deleted.

### Intermediate Messages:

None.

### Error Messages:

- Delete: Pool pool\_id failed, Pool not empty.  
Explanation: ACSLS cannot delete the scratch pool because it is not empty.  
Variable: pool\_id is the identifier of the requested pool.

## dismount

The `dismount` command dismounts a cartridge from a drive.

## Format

```
dismount vol_id drive_id [force]
```

## Options

- vol\_id

specifies the cartridge.

- `drive_id`

specifies the drive.

- `force`

forces a dismount of the actual volume in the specified drive, even if the `vol_id` of the cartridge in the drive does not match the specified `vol_id` (provided that drive is not in `STATUS_DRIVE_RESERVED` state, in which case, the `vol_ids` must match; this prevents client conflicts while a mount or dismount operation is in progress).

This option also forces a dismount, even if the drive is not unloaded.

## Usage

Use the `dismount` command to dismount a volume from a drive, and place the cartridge in an available storage cell.

## Dismount

Use the `dismount` command without the `force` option to dismount a specified cartridge from a specified drive. A successful unforced dismount requires the following:

- The drive must be online
- The `vol_id` of the cartridge in the drive must match the `vol_id` you specify on the `dismount` command
- The drive must be unloaded

### ▲ Caution:

On a normal dismount, SL500 and SL150 libraries automatically rewind and unload a cartridge that is found loaded in a drive. They do not require a dismount force to rewind and unload a cartridge. Make sure that an ACSLS client application is not reading from, or writing to a tape drive before issuing a dismount on these libraries.

## Dismount force

Use the `dismount` command with the `force` option to force a dismount of the actual cartridge mounted in a specified drive. The `vol_id` of the cartridge in the drive does not have to match the `vol_id` you specify on the `dismount` command (unless the drive is in `STATUS_DRIVE_RESERVED` state; this prevents client conflicts while a mount or dismount operation is in progress). In addition, if the cartridge is not ready for dismounting, ACSLS forces the drive to automatically rewind, unload, and dismount the cartridge. The drive must be online.

A `dismount force` must include the correct `vol_id` if a drive was left marked as reserved by a mount or dismount operation that never received a response. If the drive has no `vol_id` (as seen in the results of a `query drive` command), then any `vol_id` may be used. Match is not required, since there is nothing to match. A reserved drive is reported as in use to the `cmd_proc` or an ACSAPI client.



You can use the `force` option to dismount a cartridge with an unreadable or unknown label or a cartridge that a client application did not dismount. Notice that ACSLS returns the cartridge to an available storage cell, even if the label is unreadable or missing.



#### Note:

Any read/write activity between the drive and any application that is currently using the drive must stop or pause before the library rewinds and uploads the cartridge. If the read/write activity continue uninterrupted, the dismount force fails after the library wait timeout is exceeded.

This consideration also applies to media validation that was requested from the host to the drive, by way of the data path. Until the media validation completes or the host stops it through the data path, a dismount force from ACSLS fails.

## Examples

To dismount cartridge EDU200 from drive 0,1,10,2:

```
dismount EDU200 0,1,10,2
```

See also:

| For information about...                                                       | See...                           |
|--------------------------------------------------------------------------------|----------------------------------|
| Cancelling a command                                                           | <a href="#">cancel</a>           |
| Ejecting cartridges from the library                                           | <a href="#">eject</a>            |
| Making a CAP (manual mode) ready to enter labelled cartridges into the library | <a href="#">enter</a>            |
| Mounting a scratch cartridge on a drive                                        | <a href="#">mount *</a>          |
| Mounting a data cartridge on a drive                                           | <a href="#">mount</a>            |
| Displaying CAP status                                                          | <a href="#">query cap</a>        |
| Displaying drive status                                                        | <a href="#">query drive</a>      |
| Setting CAP mode (manual or automatic)                                         | <a href="#">set cap mode</a>     |
| Setting CAP selection priority                                                 | <a href="#">set cap priority</a> |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- `Dismount: vol_id` dismounted from `drive_id`.

Explanation: A forced dismount succeeded.

Variable:

- `vol_id` is the identifier of the dismounted cartridge.

The `vol_id` displayed is the volume that was actually dismounted, not necessarily the volume that was specified in the dismount force command.

- `drive_id` is the identifier of the specified drive.
- `Dismount: Forced dismount of vol_id from drive_id.`

**Explanation:** A forced dismount succeeded.

**Variable:**

- `vol_id` is the identifier of the dismounted cartridge.  
The `vol_id` displayed is the volume that was actually dismounted, not necessarily the volume that was specified in the `dismount force` command.
- `drive_id` is the identifier of the specified drive.

#### Intermediate Messages:

None.

#### Error Messages:

- `Dismount: Dismount failed, ACS acs_id full.`

**Explanation:** ACSLS cannot dismount the cartridge because the ACS has no free storage cells. Eject at least one other cartridge from the ACS so that ACSLS can dismount the cartridge from the drive.

**Variable:** `acs_id` is the identifier of the ACS that contains the cartridge.

#### Note:

Only online LSMs are used in locating free cells; therefore, this message can occur even when there is an LSM with free storage cells if that LSM is offline. Additionally, if an LSM or ACS has not been audited, the recorded number of free cells may not be accurate, resulting in this error. Audit the library to remedy this condition.

- `Dismount: Dismount failed, Audit in progress.`

**Explanation:** ACSLS cannot dismount the cartridge because an audit in progress has locked out access to the last unoccupied cell location in the ACS.

- `Dismount: Dismount failed, Cartridge in drive drive_id, unreadable label`

**Explanation:** ACSLS cannot dismount the cartridge because the cartridge either has no external label, an unreadable external label, or no virtual label. Use a forced dismount to dismount the cartridge.

**Variable:** `drive_id` is the identifier of the specified drive.

- `Dismount: Dismount failed, Drive identifier drive_id available.`

**Explanation:** No cartridge is mounted in the specified drive.

**Variable:** `drive_id` is the identifier of the specified drive.

- `Dismount: Dismount failed, Drive identifier drive_id in use.`

**Explanation:** ACSLS cannot dismount the cartridge because the cartridge is not rewound and unloaded. Either wait for the client application to either rewind and unload the cartridge, then enter a `dismount` command, or use a forced dismount to dismount the cartridge.

Variable: `drive_id` is the identifier of the specified drive.

- `Dismount: Dismount failed, Misplaced tape.`

Explanation: ACSLS cannot dismount the cartridge because the external label of the cartridge does not match the cartridge identifier in the ACSLS database. The cartridge identifier in the ACSLS database is updated to match the external label of the cartridge. Retry the dismount.

- `Dismount: Dismount failed, Cartridge not in drive.`

Explanation: ACSLS cannot dismount the cartridge because the external label of the cartridge does not match the cartridge identifier specified in the dismount. Re-enter the `dismount` command with the correct cartridge identifier.

## Display Area Messages

None.

## eject

The `eject` command directs the robot to take cartridges from the inside of an LSM, and place them into a CAP where they can be removed by the operator.

If you select the `lsm_id` option, you can use multiple CAPs in a single LSM to eject cartridges.

## Format

```
eject cap_id|lsm_id [opmsg opmsg_nbr] vol_id|volrange...
```

## Options

- `cap_id`  
specifies the CAP used to eject the cartridges.

- `lsm_id`

If you select the `lsm_id` option, you can use multiple CAPs that are available in a single LSM to eject cartridges. All CAPs that satisfy the requirements (such as manual, nonzero priority available) are unlocked for ejecting cartridges. You can eject cartridges through any or all of the CAPs selected, and in any order. The `eject` initially fills the CAPs with cartridges in order of highest priority.

Example: You have two CAPs - one with priority 2 and one with priority 5; `eject` fills the CAP with priority 5 status first, then the CAP with priority 2 status. If there are only enough cartridges to be ejected for one CAP, then the CAP with priority 5 is filled.

- `opmsg opmsg_nbr`

Custom operator panel message numbers can be specified for the SL8500 Bulk CAP eject commands that are entered using `cmd_proc`. The message is displayed when a Bulk CAP is unlocked so cartridges being ejected can be removed.

Valid message numbers are from 4 and 99.

- Currently a custom `opmsg` is only displayed for the Bulk CAP on SL8500 libraries.

- The `opmsg` parameter is optional. If it is not specified, a default message to remove cartridges is sent.
- Custom `opmsg` numbers cannot be specified for ejects from ACSAPI clients, the ACSLS GUI, or `lib_cmd eject`. In these cases the default messages are displayed.

To create the message to be displayed for an `opmsg` number use the SL Console and select the following:

```
Tools
  Configuration
    CAP Usage Message
```

To view the SL Console CAP Status display showing the `opmsg`, select:

```
Tools
  System Detail
    CAP Folder
      Status
```

The SL Console displays a Bulk CAP message based on the message number sent from ACSLS during an eject. The message displays on the `System Details CAP Status` page after a CAP is unlocked for cartridges being ejected to be removed.

**Example:** To specify custom operator panel message number 55 for ejecting cartridges through Bulk CAP 1,2,1:

```
eject 1,2,1 opmsg 55 T10001 T10033-T10067
```

- `vol_id | volrange`  
specifies the external or virtual label types of the cartridge or range of cartridges to eject.

## Usage

Use the `eject` command to eject cartridges from the library, which removes them from ACSLS control. The robot places the specified cartridge in the designated CAP, then ACSLS frees the cell locations where the cartridges were stored. Cartridge information is either retained if the `ABSENT_VOLUME_RETENTION_PERIOD` variable has a nonzero value, or deleted if the `ABSENT_VOLUME_RETENTION_PERIOD` is set to zero. If you specify more than a CAP full of cartridges on the `eject` command, empty the CAP when it fills, close the CAP, and continue the ejection until all cartridges are ejected.

You can use a single `eject` command to eject multiple cartridges not in a range by specifying multiple cartridge IDs separated by a space.

If your library has two connected LSMs and the PTP is down, to successfully eject cartridges, either:

- Specify a CAP in the LSM where the cartridge is stored. For example, if cartridge NN0100 is stored in LSM 0,0, to eject this cartridge through CAP 0,0,0:  

```
eject 0,0,0 NN0101
```
- For the `cap_id`, specify the LSM where the cartridge is stored but use a wildcard (\*) for the CAP number; ACSLS will select the highest priority CAP in the LSM. For example, if cartridge NN0114 is stored in LSM 0,0, to eject this cartridge through LSM 0,0's highest priority CAP:

```
eject 0,0,* NN0114
```

## Examples

- To eject cartridge NN0101 through CAP 0,0,0:  

```
eject 0,0,0 NN0101
```
- To eject cartridge range NN0101-NN0109 through the highest priority CAP on ACS 0:  

```
eject 0,* NN0101-NN0109
```
- To eject cartridges NN0101, NN0103, NN0105, and NN0107 through the highest priority CAP on ACS 0:  

```
eject 0,* NN0101 NN0103 NN0105 NN0107
```
- To use multiple non-zero priority CAPs for ejecting cartridges in LSM 1,2:  

```
eject 1,2 RB1000-RB2000
```

### Note:

Make sure you remove all ejected cartridges from the CAP. You must complete the ejection by unloading all ejected cartridges and closing the CAP door before you can use the CAP for another operation (such as an enter or audit).

See also:

| For information about...                                                       | See...                             |
|--------------------------------------------------------------------------------|------------------------------------|
| Cancelling a command                                                           | <a href="#">cancel</a>             |
| Dismounting a cartridge from a drive                                           | <a href="#">dismount</a>           |
| Guidelines and procedures for ejecting cartridges                              | <a href="#">Populating the LSM</a> |
| Making a CAP (manual mode) ready to enter labelled cartridges into the library | <a href="#">enter</a>              |
| Displaying CAP status                                                          | <a href="#">query cap</a>          |
| Displaying drive status                                                        | <a href="#">query drive</a>        |
| Displaying the location and media type of a cartridge                          | <a href="#">query volume</a>       |
| Setting CAP mode (manual or automatic)                                         | <a href="#">set cap mode</a>       |
| Setting CAP selection priority                                                 | <a href="#">set cap priority</a>   |

## Command Area Messages

This section discusses command area messages.

### Success Messages:

- Eject: Eject complete, a cartridges ejected
- Eject: vol\_id ejected from cap\_id  
Explanation: ACSLS ejected the specified cartridges.

Variable:

- nn is the number of cartridges ejected.

- `vol_id` is the cartridge identifier of the ejected cartridge.
- `cap_id` is the CAP that contains the ejected cartridge.

**Intermediate Messages:**

None.

**Error Messages:**

- `Eject: vol_id Eject failed, CAP cap_id full.`  
Explanation: A cartridge was not ejected because the CAP is full.  
Variable:
  - `vol_id` is the cartridge identifier of the cartridge that was not ejected.
  - `cap_id` is the CAP specified for the ejection.
- `Eject: vol_id Eject failed, CAP cap_id in use.`  
Explanation: A cartridge was not ejected because the CAP is in use.  
Variable:
  - `vol_id` is the cartridge identifier of the cartridge that was not ejected.
  - `cap_id` is the CAP specified for the ejection.
- `Eject: vol_id Eject failed, Misplaced tape.`  
Explanation: ACSLS cannot eject the cartridge because the external label of the cartridge does not match the database's cartridge identifier for the storage cell. The database updates the cartridge identifier, media type, and cleaning cartridge attribute for the cartridge in the storage cell.  
Variable: `vol_id` is the cartridge identifier of the cartridge that was not ejected.
- `Eject: vol_id Eject failed, Not in same ACS.`  
Explanation: A cartridge was not ejected because it is not in the ACS specified in the `cap_id`.  
Variable: `vol_id` is the cartridge identifier of the cartridge that was not ejected.
- `Eject: vol_id Eject failed, Volume identifier vol_id not found.`  
Explanation: A cartridge was not ejected because it is not:
  - In the storage cell specified in the database,
  - In-transit, or
  - In a drive.
- ACSLS deletes the volume entry from the database.  
Variable: `vol_id` is the cartridge identifier that was deleted from the database.
- `Eject: vol_id Eject failed, Cartridge in drive.`  
Explanation: A cartridge was not ejected because it is mounted in a drive.  
Variable: `vol_id` is the cartridge identifier of the cartridge that was not ejected.
- `Eject: vol_id Eject failed, Volume vol_id in use.`

Explanation: A cartridge was not ejected because it is reserved for use by another request.

Variable: `vol_id` is the cartridge identifier of the cartridge that was not ejected.

## Display Area Messages

- CAP `cap_id` Remove cartridges from CAP.

Explanation: Either the CAP is full, or all requested cartridges are in the CAP. Empty the CAP.

Variable: `cap_id` is the CAP that contains the ejected cartridges.

- CAP `cap_id` Place magazines in CAP.

Explanation: The CAP requires magazines for the ejection. Open the CAP, place magazines inside, then close the CAP.

Variable: `cap_id` is the CAP that requires magazines.

## enter

This command allows you to set the CAP to operate in either manual or automatic mode. For procedures for setting the CAP to automatic mode or manual mode, refer to the section [Entering Cartridges](#).

- Automatic mode

When a CAP is in automatic mode, you can initiate an enter operation without issuing an enter command. You do this by opening the CAP door, placing one or more cartridges inside, and closing the CAP. The CAP is locked while the enter is being processed; when the enter operation is complete, the CAP is unlocked.

- Manual mode

When in manual mode, the CAP is locked and can only be used by issuing the following command before opening the CAP and entering the cartridges.

```
enter cap_id [opmsg opmsg_nbr]
```

## Options

- `cap_id`

specifies the CAP. This can be wild-carded with an asterisk (\*) to select the CAP with the highest non-zero priority in an LSM, such as 1,1,\* or an ACS, such as 0,\*.\*

- `opmsg opmsg_nbr`

Custom operator panel message numbers can be specified for the SL8500 Bulk CAP enter commands that are entered using `cmd_proc`. The message is displayed when a Bulk CAP is unlocked for cartridges being entered.

Valid message numbers are from 4 and 99.

- Currently a custom `opmsg` is only displayed for the Bulk CAP on SL8500 libraries.
- The `opmsg` parameter is optional. If it is not specified, a default message to enter cartridges is sent.

- When cartridges were left in the CAP before the enter started or unreadable or duplicate cartridges cannot be entered, a message to remove these cartridges is displayed. This can also happen when an `opmsg` parameter was specified for the enter.
- Custom `opmsg` numbers cannot be specified for enters from ACSAPI clients or the ACSLS GUI. In these cases the default messages are displayed.

To create the message to be displayed for an `opmsg` number use the SL Console and select the following:

```
Tools
  Configuration
    CAP Usage Message
```

To view the SL Console CAP Status display showing the `opmsg`, select:

```
Tools
  System Detail
    CAP Folder
      Status
```

The SL Console displays a Bulk CAP message based on the message number sent from ACSLS during an enter. The message displays on the `System Details CAP Status` page after a CAP is unlocked for cartridges being entered.

**Example:** To specify custom operator panel message number 66 for entering cartridges through Bulk CAP 1,3,0:

```
enter 1,3,0 opmsg 66
```

- `lsm_id`

If you select the `lsm_id` option, you can use multiple CAPs that are available in a single LSM to enter cartridges. All CAPs that satisfy the requirements (such as manual, non zero priority, available) are unlocked for entering cartridges. You can enter cartridges through any or all the CAPs selected, and in any order. `venter` is not valid and returns a CAP invalid error.

## Usage

Use the `enter` command to make a manual mode CAP ready to enter labelled cartridges.

## Examples

- To make CAP 0,0,2 ready to enter cartridges:  

```
enter 0,0,2
```
- To make the highest non-zero priority CAP in LSM 0,0 ready to enter cartridges:  

```
enter 0,0,*
```
- To use multiple CAPs for entering cartridges in an LSM 1,2:  

```
enter 1,2
```



 **Note:**

Cartridges being entered into an LSM must be compatible with that LSM. For example, only T9840, T9940, and LTO cartridges can be entered into an L5500 LSM.

See also:

| For information about...                          | See...                              |
|---------------------------------------------------|-------------------------------------|
| Cancelling a command                              | <a href="#">cancel</a>              |
| Creating or modifying scratch pools               | <a href="#">define pool</a>         |
| Ejecting cartridges from the library              | <a href="#">eject</a>               |
| Guidelines and procedures for entering cartridges | <a href="#">Entering Cartridges</a> |
| Mounting a scratch cartridge on a drive           | <a href="#">mount *</a>             |
| Mounting a data cartridge on a drive              | <a href="#">mount</a>               |
| Displaying CAP status                             | <a href="#">query cap</a>           |
| Setting CAP mode (manual or automatic)            | <a href="#">set cap mode</a>        |
| Setting CAP selection priority                    | <a href="#">set cap priority</a>    |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

After all labelled cartridges have been entered, the following message appears, with the second line of the message repeated for each cartridge successfully entered:

- Enter: Enter complete, nn volumes entered
- Enter: vol\_id Entered through cap\_id
  - nn is the total number of cartridges entered
  - vol\_id is the cartridge identifier of the entered cartridge
  - cap\_id is the CAP used to enter the cartridges

### Intermediate Messages:

None.

### Error Messages:

- Enter: vol\_id Enter failed, ACS acs\_id full.

**Explanation:** A cartridge was not entered because the ACS has no free storage cells. You must `eject` at least one cartridge from the ACS so that ACSLS can enter the cartridge.

#### Variable:

- vol\_id is the external label of the cartridge that was not entered.
- acs\_id is the identifier of the ACS that has no free storage cells.

 **Note:**

ACSLs only searches online LSMs for free cells; this message can occur even if offline LSMS have free cells.

- Enter: `vol_id` Enter failed, Audit in progress.  
**Explanation:** A cartridge was not entered because an `audit` has locked out access to a cell location required for the enter.  
**Variable:** `vol_id` is the external label of the cartridge that was not entered.
- Enter: `vol_id` Enter failed, CAP `cap_id` in use.  
**Explanation:** A cartridge was not entered because the specified CAP is in use for auditing, ejecting cartridges, or another enter process.  
**Variable:**
  - `vol_id` is the external label of the cartridge that was not entered.
  - `cap_id` is the in-use CAP.
- Enter: `vol_id` Enter failed, Duplicate label.  
**Explanation:** A cartridge was not entered because the cartridge identifier of a cartridge in the CAP already exists in the ACSLS database.  
**Variable:** `vol_id` is the external label of the cartridge that was not entered.
- Enter: Enter failed, Unreadable label.  
**Explanation:** A cartridge was not entered because the cartridge either has no external label or an unreadable external label.
- Enter: `vol_id` Enter failed, Unknown media type label.  
**Explanation:** A cartridge was not entered because the cartridge does not have a media identifier on its external label.  
**Variable:** `vol_id` is the external label of the cartridge that was not entered.

## Display Area Messages

- CAP `cap_id`: Place cartridges in CAP.  
**Explanation:** The CAP is ready for cartridge entry. Open the CAP and enter cartridges.  
**Variable:** `cap_id` is the CAP used to enter cartridges.
- CAP `cap_id`: Remove cartridges from CAP.  
**Explanation:** One or more cartridges cannot be entered.  
**Variable:** `cap_id` is the CAP used to enter cartridges. Open the CAP and remove the cartridges.
- CAP `cap_id`: CAP `cap_id` Place magazines in CAP.  
**Explanation:** The CAP uses magazines for cartridge entry. Load cartridges in the correct magazine, open the CAP, and insert the magazines.  
**Variable:** `cap_id` is the CAP used to enter cartridges.

## idle

The `idle` command stops ACSLS from processing new requests.

## Format

- `idle [force]`

Enter the full command name for the `idle` command. ACSLS will reject all other forms of the command, such as `i`, `id`, or `idl`.

## Options

- `force`  
forces termination of new request processing.

## Usage

Use the `idle` command to stop ACSLS from processing new requests. For example, you `idle` ACSLS before doing maintenance or before terminating ACSLS.



### Note:

Use the `start` command to restart the request processing.

- `idle`  
Entering an `idle` command without the `force` option puts ACSLS in the idle-pending state. ACSLS completes current and pending requests (except pending lock requests, which are canceled) and rejects new requests except those listed in "Notes." ACSLS then goes to idle state and does not process subsequent requests until request processing is restarted.
- `Idle force`  
Entering an `idle` command with the `force` option puts ACSLS in the idle state. ACSLS cancels all current and pending requests and rejects new requests except those listed in the Note under "Examples". ACSLS does not process subsequent requests until restarted. Because ACSLS does not complete current requests, forcing ACSLS to idle state may leave the database in an inconsistent state with the hardware, which requires an `audit` to correct.

**HINT:** Depending on the current requests processing when you enter an `idle force`, the Event Log may report process failures. If so, `vary` the affected LSMs offline and back online. See "`vary`" for information about the `vary` command.

## Examples

- To put ACSLS in idle-pending state:  
`idle`

- To force ACSLS to idle state:

```
idle force
```

 **Note:**

In either `idle` or `idle-pending` state, ACSLS accepts new requests for the `cancel`, `idle`, `query`, `start`, and `vary` requests.

See also:

| For information about...                     | See...                         |
|----------------------------------------------|--------------------------------|
| Displaying the status of a library component | <a href="#">query commands</a> |
| Starting ACSLS request processing            | <a href="#">start</a>          |
| Changing the state of a library component    | <a href="#">vary</a>           |

## Command Area Messages

This section discusses the area command messages.

### Success Messages:

The following message appears when ACSLS request processing stops:

```
ACSLM Request Processing Stopped: Success.
```

### Intermediate Messages:

None.

### Error Messages:

```
ACSLM Request Processing Stopped: status
```

Explanation: ACSLS did not stop processing requests.

Variable: `status` is the reason for the failure. For more information on common status messages, see ACSLS Messages.

## Display Area Messages

Variable:

- `Server system idle`

Explanation: Request processing has stopped, and ACSLS is in idle state.

- `Server system idle is pending`

Explanation: ACSLS is processing current or pending requests; idle state is pending.

## lock

The `lock` command locks a cartridge or drive to a lock ID that ACSLS assigns.

## Format

```
lock type identifier...[wait]
```

## Options

- `type identifier`

specifies a library component. The following table lists the components that you can lock. You can specify either a drive or a volume but not both in a single `lock` command. You can, however, use the same `lock ID` for both drives and volumes..

**Table 12-4 Valid Components for Lock**

| Library Component | type   | identifier      |
|-------------------|--------|-----------------|
| drive             | drive  | <i>drive_id</i> |
| volume            | volume | <i>vol_id</i>   |

- `wait`

specifies that the lock is pending if the component is unavailable (locked or in use). ACSLS then `locks` the component when it is available. You can `clear` or `cancel` a pending lock request. Idling ACSLS also cancels a pending lock request.

## Usage

Use the `lock` command to lock a volume or drive to a lock ID that ACSLS assigns. You can only lock available (not locked or in use) volumes or drives.

 **Note:**

When you enter a `lock` command to lock a cartridge or drive, ACSLS assigns a lock ID to the volume or drive and then changes your lock ID to the lock ID of the volume or drive. You cannot use the `set lock` command to set your lock ID and then use the `lock` command to lock a volume or drive with the lock ID that you set with `set lock`.

## Examples

- To lock drive 0,1,10,2:

```
lock drive 0,1,10,2
```

- To lock volume EDU445:

```
lock volume EDU445
```

See also:

| For information about...                                               | See...                     |
|------------------------------------------------------------------------|----------------------------|
| Removing all active or pending locks on a specified drive or cartridge | <a href="#">clear lock</a> |
| Displaying the lock status of a drive or cartridge                     | <a href="#">query lock</a> |
| Setting your lock ID                                                   | <a href="#">set lock</a>   |
| Displaying your lock or user ID                                        | <a href="#">show</a>       |
| Removing active locks on drives or cartridges                          | <a href="#">unlock</a>     |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- The following message appears when a `lock` request succeeds:  
Lock: Lock completed, Success.
- For each identifier in the request, one of the following messages appears:
  - Lock: Drive `drive_id` locked under `lock_id` `lock_id`.
  - Lock: Volume `vol_id` locked under `lock_id` `lock_id`.

Where:

- \* `drive_id` is the locked drive.
- \* The `vol_id` is the cartridge identifier of the locked cartridge.
- \* `lock_id` is the lock ID.

### Intermediate Messages:

None.

### Error Messages:

One of the following messages appears if the `lock` request fails:

- Lock: Lock of drive `drive_id` failed, Drive in use.  
Explanation: ACSLS cannot lock the specified drive because the drive is already locked or in use.  
Variable: `drive_id` is the drive that ACSLS cannot lock.
- Lock: Lock of drive `drive_id` failed, Lock failed.  
Explanation: ACSLS cannot lock the specified drive. Re-enter the `lock` command with the correct syntax and drive identifier.  
Variable: `drive_id` is the drive that ACSLS cannot lock.
- Lock: Lock of volume `vol_id` failed, Volume in use.  
Explanation: ACSLS cannot lock the specified cartridge because the cartridge is already locked or in use.  
Variable: `vol_id` is the cartridge that ACSLS cannot lock.

- Lock: Lock of volume `vol_id` failed, Lock failed.

Explanation: ACSLS cannot lock the specified cartridge. Re-enter the `lock` command with the correct syntax and cartridge identifier.

Variable: `vol_id` is the cartridge that ACSLS cannot lock.

## Display Area Messages

None.

## logoff

The `logoff` command exits from a `cmd_proc`.

## Format

`logoff`

## Options

None.

## Usage

Use the `logoff` command to exit from `cmd_proc`. If you run an interactive (window), `cmd_proc`, `logoff` also quits the `cmd_proc` window.

## Examples

- To exit a `cmd_proc`:

```
logoff
```

 **Note:**

`logoff` exits from `cmd_proc` only, is valid in any ACSLS state, and does not affect ACSLS operations.

See also:

| For information about...         | See...                                         |
|----------------------------------|------------------------------------------------|
| Starting a <code>cmd_proc</code> | <a href="#">Starting <code>cmd_proc</code></a> |
| Using a <code>cmd_proc</code>    | <a href="#">Starting <code>cmd_proc</code></a> |

## Command Area Messages

None.

## Display Area Messages

None.

## mount

The `mount` command mounts a data cartridge.

## Format

```
mount vol_id drive_id [bypass] [readonly]
```

## Options

- `vol_id`  
specifies the cartridge.
- `drive_id`  
specifies the drive.
- `bypass`

The `bypass` option overrides the checks that ACSLS does before trying to mount a cartridge as follows:

- The `bypass` option always overrides the ACSLS verification of the cartridge ID on the external label.
- It may override ACSLS verification of the compatibility between the tape drive and the cartridge's media type.

When the library receives the `mount` request, it verifies that the cartridges' media is compatible with the tape drive; the library fails the mount if the media type is incompatible or is an unknown type.

- `readonly`  
specifies that the cartridge is mounted write-protected.

### **Caution:**

LTO drives do not support mounts with write protect. If a "mount read-only" is attempted to an LTO drive, it will fail with a "Drive cannot honor write protect" message in the Event Log.

## Usage

Use the `mount` command to mount a data cartridge. You can only mount one cartridge on one drive each time you enter a `mount` command.

A successful mount requires the following:

- The cartridge and drive must be in the same ACS.



- The cartridge must be available, and the drive must be online and available.

## Examples

To mount volume EDU010 on drive 0,0,10,2:

```
mount EDU010 0,0,10,2
```

In the following example, YUMA15 is a DD3C volume whose cartridge label is missing the seventh character. The `bypass` option bypasses the media compatibility checks and forces a `mount` of the cartridge on drive 0,0,4,0, an SD3 drive.

To mount YUMA15 on drive 0,0,4,0 using the `bypass` option:

See also:

| For information about...                                                        | See...                           |
|---------------------------------------------------------------------------------|----------------------------------|
| Dismounting a cartridge from a drive                                            | <a href="#">dismount</a>         |
| Making a CAP (manual mode) ready to enter labelled cartridges into the library  | <a href="#">enter</a>            |
| Displaying CAP status                                                           | <a href="#">query cap</a>        |
| Displaying drive status                                                         | <a href="#">query drive</a>      |
| Displaying the lock status of a drive or cartridge                              | <a href="#">query lock</a>       |
| Displaying the status of media-compatible drives for a specified data cartridge | <a href="#">query mount</a>      |
| Displaying the location and media type of a cartridge                           | <a href="#">query volume</a>     |
| Setting CAP mode (manual or automatic)                                          | <a href="#">set cap mode</a>     |
| Setting CAP selection priority                                                  | <a href="#">set cap priority</a> |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

```
Mount: vol_id mounted on drive_id
```

Explanation: ACSLS mounted the specified cartridge.

Variable:

- `vol_id` is the cartridge identifier of the cartridge that ACSLS mounted.
- `drive_id` is the drive that mounted the cartridge.

### Intermediate Messages:

None.

### Error Messages:

- `Mount: Mount failed, Audit in progress.`

Explanation: ACSLS cannot mount the cartridge because an `audit` has locked out access to the cell location of the specified cartridge.

- `Mount: Mount failed, In use.`  
Explanation: ACSLS cannot mount the cartridge because either the drive is in use or the requested cartridge is reserved for another command.
- `Mount: Mount failed, Misplaced tape.`  
Explanation: ACSLS cannot mount the cartridge because the external label of the cartridge does not match the databases' cartridge identifier for the storage cell. The database updates the cartridge identifier, media type, and cleaning cartridge attribute for the cartridge in the storage cell.
- `Mount: Mount failed, Not in same ACS.`  
Explanation: ACSLS cannot mount the cartridge because the specified cartridge and drive are not in the same ACS.
- `Mount: Mount failed, Cartridge in drive.`  
Explanation: ACSLS cannot mount the cartridge because the specified cartridge is already mounted in a drive.
- `Mount: Mount failed, Unreadable label.`  
Explanation: ACSLS cannot mount the cartridge because the cartridge has no label, an unreadable label, or no virtual label.
- `Mount: Mount failed, Invalid media type.`  
Explanation: ACSLS cannot mount the cartridge because the specified cartridges' media type is invalid.
- `Mount: Mount failed, Invalid drive type.`  
Explanation: ACSLS cannot mount the cartridge because the specified drive type is invalid.
- `Mount: Mount failed, Incompatible media type.`  
Explanation: ACSLS cannot mount the cartridge because the cartridges' media type is incompatible with the specified drive.

## Display Area Messages

None.

## mount \*

The `mount *` command selects a scratch cartridge and mounts it.

## Format

```
mount * drive_id [pool_id] [media media_type | media *]
```

## Options

- `drive_id`  
specifies the drive.
- `pool_id`

specifies the pool from which ACSLS selects the scratch cartridge. `pool_id` is optional; if you do not specify a `pool_id`, ACSLS tries to find a scratch cartridge in the common pool (pool 0).

If you specify a `pool_id` and the pool does not contain a scratch cartridge (or one of the correct media for mixed-media libraries), or if the pool is set for overflow; ACSLS tries to find a scratch cartridge in the common pool (pool 0).

- `media media_type | media *`

specifies the cartridge media type. Specifying the media type is optional.

## Usage

Use the `mount *` command to select a scratch cartridge and mount it. The following sections describe how ACSLS selects the scratch cartridge to be mounted and how to specify the cartridge media type for a scratch cartridge.

### How ACSLS Selects Scratch Cartridges

The `mount *` command selects a scratch cartridge by:

- Creating a list of LSMs in the ACS, based on proximity to the LSM containing the specified drive.
- Examining each LSM in the list, until it finds scratch cartridges that meet the pool and media type criteria.
- Choosing the scratch cartridge with the least recent access date in that LSM.

The selected scratch cartridge is then mounted on the drive.

### Specifying the Media Type

You can specify cartridge media type in any of the following ways:

- Explicitly specify the media type in this form of the `mount *` command:

```
mount * drive_id [pool_id] media media_type
```

Example: To mount a T10000 scratch cartridge from the common pool (pool 0):

```
mount * 0,0,10,2 media T10000T1
```

- Use a wildcard (\*) for the media type so that ACSLS selects the media by scratch preferences; for more information see "[Using the Extended Store Feature](#)". To use scratch preferences, enter this form of the `mount *` command:

```
mount * drive_id [pool_id] media *
```

Example: To use scratch preferences to mount a scratch cartridge from the common pool (pool 0):

```
mount * 0,0,10,2 media *
```

- Omit the media option in this form of the `mount *` command. This instructs ACSLS to pick a media type compatible with the drive:

```
mount * drive_id [pool_id]
```

Example: To mount a scratch cartridge compatible with the drive from the common pool (pool 0):

```
mount * 0,0,10,2
```

## Examples

The following sections show mount scratch examples for single-media and mixed-media libraries.

### Single-Media Libraries

- To mount a scratch cartridge from pool 5 on drive 0,0,10,2:

```
mount * 0,0,10,2 5
```

**HINT:** If no cartridge is available from pool 5 and it has been set for `overflow`, ACSLS will select a cartridge from the common pool (pool 0).

- To mount a scratch cartridge from the common pool (pool 0) on drive 0,0,10,0:

```
mount * 0,0,10,0
```

### Mixed-Media Libraries

- To mount a scratch cartridge from pool 5 with a media type of T10000T2 on drive 0,0,10,2:

```
mount * 0,0,10,2 5 media T10000T2
```

If no cartridge is available from pool 5 and it has been set for `overflow`, ACSLS will select a cartridge with the specified media type from the common pool (pool 0).

- To mount a scratch cartridge from pool 10 with media type determined by scratch preferences on drive 0,0,2,3:

```
mount * 0,0,2,3 10 media *
```

**HINT:** If no cartridge is available from pool 10 and it has been set for `overflow`, ACSLS will select a cartridge with the specified media type from the common pool (pool 0).

- To mount a scratch cartridge from the common pool (pool 0) with media type of T10000T2 on drive 0,0,10,2:

```
mount * 0,0,10,2 media T10000T2
```

- To mount a scratch cartridge from the common pool (pool 0) with media type determined by scratch preferences on drive 0,0,2,3:

```
mount * 0,0,2,3 media *
```

- To mount a scratch cartridge from the common pool (pool 0) with media compatible with drive 0,0,2,3:

```
mount * 0,0,2,3
```

See also:

| For information about...             | See...                                 |
|--------------------------------------|----------------------------------------|
| Creating or modifying scratch pools  | " <a href="#">define pool</a> "        |
| Deleting empty scratch pools         | " <a href="#">delete pool</a> "        |
| Dismounting a cartridge from a drive | " <a href="#">dismount</a> "           |
| Managing scratch cartridges          | " <a href="#">Populating the LSM</a> " |
| Displaying drive status              | " <a href="#">query drive</a> "        |

| For information about...                                                      | See...                                               |
|-------------------------------------------------------------------------------|------------------------------------------------------|
| Displaying the lock status of a drive or cartridge                            | " <a href="#">query lock</a> "                       |
| Displaying the status of media-compatible drives for a specified scratch pool | " <a href="#">query mount *</a> "                    |
| Displaying scratch pool attributes                                            | " <a href="#">query pool</a> "                       |
| Displaying scratch cartridge status                                           | " <a href="#">query scratch</a> "                    |
| Displaying the location and media type of a cartridge                         | " <a href="#">query volume</a> "                     |
| Setting or clearing cartridge scratch attributes                              | " <a href="#">set scratch</a> "                      |
| Setting scratch preferences                                                   | " <a href="#">Using the Extended Store Feature</a> " |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- `Mount: vol_id mounted on drive_id`  
Explanation: ACSLS mounted the specified cartridge.  
Variable:
  - `vol_id` is the cartridge identifier of the cartridge that ACSLS mounted.
  - `drive_id` is the drive that mounted the cartridge.

### Intermediate Messages:

None.

### Error Messages:

- `Mount: Mount failed, Audit in progress.`  
Explanation: ACSLS cannot mount the cartridge because an `audit` has locked out access to the cell location of the specified cartridge.
- `Mount: Mount failed, In use.`  
Explanation: ACSLS cannot mount the cartridge because either the drive is in use or the requested cartridge is reserved for another command.
- `Mount: Mount failed, Misplaced tape.`  
Explanation: ACSLS cannot `mount` the cartridge because the external label of the cartridge does not match the database cartridge identifier for the storage cell. The database updates the cartridge identifier, media type, and cleaning cartridge attribute for the cartridge in the storage cell.
- `Mount: Mount failed, Not in same ACS.`  
Explanation: ACSLS cannot `mount` the cartridge because the specified cartridge and drive are not in the same ACS.
- `Mount: Mount failed, Cartridge in drive.`  
Explanation: ACSLS cannot mount the cartridge because the specified cartridge is already mounted in a drive.

- Mount: Mount failed, Unreadable label.  
Explanation: ACSLS cannot mount the cartridge because the cartridge has no label, an unreadable label, or no virtual label.
- Mount: Mount failed, Invalid media type.  
Explanation: ACSLS cannot mount the cartridge because the specified cartridges' media type is invalid.
- Mount: Mount failed, Invalid drive type.  
Explanation: ACSLS cannot mount the cartridge because the specified drive type is invalid.
- Mount: Mount failed, Incompatible media type.  
Explanation: ACSLS cannot mount the cartridge because the cartridges' media type is incompatible with the specified drive.
- Mount: Mount failed, No compatible scratch cartridges in pool.  
Explanation: ACSLS cannot mount the cartridge because there are no scratch cartridges in the ACS of the specified drive that match the scratch cartridges' media type. In addition, if the overflow attribute is set for the pool, there are no scratch cartridges with valid media types.

## Display Area Messages

- Pool pool\_id: low water mark warning.  
Explanation: The number of cartridges in the specified scratch pool is less than or equal to the low cartridge threshold.  
Variable: low\_water\_mark is the low threshold of the specified scratch pool.
- Pool pool\_id: high water mark warning.  
Explanation: The number of cartridges in the specified scratch pool is greater than or equal to the high cartridge threshold.  
Variable: high\_water\_mark is the high water cartridge threshold of the specified scratch pool.

## move

The `move` command moves a specified cartridge to an available storage cell in a specified LSM or to a specific storage cell.



### Note:

SL3000 allows for cartridges to be moved to a specific cell. For more information, see [Partitioning a Library or Changing Partition IDs](#).

## Format

```
move vol_id lsm_id or move vol_id cell_id
```

## Options

- `vol_id`  
specifies the cartridge.
- `lsm_id`  
specifies the LSM that will contain the moved cartridge.
- `cell_id`  
specifies the cell to which the cartridge will be moved.

## Usage

Use the `move` command to move a specified cartridge to an available storage cell in either:

- A different panel in the same LSM. For example, if you want to empty an entire panel in an LSM, move all of the cartridges in that panel to a different location in the same LSM.
- A different LSM.

You can move only one cartridge each time you enter the `move` command. If you specify the LSM where the cartridge currently resides, ACSLS will move the cartridge to another panel within that LSM. Otherwise, ACSLS will move it to the LSM you specify. You cannot cancel a `move` command.

A successful move requires the following:

- The cartridge must be available and in the same ACS as the specified LSM.
- The LSM where the cartridge currently resides, and the specified LSM, must both be online. The specified LSM must have at least one available storage cell. If the move is within the same LSM, there must be at least one available cell in a different panel in that LSM. If the move requires a pass-thru, any LSMs used must also be online.

## Examples

- To move cartridge EDU010 (which resides in LSM 0,1) to a different panel in this LSM:

```
move EDU010 0,1
```

- To move cartridge EDU010 (which resides in LSM 0,1) to LSM 0,2:

```
move EDU010 0,2
```

See also:

| For information about...                              | See...                       |
|-------------------------------------------------------|------------------------------|
| Displaying the lock status of a drive or cartridge    | <a href="#">query lock</a>   |
| Displaying the status of an LSM                       | <a href="#">query lsm</a>    |
| Displaying the location and media type of a cartridge | <a href="#">query volume</a> |

## Command Area Messages

This section discusses the command area messages.

**Success Messages:**

- Move: `vol_id` moved to location `cell_id`

Explanation: ACSLS moved the specified cartridge to the specified cell location.

Variable:

- `vol_id` is the cartridge identifier of the cartridge that ACSLS moved.
- `cell_id` is the new cell location of the specified cartridge.

**Intermediate Messages:**

None.

**Error Messages:**

Move: Move failed

## query commands

The `query` commands display the status of a library component. See the following sections for more information about each `query` command, including format, options, and usage.

## Format

The following shows the general format of the `query` commands:

```
query type [subtype | *] identifier... | all
```

**Note:**

If a `cancel` request is issued against a pending or current `query` request, the display of information is halted.

See also:

| For information about... | See...                                            |
|--------------------------|---------------------------------------------------|
| Cancelling a command     | " <a href="#">cancel</a> "                        |
| Display Command          | " <a href="#">Using display Command Options</a> " |

## Command Area Messages

This section discusses the command area messages.

**Success Messages:**

None.



**Note:**

If the `query` succeeds, the requested status appears.

**Intermediate Messages:**

None.

**Error Messages:**

- Library not available.

Explanation: The `query` failed because ACSLS is recovering (all queries except query server).

- Cartridge identifier `vol_id` not found.

Explanation: The `query mount *` command cannot display status for the specified cartridge because it is not in the library.

Variable: `vol_id` is the specified cartridge.

- Invalid media type

Explanation: The `query mount *` command cannot display status for the specified cartridge because it either has an invalid media type or you specified an invalid media type on the command.

Variable: `vol_id` is the specified cartridge.

## Display Area Messages

None.

## query acs

The `query acs` command displays ACS status.

## Format

```
query acs acs_id... | all
```

## Options

```
acs_id | all
```

specifies the ACS to `query` or `all` for all ACSs.

## Usage

Use the `query acs` command to display the status of an ACS in the following format:

```
yyy-mm-dd hh:mm:ss ACS Status
Identifier State Free Cell Audit Mount Dismount Enter Eject
Count C/P C/P C/P C/P C/P
```

```
acs_id      state  count      n/n   n/n   n/n      n/n   n/n
```

**Where:**

- `acs_id`  
is the ACS identifier.
- `state` is one of the following ACS states:
  - `online`  
The ACS is online.
  - `offline`  
The ACS is offline.
  - `offline pending`  
The ACS will process current and pending requests, then go offline. The ACS will reject any new requests.
  - `diagnostic`  
The ACS will process only current and pending requests and will reject any new requests. The ACS is unavailable to client applications and can only be controlled using `cmd_proc`. Use the `vary` command to vary the ACS online.
  - `recovery`  
The ACS is initializing or recovering from errors. Wait for the ACS to go online.
- `count`  
is the number of free cells in the ACS.
- `n`  
is the number of current (C) and pending (P) requests for the ACS for each command that requires library resources (`audit`, `mount`, `dismount`, `enter`, and `eject`).

## Examples

- To query ACS 1:  

```
query acs 1
```
- To query all ACSs in the library:  

```
query acs all
```

**See also:**

| For information about...                  | See...                           |
|-------------------------------------------|----------------------------------|
| Changing the state of a library component | <a href="#">"vary "</a>          |
| Displaying request status                 | <a href="#">"query request "</a> |

## query cap

The `query cap` command displays CAP status.

### Format

```
query cap cap_id... | all
```

### Options

- `cap_id | all`  
specifies the CAP to `query` or `all` for all CAPs.

 **Note:**

You cannot specify a `cap_id` that contains asterisks.

### Usage

Use the `query cap` command to display the status of a CAP.

The `query cap` command displays CAP status in the following format:

| yyy-mm-dd  | hh:mm:ss     | CAP      | Status    |          |        |  |
|------------|--------------|----------|-----------|----------|--------|--|
| Identifier | Priority     | Size     | State     | Mode     | Status |  |
| cap_id     | cap_priority | cap_size | cap_state | cap_mode | status |  |

- `cap_id`  
is the CAP identifier.
- `cap_priority`  
is the CAP priority.
- `cap_size`  
is the number of cells in the CAP.
- `cap_state`  
is one of the following CAP states:
  - `online`  
The CAP is online.
  - `offline`  
The CAP is offline.
  - `offline-pending`  
The CAP will process current and pending requests, then go offline. The CAP will reject any new requests.

- `diagnostic`  
The CAP will process only current and pending requests, and will reject any new requests. The CAP is unavailable to client applications, and can only be controlled using `cmd_proc`. Use the `vary` command to vary the CAP online.
- `recovery`  
The CAP is initializing or recovering from errors. Wait for the CAP to go online.
- `cap_mode`  
is one of the following CAP entry modes:
  - `manual`  
You must unlock the CAP before entering cartridges.
  - `automatic`  
The CAP is ready for cartridge entry.
- `status`  
is one of the following CAP statuses:
  - `available`  
The CAP is available.
  - `enter`  
The CAP is unavailable (reserved for entering cartridges).
  - `eject`  
The CAP is unavailable (reserved for ejecting cartridges).
  - `audit`  
The CAP is unavailable (reserved for audit processing).

## Examples

- To query CAP 0,1,0:  
`query cap 0,1,0`
- To query all CAPs in the library:  
`query cap all`

See also:

| For information about...                  | See...                           |
|-------------------------------------------|----------------------------------|
| Displaying request status                 | <a href="#">query request</a>    |
| Setting CAP mode (manual or automatic)    | <a href="#">set cap mode</a>     |
| Setting CAP selection priority            | <a href="#">set cap priority</a> |
| Changing the state of a library component | <a href="#">vary</a>             |

## query clean

The `query clean` command displays cleaning cartridge status. Absent or ejected cartridges are not displayed. Cleaning cartridges that were reported as spent (used-up) by tape drives are not reported.

### Format

```
query clean vol_id... | all
```

### Options

- `vol_id | all`  
specifies the cleaning cartridge to query or all for all cartridges.

### Usage

Use the `query clean` command to display the status of a cleaning cartridge in the following format:

```
yyy-mm-ddhh:mm:ssDrive Status  
Identifier State Status volumeType  
  
drive_id state status vol_idtype
```

Where:

- `vol_id`  
is the cartridge identifier of the cleaning cartridge.
- `cell_id`  
is the location of the cleaning cartridge.
- `max_usage`  
is the number of times the cleaning cartridge can be used.
- `current_usage`  
is the number of times the cleaning cartridge has been used.
- `status`  
is the location of the cleaning cartridge:
  - `home`  
The cartridge is in a storage cell.
  - `in drive`  
The cartridge is in a drive.
  - `in transit`  
The cartridge is being moved.
- `type`

is the cartridge media type (for example, 3480, DD3D, DLTIII, or STK1R).

## Examples

- To display status information for cleaning cartridge J35992:

```
query clean J35992
```

- To query all cleaning cartridges:

```
query clean all
```

See also:

| For information about...                                     | See...                                        |
|--------------------------------------------------------------|-----------------------------------------------|
| Guidelines and procedures for cleaning drives                | <a href="#">Populating the LSM</a>            |
| Setting cleaning cartridge attributes                        | <a href="#">set clean</a>                     |
| Displaying cleaning cartridges and spent cleaning cartridges | <a href="#">Using display Command Options</a> |

## query drive

The `query drive` command displays drive status.

## Format

```
query drive drive_id... | all
```

## Options

- `drive_id | all`  
specifies the drive to query or all for all drives.

## Usage

Use the `query drive` command to display the status of a drive in the following format:

Where:

- `drive_id`  
is the drive identifier.
- `state`  
is one of the following:
  - `online`  
The drive is online.
  - `offline`  
The drive is offline.
  - `diagnostic`

The drive will process only current and pending requests and will reject any new requests. The drive is unavailable to client applications, and can only be controlled using `cmd_proc`. Use the `vary` command to vary the drive online.

- `recovery`

The drive is initializing or recovering from errors. Wait for the drive to go online.

- `status`

is one of the following drive statuses:

- `In use`

The drive has a cartridge mounted or is reserved for a mount.

Possible Scenario: You perform a `query drive all` and get a message saying the drive is in use. You then perform a `display drive *` and get a message saying the drive is reserved. This means: the reserved status for the drive implies that a `mount` request is in progress, and that a cartridge is on its way to the drive. At the same time, the drive is considered to be in use.

- `Available`

The drive is available for a mount.

- `vol_id`

is the identifier of the cartridge in the drive. This field is blank if there is no cartridge in the drive or if the cartridge's external label is unreadable or unknown.

- `drive_type`

is the drive type.

## Examples

- To query drive 0,3,1,0:

```
query drive 0,3,1,0
```

- To query all drives:

```
query drive all
```

See also:

| For information about...                  | See...                                        |
|-------------------------------------------|-----------------------------------------------|
| Changing the state of a library component | <a href="#">vary</a>                          |
| Dismounting a cartridge from a drive      | <a href="#">dismount</a>                      |
| Drive serial numbers                      | <a href="#">Using display Command Options</a> |

## query lmu

The `query lmu` command displays LMU and port status for both single-LMU and Dual-LMU ACS configurations and the desired state for ACSs and ports. If a library is partitioned, it also displays the partition ID.

The `query lmu` command is the best way to monitor ACSLS communication with the libraries it manages. `query lmu:`

- Shows the status of ACSLS connections to the library.
- Shows Redundant Electronics (RE) if any library in a string reports RE.
- Shows single or Dual-LMU for non- RE libraries or a non- SL8500.
- Displays the desired state for ACSs, port connections, LSMs, and tape drives.
- Displays the partition ID of a partitioned library.

## Format

```
query lmu acs_id... | all
```

## Options

- `acs_id | all`  
specifies the ACS whose LMUs you want to query or select `all` to query LMUs for all ACSs.

## Usage

Use the `query lmu` command to display LMU and port status and desired state for both single-LMU and Dual-LMU ACS configurations. The following examples show output for a library without Redundant Electronics and a library with Redundant Electronics.

Output Example without Redundant Electronics

```
ACSSA> q lmu all
2010-04-02 14:43:54          LMU Status
ACS: 0      Mode: Single LMU   Active Status: Communicating
Not Partitioned             Standby Status: -
      ACS State   Desired State
      online      online
Port  Port State   Desired State Role      CL
0,0   online      online      -        21 springtime:9997
ACS: 1      Mode: Dual LMU    Active Status: Communicating
Not Partitioned             standby Status: Communicating
      ACS State   Desired State
      online      online
Port  Port State   Desired State Role      CL
1,0   online      online      Active(A) 13 springtime:51100
1,1   online      online      standby(B) 13 springtime:51101
ACSSA>
```

Output Example of library with Redundant Electronics

```
ACSSA> q lmu all
2010-05-03 11:03:11          LMU Status
ACS: 0      Mode: Redundant   Active Status: Communicating
Not Partitioned             Standby Status: Communicating
      ACS State   Desired State
      Online      Online
Port  Port State   Desired State Role      CL
0,0   online      online      Standby(A) 21 10.80.92.43
0,1   online      online      Standby(A) 21 10.80.93.33
0,2   online      online      Active(B)  21 10.80.92.52
0,3   online      online      Active(B)  21 10.80.93.47
0,4   online      online      Standby(B) 21 10.80.92.44
0,5   online      online      Active(A)  21 10.80.92.53
```



**Where:**

- `acs_id`  
is the ACS identifier.
- `mode`  
is the LMU mode (Dual LMU, Single LMU, or SCSI LMU).
- `status`  
is the active or standby LMU status (Communicating, Not Communicating, or Offline).
- `partition_status`  
shows if the library is partitioned. If it is partitioned, the library shows the partition to which you are connected. The partition status is:
  - Not Partitioned
  - Partition 1-n  
The partition number.
- `acs_state`  
is the actual ACS state. States are:
  - `online`  
The ACS is online.
  - `diagnostic`  
The ACS will process only current and pending requests, and will reject any new requests. The ACS is unavailable to client applications, and can only be controlled using `cmd_proc`. Use the `vary` command to vary the ACS online.
  - `offline`  
The ACS is offline.
  - `offline pending`  
The ACS will process current and pending requests, then go offline. The ACS will reject any new requests.
- `acs_desired_state`  
you want the ACS to be in this state. The desired states are:
  - `online`
  - `diagnostic`
  - `offline`
- `port_id`  
is the port identifier.
- `port_state`  
is one of the following actual port states:
  - `online`  
The port is online.

- offline  
The port is offline.
- `port_desired_state`  
you want to port to be in this state. The desired states are:
  - online
  - offline
- `role (des)`  
is the LMUs' role and designation (A or B), where roles are:
  - Active  
LMU is in active role (the LMU is managing the ACS).
  - Standby  
LMU is in standby `role` (not managing the ACS, communicating with the active LMU and available for switch over).

 **Note:**

During switch over, the `role` field information is not current, and may display as a single dash (-). As the information becomes current, ACSLS refreshes the `role` field with the actual role of each LMU.

- `compat_level`  
is the host/LMU microcode compatibility level. Level 11 or above is required for Dual-LMU configurations.
- `dev_name`  
is the port device name.

## Examples

- To display LMU and port status for all LMUs managing all ACSs:  

```
query lmu all
```
- To display LMU and port status for all LMUs managing ACSs 0 and 1:  

```
query lmu 0 1
```

**Table 12-5** `query lmu` references

| For information about:                                                         | See:                       |
|--------------------------------------------------------------------------------|----------------------------|
| Manually switching ACS management from the ACS's active LMU to the standby LMU | <a href="#">switch lmu</a> |

## query lock

The `query lock` command displays the lock status of a drive or cartridge.

## Format

```
query lock type identifier... | all
```

## Options

- `type identifier | all`  
specifies the drive or cartridge to query or all for all drives or cartridges as shown in the following table.

**Table 12-6 Valid Lock Types for query lock**

| Library Component | type   | identifier |
|-------------------|--------|------------|
| drive             | drive  | drive_id   |
| volume            | volume | vol_id     |

## Usage

Use the `query lock` command to display the lock status of a drive or cartridge in the following format:

```

yyy-mm-ddhh:mm:ssLock          Status
Identifier  Lock-id  Duration  Pending  StatusUser Identifier

vol_id      lock_id  duration  pending  status  user_id
or
drive_id    lock_id  duration  pending  status  user_id

```

Where:

- `vol_id`  
is the identifier of the specified cartridge.
- `drive_id`  
is the identifier of the specified drive.
- `lock_id`  
is the lock ID.
- `duration`  
is the amount of time, in seconds, that the lock has been active.
- `pending`  
is the number of lock requests that are waiting for cartridge or drive.
- `status` is one of the following statuses:
  - `available`  
The cartridge or drive is available.
  - `in use`  
The cartridge or drive is in use, or is reserved for a mount.

- `user_id`  
is the user ID that has locked the cartridge or drive. The `user_id` wraps after 80 characters.

## Examples

- To display lock status information for drive 1,0,4,0:  

```
q loc dr 1,0,4,0
```
- To display lock status information for all drives:  

```
query lock drive all
```
- To display lock status information for cartridge SL4493:  

```
query lock cartridge SL4493
```
- To display lock status information for all cartridges:  

```
query lock cartridge all
```

**Table 12-7** query lock references

| For information about                                                  | See:                       |
|------------------------------------------------------------------------|----------------------------|
| Removing all active or pending locks on a specified drive or cartridge | <a href="#">clear lock</a> |
| Locking drives and cartridges                                          | <a href="#">lock</a>       |
| Setting your lock ID                                                   | <a href="#">set lock</a>   |
| Displaying your lock or user ID                                        | <a href="#">show</a>       |
| Removing active locks                                                  | <a href="#">unlock</a>     |

## query lsm

The `query lsm` command displays LSM status.

## Format

```
query lsm lsm_id... | all
```

## Options

- `lsm_id | all`  
specifies the LSM to query or all, for all locks.  
Use the `query lsm` command to display the status of an LSM in the following format:

```
yyy-mm-ddhh:mm:ssLSM Status
Identifier State Free Cell Audit Mount Dismount Enter Eject
Count C/P C/P C/P C/P C/P
```

Where:

- `lsm_id`  
is the LSM identifier.

- `state` is one of the following LSM states:
  - `diagnostic`  
The LSM will process only current and pending requests and will reject any new requests. The LSM is unavailable to client applications and can only be controlled using `cmd_proc`. Use the `vary` command to vary the LSM online.
  - `offline`  
The LSM is offline.
  - `offline pending`  
The LSM will process current and pending requests, then go offline. The LSM will reject any new requests.
  - `online`  
The LSM is online.
  - `recovery`  
The LSM is initializing or recovering from errors. Wait for the LSM to go online.
- `count`  
is the number of free storage cells in the LSM.
- `n`  
is the number of current (C) and pending (P) requests for the LSM for each command that requires library resources (`audit`, `mount`, `dismount`, `enter`, and `eject`).

## Examples

- To display status information for LSM 1 of ACS 0:  
`query lsm 0,1`
- To display status information for all LSMs:  
`query lsm all`

See also:

| For information about...                  | See...                           |
|-------------------------------------------|----------------------------------|
| Displaying request status                 | <a href="#">"query request "</a> |
| Changing the state of a library component | <a href="#">"vary "</a>          |

## query mount

Use the `query mount` command to optimize the performance of a library with multiple LSMs, such as an SL8500 or a string of connected SL8500s. It displays the status of media-compatible drives for a specified data cartridge. These drives are not displayed if a cartridge(s) is absent or rejected. In addition:

- When selecting the list of drives to be returned to a `query mount` request, the drives must be compatible with the volume specified.
- Drives are ordered primarily by pass-thru distance. The drives in LSMs closest to the cartridge are listed first.

- With ACSLS 7.3 and later releases, drives that are the same pass-thru distance from the cartridge are in least-recently-used order.

Example: The compatible drive with the longest time since a cartridge was dismounted, is first; the drive with the next longest time, is second, and so on.

## Format

```
query mount vol_id
```

## Options

- `vol_id`  
specifies the cartridge to query.

## Usage

Use the `query mount` command to display the status of all library drives attached to the same ACS as the cartridge, and compatible with the media type of the specified cartridge. `query mount` displays drive status for LSMs in any state (online, offline, offline pending, or diagnostic).

The compatible drives are ordered by proximity to the specified cartridge in the following format:

```

yyy-mm-dd hh:mm:ss Mount Status
Identifier  Status  Drive   State  Status      Volume  Drive Type
vol_id     vol_stat drive_id state  drive_stat  inu_id  drive_type

```

Where:

- `vol_id`  
is the identifier of the specified cartridge.
- `vol_stat`  
is the location of the cartridge:
  - `home`  
The cartridge is in a storage cell.
  - `in drive`  
The cartridge is in a drive.
  - `in transit`  
The cartridge is being moved or is missing.
- `drive_id`  
is a list of all library drives compatible with the media type of the specified cartridge.
- `drive_id`  
is the drive identifier.

- `state`  
is one of the following drive states:
  - `online`  
The drive is online.
  - `offline`  
The drive is offline.
  - `diagnostic`  
The drive will process only current and pending requests, and will reject any new requests. The drive is unavailable to client applications, and can only be controlled using `cmd_proc`. Use the `vary` command to vary the drive online.
  - `recovery`  
The drive is initializing or recovering from errors. Wait for the drive to go online.
- `status`  
is one of the following drive statuses:
  - `In use`  
The drive has a cartridge mounted or is reserved for a mount.
  - `Available`  
The drive is available for a mount.
- `inu_id`  
is the identifier of the cartridge in the drive. The cartridge id appears only if the `drive_stat` is `In use`.
- `drive_type`  
is the drive type.

## Examples

- To display status information of drives ordered by proximity to cartridge ZUNI14:  

```
query mount ZUNI14
```

See also:

| For information about...                              | See...                       |
|-------------------------------------------------------|------------------------------|
| Mounting a data cartridge on a drive                  | <a href="#">mount</a>        |
| Displaying the lock status of a drive or cartridge    | <a href="#">query lock</a>   |
| Displaying the location and media type of a cartridge | <a href="#">query volume</a> |
| Changing the state of a library component             | <a href="#">vary</a>         |

## query mount \*

The `query mount *` command displays the status of drives that are compatible with the media in one or more specified scratch pools (and, optionally, only the drives compatible with a specific cartridge media type within the pool).

## Format

```
query mount * pool_id... [media media_type | media *]
```

## Options

- `pool_id`  
specifies the scratch pool or pools to query.
- `media media_type | media *`  
specifies the media type.

## Usage

Use the `query mount *` command to display the status of all library drives compatible with all cartridge media types in a specified scratch pool(s), and in the same ACS as the cartridges. Pool 0 is the common scratch pool. Specify the `media_type` option to restrict the display, to drives compatible with a specific cartridge media type within the pool. The displayed drives are ordered by proximity to the densest scratch pools.

`query mount *` displays drive status for drives in LSMs in any state (online, offline, offline-pending, or diagnostic).

The `query mount *` command displays drive status in the following format:

```
yyy-mm-dd hh:mm:ss Mount Scratch Status
Identifier Drive State Volume Status Drive Type
pool_id drive_id state vol_id drive_stat drive_type
```

Where:

- `pool_id`  
is the specified scratch pool(s).
- `drive_id`  
is a list of all library drives compatible with all media types of the specified pool(s), or a specific media type within the pool if a media type was specified.
- `state`  
is one of the following drive states:
  - `online`  
The drive is online.
  - `offline`  
The drive is offline.
  - `diagnostic`  
The drive will process only current and pending requests, and will reject any new requests. The drive is unavailable to client applications, and can only be controlled using `cmd_proc`. Use the `vary` command to vary the drive online.
  - `recovery`



The drive is initializing or recovering from errors. Wait for the drive to go online.

- `vol_id`  
is the identifier of the cartridge in the drive. The cartridge id appears only if the `drive_stat` is in use.
- `drive_stat`  
is one of the following drive statuses:
  - In use  
The drive has a cartridge mounted or is reserved for a mount.
  - Available  
The drive is available for a mount.
- `drive_type`  
is the drive type.

## Examples

To display the status of compatible drives listed by proximity, to the largest concentration of scratch tapes in pool 5:

```
query mount * 5
```

See also:

| For information about...                           | See...                        |
|----------------------------------------------------|-------------------------------|
| Creating or modifying scratch pools                | <a href="#">define pool</a>   |
| Deleting empty scratch pools                       | <a href="#">delete pool</a>   |
| Mounting a scratch cartridge on a drive            | <a href="#">mount *</a>       |
| Displaying the lock status of a drive or cartridge | <a href="#">query lock</a>    |
| Displaying scratch pool attributes                 | <a href="#">query pool</a>    |
| Displaying scratch cartridge status                | <a href="#">query scratch</a> |
| Setting or clearing cartridge scratch attributes   | <a href="#">set scratch</a>   |
| Changing the state of a library component          | <a href="#">vary</a>          |

## query pool

The `query pool` command displays scratch pool attributes.

### Format

```
query pool pool_id... | all
```

### Options

- `pool_id | all`  
specifies the scratch pool to query or all, for all pools. Pool 0 is the common pool.

## Usage

Use the `query pool` command to display scratch pool attributes in the following format:

```
yyy-mm-dd hh:mm:ss Pool Status
Identifier Volume Count Low Water Mark High Water Mark Attributes

pool_id    vol_count    low_water_mark high_water_mark attribute
```

Where:

- `pool_id`  
is the specified scratch pool.
- `vol_count`  
is the number of scratch cartridges in the pool.  
Cartridges not counted are absent and ejected scratch cartridges and data cartridges in a pool.
- `low_water_mark`  
is the low cartridge warning threshold. If the scratch cartridge count falls below this threshold, ACSLS logs a warning message in the Event Log.  
A "-" after the value means that the scratch cartridge count is below the low cartridge threshold.
- `high_water_mark`  
is the high cartridge warning threshold. If the scratch cartridge count reaches or exceeds this threshold, ACSLS logs a warning message in the Event Log. A "+" after the value means that the scratch cartridge count is at or above the high cartridge threshold.
- `attribute`  
is displayed if `overflow` is set (using the `set scratch` command) for the specified scratch pool. Overflow specifies that scratch cartridges are selected from the common scratch pool (Pool 0) if `mount scratch *` requests cannot be satisfied with a cartridge from the specified scratch pool.

## Examples

- To display status information for scratch pool 5:  

```
query pool 5
```
- To display status information for all scratch pools:  

```
query pool all
```

See also:

| For information about...            | See...                        |
|-------------------------------------|-------------------------------|
| Creating or modifying scratch pools | <a href="#">define pool</a>   |
| Deleting empty scratch pools        | <a href="#">delete pool</a>   |
| Displaying scratch cartridge status | <a href="#">query scratch</a> |

---

| For information about...                         | See...                      |
|--------------------------------------------------|-----------------------------|
| Setting or clearing cartridge scratch attributes | <a href="#">set scratch</a> |

---

## query port

The `query port` command displays port status.

### Format

```
query port port_id... | all
```

### Options

- `port_id | all`  
specifies the port to query or all, for all ports.

### Usage

Use the `query port` command to display the status of a port in the following format:

```
yyy-mm-dd hh:mm:ss Port Status  
State Identifier
```

```
state port_id
```

Where:

- `state`  
is one of the following port states:
  - `online`  
The port is online.
  - `offline`  
The port is offline.
- `port_id`  
is the port identifier.

### Examples

- To display status information for port 0,0:  

```
query port 0,0
```
- To query all ports:  

```
query port all
```

See also:

---

| For information about...                  | See...               |
|-------------------------------------------|----------------------|
| Changing the state of a library component | <a href="#">vary</a> |

---

## query request

The `query request` command displays request status.

### Format

```
query request request_id... | all
```

### Options

- `request_id | all`  
specifies the request to query or all, for all requests.

### Usage

Use the `query request` command to display the status of a request in the following format:

Where:

- `request_id`  
is the ACSLS request identifier.
- `command`  
is the ACSLS command that corresponds to the request identifier.
- `status`  
is one of the following request statuses:
  - `Current`  
ACSLS is processing the request.
  - `Pending`  
The request is waiting to be processed.
  - `Not found`  
The specified request is not a valid ACSLS request.

### Examples

- To display status information for request 33179:  

```
query request 33179
```
- To display all current and pending requests:  

```
query request all
```

See also:

---

| For information about... | See...                 |
|--------------------------|------------------------|
| Cancelling a command     | <a href="#">cancel</a> |

---

## query scratch

The `query scratch` command displays the status of scratch cartridges in a pool sorted by ascending order of access date. The cartridges with the earliest access date are at the top of the list, with the most recently used cartridges at the bottom of the list. Pool IDs in the first column do not appear in any particular order. Only those cartridges permitted through Access Control are displayed.

### Format

```
query scratch pool_id... | all
```

### Options

```
pool_id | all
```

specifies the scratch pool to query or all, for all pools. Pool 0 is the common pool.

### Usage

Use the `query scratch` command to display the status of scratch cartridges in a pool in the following format:

Where:

- `pool_id`  
is the specified scratch pool.
- `vol_id`  
is the identifier of the scratch cartridge.
- `cell_id`  
is the storage cell that contains the cartridge.
- `status`  
is the location of the cartridge:
  - `home`  
The cartridge is in a storage cell.
  - `in drive`  
The cartridge is in a drive.
  - `in transit`  
The cartridge is being moved.
- `media_type`  
is the cartridges' media type.

 **Note:**

Scratch cartridges with a cartridge status of absent, ejected, or missing are not included in the `query scratch` output.

To see cartridges sorted by pool, you issue successive `query scratch` commands for each pool. Alternately, you can issue the `display volume` command to see the information and sort it by pool, but you will not be guaranteed of getting the least recently used scratch cartridge.

## Examples

- To display status information for scratch cartridges in scratch pool 29015:

```
query scratch 29015
```

- To display status information for scratch cartridges in all scratch pools:

```
query scratch all
```

See also:

| For information about...                         | See...                      |
|--------------------------------------------------|-----------------------------|
| Creating or modifying scratch pools              | <a href="#">define pool</a> |
| Deleting empty scratch pools                     | <a href="#">delete pool</a> |
| Setting or clearing cartridge scratch attributes | <a href="#">set scratch</a> |

## query server

The `query server` command displays ACSLS and library status.

### Format

```
query server
```

### Options

None.

### Usage

Use the `query server` command to display the status of ACSLS and the library in the following format:

Where:

- Identifier  
is blank
- state  
is one of the following ACSLS states:

- `idle`  
ACSLs is idle (not processing requests).
  - `idle pending`  
ACSLs will process current and pending requests, reject new requests, then go idle.
  - `recovery`  
ACSLs is initializing (going to run state), or recovering from errors; ACSLS is not processing requests.
  - `run`  
ACSLs is running (processing requests).
- `count`  
is the number of free storage cells in the library.
  - `n`  
is the number of current (C) and pending (P) ACSLS requests for each command that requires library resources (`audit`, `mount`, `dismount`, `enter`, and `eject`).

**HINT:** You cannot cancel a `query server` request.

## Examples

- To display status information on the server:

```
query server
```

See also:

| For information about...                  | See...                        |
|-------------------------------------------|-------------------------------|
| Changing the state of a library component | <a href="#">vary</a>          |
| Displaying request status                 | <a href="#">query request</a> |

## query volume

The `query volume` command displays the location of a cartridge.

### Format

```
query volume vol_id... | all
```

### Options

- `vol_id | all`  
specifies the volume to query or `all`, for all volumes.

### Usage

Use the `query volume` command to display the location of a volume in the following format:

Where:

- `vol_id`  
is the volume identifier.
- `status`  
is the location of the cartridge.
  - `home`  
The cartridge is in a storage cell.
  - `in drive`  
The cartridge is in a drive.
  - `in transit`  
The cartridge is being moved or is missing.
  - `absent`  
The cartridge cannot be found.
  - `ejected`  
The cartridge has been ejected from the library.
- `location`  
specifies the location as one of the following:
  - If the status is `home`, the location is a storage cell identifier.
  - If the status is `in transit`, the location is either a cell identifier or a drive identifier.
  - If the status is `in drive`, the location is a drive identifier.
- `media_type`  
is the volume's media type.

## Examples

- To display status information for volume 2903B:

```
query volume 2903B
```

- To display all volumes in the library:

```
q volume all
```

See also:

| For information about...                                                     | See...                                                      |
|------------------------------------------------------------------------------|-------------------------------------------------------------|
| Other volume information                                                     | <a href="#">Creating a Logging Volume Statistics Report</a> |
| Display Command                                                              | <a href="#">Using display Command Options</a>               |
| Mounting a data volume on a drive                                            | <a href="#">mount</a>                                       |
| Displaying the lock status of a drive or volume                              | <a href="#">query lock</a>                                  |
| Displaying the status of media-compatible drives for a specified data volume | <a href="#">query mount</a>                                 |
| Displaying scratch pool attributes                                           | <a href="#">query pool</a>                                  |



| For information about...                      | See...                      |
|-----------------------------------------------|-----------------------------|
| Setting or clearing volume scratch attributes | <a href="#">set scratch</a> |
| Changing the state of a library component     | <a href="#">vary</a>        |

## set commands

The `set` commands set various attributes of different library components. See the following sections for more information about each `set` command, including format, options, usage, and messages.

### Format

The following shows the general format of the `set` commands:

```
set type [off | subtype] [*] identifier...
```

See also:

| For information about... | See...                    |
|--------------------------|---------------------------|
| Cancelling a command     | <a href="#">"cancel "</a> |

## Command Area Messages

This section discusses command area messages.

### Success Messages:

The following message appears when a `set` command succeeds:

```
Set: Set completed, Success.
```

See each `set` command for specific success messages.

### Intermediate Messages:

None.

### Error Messages:

See each `set` command for specific messages.

## Display Area Messages

See each `set` command for specific messages.

## set cap mode

The `set cap mode` command sets the entry mode for a CAP.

### Format

```
set cap mode cap_mode cap_id
```

## Options

- `cap_mode`  
specifies manual or automatic CAP mode.
- `cap_id`  
specifies the CAP identifier.  
You cannot specify a CAP identifier that contains an asterisk (\*) unless the priority is set.

## Usage

Use the `set cap mode` command to set a CAP's entry mode to one of the following modes:

- `manual`  
You must enter an `enter` command before entering cartridges.
- `automatic`  
You can enter cartridges without first entering an `enter` command.

 **Note:**

You cannot set the CAP mode to automatic in a partitioned library.

**HINT:** You cannot change a CAP's mode while the CAP is in use. That is, if the door is open during either manual or automatic enter operations, you cannot change its mode until you complete the `enter` operation.

## Examples

- To set CAP 0,3,1 in manual mode:  

```
set cap mode manual 0,3,1
```
- To set CAP 0,3,1 in automatic mode:  

```
set cap mode automatic 0,3,1
```

See also:

| For information about...                                                       | See...                    |
|--------------------------------------------------------------------------------|---------------------------|
| Making a CAP (manual mode) ready to enter labelled cartridges into the library | <a href="#">enter</a>     |
| Displaying CAP status                                                          | <a href="#">query cap</a> |
| Changing the state of a library component                                      | <a href="#">vary</a>      |

## Command Area Messages

This section discusses the command messages.

**Success Messages:**

- Set: CAP `cap_id`, mode changed to `cap_mode`.  
Explanation: ACSLS changed the mode of the specified CAP.  
Variable:
  - `cap_id` is the CAP whose mode is changed.
  - `cap_mode` is the new enter mode of the CAP.

**Intermediate Messages:****Error Messages:**

- Set: CAP `cap_id` Set failed, Incorrect attribute.  
Explanation: ACSLS cannot change the mode of the specified CAP because you specified an invalid CAP mode.  
Variable: `cap_id` is the CAP whose mode is not changed.
- CAP `cap_id`: Automatic mode.  
Explanation: ACSLS changed the specified CAP's mode to automatic.  
Variable: `cap_id` is the CAP whose mode is changed.
- CAP `cap_id`: Manual mode.  
Explanation: ACSLS changed the specified CAP's mode to manual.  
Variable: `cap_id` is the CAP whose mode is changed.

## set cap priority

The `set cap priority` command sets the automatic selection priority for a CAP.

### Format

```
set cap priority cap_priority cap_id
```

### Options

- `cap_priority`  
specifies the CAP priority. Valid values are 0 to 16, where 16 is the highest priority. All CAPs initially have a 0 priority, which means that ACSLS does not automatically select the CAP.

 **Note:**

If you are using the AEM option. Since the AEM should not be used for small enters and ejects, the maximum CAP priority is 1. This aids in preventing the AEM from being selected when the CAP ID is wild-carded with an asterisk on an `audit`, `enter`, or `eject`. For more information on the AEM, refer to "[Using the AEM](#)".

- `cap_id`

specifies the CAP identifier. You must specify a particular CAP; you cannot specify an asterisk (\*) to set the same priority for all CAPs.

## Usage

Use the `set cap priority` command to set a CAP's automatic selection priority.

If a CAP request specifies an asterisk (\*) for the CAP ID, ACSLS automatically selects an available CAP with the highest nonzero priority for each ACS specified in the request.

## Examples

- To assign priority 16 to CAP 0,3,1:

```
set cap priority 16 0,3,1
```

See also:

| For information about...                                                        | See...                    |
|---------------------------------------------------------------------------------|---------------------------|
| Updating the ACSLS database to match the actual inventory of library cartridges | <a href="#">audit</a>     |
| Ejecting cartridges from the library                                            | <a href="#">eject</a>     |
| Making a CAP (manual mode) ready to enter labelled cartridges into the library  | <a href="#">enter</a>     |
| Displaying CAP status                                                           | <a href="#">query cap</a> |
| Changing the state of a library component                                       | <a href="#">vary</a>      |

## Command Area Messages

Updates the ACSLS database to match the actual inventory of library cartridges.

### Success Messages:

- Set: CAP `cap_id`, priority changed to `cap_priority`.  
Explanation: ACSLS changed the priority of the specified CAP.

Variable:

- `cap_id` is the CAP whose priority is changed.
- `cap_priority` is the new CAP priority.

### Intermediate Messages:

None.

### Error Messages:

- Set: CAP `cap_id` Set failed, Incorrect attribute.  
Explanation: ACSLS cannot change the mode of the specified CAP because you specified an invalid CAP priority.  
Variable: `cap_id` is the CAP whose priority is not changed.

## Display Area Messages

None.

## set clean

The `set clean` command sets cleaning cartridge attributes.

For all modern cleaning cartridges, cleaning cartridge attributes are automatically set when the cleaning cartridges are added by `audit`, `enter`, or `Cartridge Recovery`. This includes setting the cleaning cartridges `max_usage`.

## Format

```
set clean max_usage | off vol_id | volrange
```

## Options

- `max_usage | off`

specifies the number of times a cleaning cartridge is used before ACSLS stops selecting the cartridge to clean drives. `off` specifies that ACSLS will not select the cartridge and redefines the cartridge as a data cartridge.

 **Note:**

You cannot: specify a maximum cleaning usage for a media type that is reserved for only data cartridges; set cleaning `off` for a media type that is only a cleaning cartridge

- `vol_id | volrange`

specifies the cleaning cartridge or range of cartridges.

## Usage

Use the `set clean` command to set the number of times ACSLS will select a cleaning cartridge. You also use `set clean` to set a cartridge's cleaning cartridge attribute `off`, which you do if you have incorrectly defined a data cartridge as a cleaning cartridge.

## Examples

- To set the maximum use to 10 for cleaning cartridges `CLN108 - CLN112`:  

```
set clean 10 CLN108-CLN112
```
- To set the cleaning cartridge attribute `off` and redefine cartridges `HRR234 - HRR244` as data cartridges:  

```
set clean off HRR234-HRR244
```

See also:

---

| For information about...                      | See...                             |
|-----------------------------------------------|------------------------------------|
| Guidelines and procedures for cleaning drives | <a href="#">Populating the LSM</a> |
| Displaying cleaning cartridge status          | <a href="#">query clean</a>        |

---

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- Set: volume `vol_id` is a cleaning cartridge.  
Explanation: ACSLS specified that the cleaning cartridge is valid for drive cleaning.  
Variable: The `vol_id` is the valid cleaning cartridge.
- Set: volume `vol_id` is not a cleaning cartridge.  
Explanation: ACSLS specified that the cleaning cartridge is not valid for drive cleaning.  
Variable: The `vol_id` is the invalid cleaning cartridge.

### Intermediate Messages:

None.

### Error Messages:

- Set: Clean `vol_id` Set failed, Incorrect attribute.  
Explanation: ACSLS cannot change the cleaning attribute because the specified cartridge is not a cleaning cartridge.  
Variable: The `vol_id` is the data or scratch cartridge.

## Display Area Messages

None.

## set lock

The `set lock` command sets your lock ID.

## Format

```
set lock lock_id
```

## Options

- `lock_id`  
specifies the lock ID. Valid lock IDs are 0 to 32767.

## Usage

Use the `set lock` command to set or change your lock ID. Change your current lock ID when you want to remove locks on a drive or cartridge whose lock ID does not match your current lock ID, as follows:

- To clear all locks on a specified drive or cartridge, set your lock ID to 0, then enter a `clear lock` command.
- To remove active locks on drives or cartridges, set your lock ID to the lock ID of the locked components, then enter an `unlock` command.

### Note:

When you enter a `lock` command to lock a cartridge or drive and your lock ID is zero (0), ACSLS assigns a lock ID to the cartridge or drive, and then changes your lock ID to the lock ID of the cartridge or drive. You cannot use the `set lock` command to set your lock ID and then use the `lock` command to lock a cartridge or drive with the lock ID that you set with `set lock`.

## Examples

- To change your current lock ID lock to new lock ID 354:

```
set lock 354
```

See also:

| For information about...                                               | See...                     |
|------------------------------------------------------------------------|----------------------------|
| Removing all active or pending locks on a specified drive or cartridge | <a href="#">clear lock</a> |
| Locking drives and cartridges                                          | <a href="#">lock</a>       |
| Displaying the lock status of a drive or cartridge                     | <a href="#">query lock</a> |
| Displaying your lock or user ID                                        | <a href="#">show</a>       |
| Removing active locks                                                  | <a href="#">unlock</a>     |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- Set: Changed lock identifier from `previous_lock_id` to `lock_id`.

Explanation: ACSLS changed your lock ID.

Variable:

- The `previous_lock_id` was your previous lock ID.
- The `lock_id` is your new lock ID.

### Intermediate Messages:

None.

**Error Messages:**

None.

## Display Area Messages

None.

## set owner

The `set owner` command sets volume ownership.

## Format

```
set owner owner_id volume vol_id | volrange
```

## Options

- `owner_id`  
specifies the owner identifier. You must enclose this value in quotes (" ").
- `volume vol_id | volrange`  
specifies the volume or range of volumes.

## Usage

Use the `set owner` command to set volume ownership. Enter `set owner` from `cmd_proc`; you cannot set volume ownership from a client application.

## Examples

- To assign ownership of volume YUMA06:  

```
set owner "cray" volume YUMA06
```

  
To remove volume ownership of volume YUMA06:  

```
set owner:"" volume 0YUMA06
```

## Command Area Messages

None.

## Display Area Messages

None.

## set scratch

The `set scratch` command sets or clears a scratch attribute for a volume and assigns the volume to a pool.



You can also use the `watch_vols` utility to automatically set scratch cartridge attributes automatically when scratch cartridges are added or re-activated by `audit`, `enter`, or Cartridge Recovery.

## Format

```
set scratch [off] pool_id vol_id | volrange
```

## Options

- `off`  
specifies that the volume is a data cartridge.
- `pool_id`  
specifies the cartridges' scratch pool. Specify an asterisk (\*) to reassign a data cartridge to its current pool.
- `vol_id | volrange`  
specifies the volume or range of volumes.

## Usage

Use the `set scratch` command to set or clear a volumes' scratch attribute, and assign the volume to a pool.

## Examples

- To define volumes YUMA10-YUMA20 as scratch volumes, and assign them to scratch pool 5:  

```
set scratch 5 YUMA10-YUMA20
```
- To move scratch volumes YUMA10-YUMA15 to pool 10:  

```
set scratch 10 YUMA10-YUMA15
```
- To "unscratch" (change from scratch to data) volumes YUMA16-YUMA20, and move them to the common pool (pool 0):  

```
set scratch off 0 YUMA16-YUMA20
```
- To scratch to data volumes YUMA16-YUMA20 and keep them in their current pool:  

```
set scratch * YUMA16-YUMA20
```

See also:

---

| For information about...            | See...                        |
|-------------------------------------|-------------------------------|
| Displaying scratch cartridge status | <a href="#">query scratch</a> |
| Displaying scratch pool attributes  | <a href="#">query pool</a>    |
| Creating or modifying scratch pools | <a href="#">define pool</a>   |
| Deleting empty scratch pools        | <a href="#">delete pool</a>   |

---

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- Set: volume `vol_id` in tape pool `pool_id` is a scratch cartridge.  
Explanation: ACSLS changed the specified data cartridge to a scratch cartridge and assigned it to a pool.  
Variable:
  - The `vol_id` is the specified volume.
  - The `pool_id` is the pool to which the volume is assigned.
- Set: volume `vol_id` in tape pool `pool_id` is a data volume.  
Explanation: ACSLS changed the specified scratch cartridge to a data cartridge.  
Variable: The `vol_id` is the specified volume.

### Intermediate Messages:

None.

### Error Messages:

- Set: Scratch `vol_id` Set failed, Incorrect attribute.  
Explanation: ACSLS cannot change the scratch attribute because the specified cartridge is a cleaning cartridge.  
Variable: The `vol_id` is the cleaning cartridge.

## Display Area Messages

- Pool `pool_id`: low water mark warning.  
Explanation: The number of cartridges in the specified scratch pool is less than or equal to the low threshold.  
Variable: The `low_water_mark` is the low threshold of the specified scratch pool.
- Pool `pool_id`: high water mark warning.  
Explanation: The number of cartridges in the specified scratch pool is greater than or equal to the high threshold.  
Variable: The `high_water_mark` is the high threshold of the specified scratch pool.

## show

The `show` command displays your lock ID or user ID.

## Format

`show type`

## Options

- `type`  
specifies one of the following types:
- `lock`  
Your lock ID.
- `user`  
Your user ID.

## Usage

Use the `show` command to display your lock ID or user ID.

## Examples

- To display requestor's `user_id`:  

```
show user
```
- To display current `lock_id`:  

```
show lock
```

See also:

| For information about...                                               | See...                     |
|------------------------------------------------------------------------|----------------------------|
| Removing all active or pending locks on a specified drive or cartridge | <a href="#">clear lock</a> |
| Locking drives and cartridges                                          | <a href="#">lock</a>       |
| Displaying the lock status of a drive or cartridge                     | <a href="#">query lock</a> |
| Setting your lock ID                                                   | <a href="#">set lock</a>   |
| Removing active locks                                                  | <a href="#">unlock</a>     |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- `Show: Current lock identifier is lock_id`  
Explanation: Your current lock ID is `lock_id`.
- `Show: User identifier is user_id`  
Explanation: Your current user ID is `user_id`.

### Intermediate Messages:

None.

### Error Messages:

None.

## Display Area Messages

None.

## start

The `start` command starts ACSLS request processing.

## Format

`start`

## Options

None.

## Usage

Use the `start` command to put ACSLS in run state and start ACSLS request processing. You typically use the `start` command to restart request processing if ACSLS is idled.

## Examples

- To restart ACSLS request processing:

```
start
```

If ACSLS is in run state, entering the `start` command has no effect.

See also:

| For information about...                     | See...                         |
|----------------------------------------------|--------------------------------|
| Stopping ACSLS from processing new requests  | <a href="#">idle</a>           |
| Displaying the status of a library component | <a href="#">query commands</a> |
| Changing the state of a library component    | <a href="#">vary</a>           |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- ACSLM Request Processing Started: Success.  
Explanation: ACSLS started processing requests.

### Intermediate Messages:

None.

### Error Messages:

- `ACSLM Request Processing Not Started: status`

Explanation: ACSLS did not start processing requests.

Variable: `status` is the reason for the failure. For more information on common status messages, see ACSLS Messages.

## Display Area Messages

- `Server system running`

Explanation: ACSLS started processing requests.

## switch lmu

In Redundant Electronics (RE) or Dual-LMU configurations, the `switch lmu` command manually switches ACS management from the active library controller (LC) to the standby LC.

- `switch lmu` only supports ACSs with a single RE library, either an SL3000 or a standalone SL8500.
- `switch lmu` does not support partitioned SL8500s or SL3000s.

## Format

```
switch lmu acs_id
```

## Options

- `acs_id`  
specifies the ACS to switch library management from the active to the standby LC or LMU.

## Usage

Use the `switch lmu` command to manually switch library management from the library's active LC or LMU to the standby LC or LMU. Before you enter a `switch lmu` command, make sure that:

- ACSLS is in run state
- The ACS you specify is in the online or diagnostic state
- At least one port is online to each LC or LMU

## Examples

In RE or Dual-LMU configurations, use the `switch lmu` command to switch ACS management from the library's active LC or LMU to the standby LC or LMU. Assume the following RE configuration:

- In a library, LC A is in the active role and LC B is in the standby role.  
If ACSLS loses communication to LC A, but can still communicate with LC B, use `switch lmu` to make LC B the active LC.

See also:

| For information about...                    | See...                       |
|---------------------------------------------|------------------------------|
| Displaying ACS status                       | <a href="#">query acs</a>    |
| Displaying ACS and port status with details | <a href="#">query lmu</a>    |
| Displaying port status                      | <a href="#">query port</a>   |
| Displaying ACSLS and library status         | <a href="#">query server</a> |
| Starting ACSLS request processing           | <a href="#">start</a>        |
| Changing the state of a library component   | <a href="#">vary</a>         |
| Redundant Electronics                       | <a href="#">Overview</a>     |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- `Switch: Switch lmu completed for acs_id, Success.`  
 Explanation: ACSLS switched library management from the active to the standby LMU.  
 Variable: The `acs_id` is the ACS whose LC or LMUs switched roles.
- `Switch: Switch lmu initiated for acs_id, Success.`  
 Explanation: ACSLS started switching library management from the active to the standby LC or LMU.  
 Variable: The `acs_id` is the ACS whose LC or LMUs are switching roles.
- `Switch: Switch lmu failed, acs_id not found.`  
 Explanation: You specified an invalid ACS.  
 Variable: The `acs_id` is the ACS identifier you specified on the `switch lmu` command.
- `Switch: Switch lmu failed, acs_id is offline.`  
 Explanation: You specified an ACS that is offline. Use the `vary` command to vary the ACS to online or diagnostic mode.  
 Variable: The `acs_id` is the ACS identifier you specified on the `switch lmu` command.
- `Switch: Switch lmu failed, acs_id not configured for dual lmu.`  
 Explanation: You specified an ACS that is not configured with active and standby LC or LMUs. Either:
  - Reconfigure the specified ACS, then re-enter the `switch lmu` command
  - Enter a `switch lmu` command that specifies a library that is configured with active and standby LCs or LMUs.
 Variable: The `acs_id` is the ACS identifier you specified on the `switch lmu` command.
- `Switch: Switch lmu failed, not communicating.`

Explanation: switch lmu switch failed because the standby LC or LMU is not communicating. Check the standby LC or LMU for communications problems.

- Switch: Switch lmu failed, switch already active.

Explanation: switch lmu is in progress for the ACS you specified. Wait until the switch completes, then, re-enter the switch lmu command.

- Switch: Switch lmu failed, port is offline.

Explanation: You specified an ACS with no ports online. Vary at least one port online to each LC or LMU.

## Display Area Messages

None.

## unlock

The `unlock` command removes active locks (associated with your current lock ID) on a specified drive or cartridge or removes all active locks.

## Format

```
unlock type identifier...| all
```

## Options

- `type identifier`  
specifies a library component. The following table lists the components that you can unlock.

**Table 12-8 Valid Components for Unlock**

| Library Component | type   | identifier      |
|-------------------|--------|-----------------|
| drive             | drive  | <i>drive_id</i> |
| volume            | volume | <i>vol_id</i>   |

- `all`  
specifies all active locks.

## Usage

Use the `unlock` command to remove active locks on specified cartridges and drives or remove all active locks. The `lock ID` of the component must match your current lock ID.

When you unlock a component that is not in use, ACSLS checks for pending locks on the component. If there is a pending lock, ACSLS locks the component with the lock ID of the pending lock.

**Note:**

The `unlock` command always resets your lock ID to 0.

## Examples

- To remove active locks on drive 0,0,2,0:  
`unlock drive 0,0,2,0`
- To remove active locks on all locked cartridges:  
`unlock volume all`

See also:

| For information about...                                            | See...                     |
|---------------------------------------------------------------------|----------------------------|
| Removing all active or pending locks on a specified drive or volume | <a href="#">clear lock</a> |
| Locking drives and volumes                                          | <a href="#">lock</a>       |
| Displaying the lock status of a drive or cartridge                  | <a href="#">query lock</a> |
| Setting your lock ID                                                | <a href="#">set lock</a>   |
| Displaying your lock or user ID                                     | <a href="#">show</a>       |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- `Unlock: Unlock completed, Success.`  
`Unlock: drive drive_id unlocked.`  
Explanation: ACSLS unlocked the specified drive.  
Variable: The `drive_id` is the unlocked drive.
- `Unlock: Unlock completed, Success.`  
`Unlock: volume vol_id unlocked.`  
Explanation: ACSLS unlocked the specified volume.  
Variable: The `vol_id` is the unlocked volume.

### Intermediate Messages:

None.

### Error Messages:

- `Unlock: Unlock of drive drive_id failed, status.`  
Explanation: ACSLS cannot unlock the drive.  
Variable: The `status` is the reason for the failure. For more information on common status messages, see ACSLS Messages.



- `Unlock: Unlock of volume vol_id failed, status.`

Explanation: ACSLS cannot unlock the volume.

Variable: The `status` is the reason for the failure. For more information on common status messages, see ACSLS Messages.

## Display Area Messages

None.

## uptime

The `uptime` command indicates the amount of elapsed time ACSLS has been enabled and/or the last restart of the ACSLS library manager.

## Format

```
uptime
```

## Usage

Use the `uptime` command to display the amount of elapsed time ACSLS has been enabled and/or the last restart of the ACSLS library manager, in the following format:

```
ACSLS uptime=<days-hrs:min:sec>
```

Where:

- `days`  
is the number of days.
- `hrs`  
is the number of hours.
- `min`  
is the number of minutes.
- `sec`  
is the number of seconds:

## Example

```
uptime
```

```
ACSLS uptime=4-02:08:27
```

## vary

The `vary` command changes the state of a library component.

## Format

```
vary type identifier...state [force]
```

## Options

- `type [identifier]`  
specifies a library component. The following table lists the components you can vary.

**Table 12-9 Valid Components for Vary**

| ACS Component | type  | identifier           |
|---------------|-------|----------------------|
| ACS           | acs   | <i>acs_id</i>        |
| LSM           | lsm   | <i>acs_id,lsm_id</i> |
| CAP           | cap   | <i>cap_id</i>        |
| drive         | drive | <i>drive_id</i>      |
| port          | port  | <i>port_id</i>       |

- `state`  
specifies one of the following device states: `online` (online to clients and `cmd_proc`), `offline`, or `diagnostic` (online to `cmd_proc` only).  
You can vary an ACS, LSM, CAP, or drive online, offline, or diagnostic. You can vary a port either online or offline. See "Usage " for more information about device states.
- `force`  
varies an ACS, LSM, CAP, or drive `offline` after processing only the current robotic request.

## Usage

Use the `vary` command to change the state of an ACS, LSM, CAP, drive, or port. The following sections tell how each of the device states affects library components.

- `vary offline`  
The following table shows the results of `vary offline` for each ACS component. When you vary a component offline and it first goes to offline-pending state, ACSLS processes all active and pending requests for the component, rejects new requests, then varies the component offline.

**Table 12-10 Vary Offline Results**

| ACS Component | Results                                                                                                   |
|---------------|-----------------------------------------------------------------------------------------------------------|
| ACS           | ACS and subcomponents go offline-pending, then offline. LSMs must go offline before the ACS goes offline. |
| LSM           | LSM goes offline-pending, then offline.                                                                   |
| CAP           | CAP goes offline-pending, then offline.                                                                   |
| drive         | If available, drive goes offline immediately. If in use, drive stays online.                              |

**Table 12-10 (Cont.) Vary Offline Results**

| ACS Component | Results                                                                                                                                                                           |
|---------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| port          | Port goes offline immediately if: <ul style="list-style-type: none"> <li>• ACS has other ports online, or</li> <li>• ACS is offline.</li> </ul> Otherwise, the port stays online. |

- `vary offline force`

The following table shows the results of `vary offline force` for each ACS component. This option is useful if the device must be taken to the offline state while it is involved in extended activity (for example, during an audit).

**Table 12-11 Vary Offline Force Results**

| ACS Component | Results                                                                                                                                                                                                            |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ACS           | Only the current robotic request completes, then the ACS and subcomponents go offline immediately. Pending requests are discarded and new requests are rejected. LSMs must go offline before the ACS goes offline. |
| LSM           | Only the current robotic request completes, then the LSM goes offline immediately. Pending requests are discarded and new requests are rejected.                                                                   |
| CAP           | Only the current robotic request completes, then the CAP goes offline immediately. Pending requests are discarded and new requests are rejected.                                                                   |
| drive         | Only the current robotic request completes, then the drive goes offline immediately. Pending requests are discarded and new requests are rejected.                                                                 |
| port          | Not valid.                                                                                                                                                                                                         |

- `vary diagnostic`

The `vary diagnostic` request places the specified component in the diagnostic state (online to `cmd_proc` only). ACSLS processes all active and pending requests for the component, rejects new client application requests, then varies the component to diagnostic state. For an ACS, all of its subordinate LSMs are also varied to diagnostic.

- `vary online`

The following table shows the results of `vary online` for each ACS component. When you `vary` a component online and it first goes to recovery state, ACSLS processes all active and pending requests for the component, rejects new requests, then varies the component online. When the component goes to online, ACSLS processes all requests for the component.

**Table 12-12 Vary Online Results**

| ACS Component | Results                                                                                                                                     |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| ACS           | If ACS is offline, ACS and its LSM go to recovery, then online.<br>If ACS is in diagnostic state, ACS and its LSM go to online immediately. |
| LSM           | LSM goes to recovery, then online. You cannot vary an LSM online if it is attached to an offline ACS.                                       |

**Table 12-12 (Cont.) Vary Online Results**

| ACS Component | Results                              |
|---------------|--------------------------------------|
| CAP           | CAP goes to recovery, then online.   |
| drive         | drive goes to recovery, then online. |
| port          | Port goes online immediately.        |

## Examples

- To vary drive 0,0,9,3 offline:  
`vary drive 0,0,9,3 offline`
- To vary CAP 0,0,0 diagnostic:  
`vary cap 0,0,0 diagnostic`
- To force lsm 0,1 offline:  
`vary lsm 0,1 offline force`



### Note:

IPLing the system does not change the state of these components. Installing or reconfiguring ACSLS places all components in the online state wherever possible.

When two or more partitions of the same library are managed by ACSLS, use separate `vary` commands to vary the same CAP (identified as being in different ACSs).

**Example:** If ACS 0 and ACS 1 are two partitions of the same library, vary the CAP online or offline independently for each partition (and each ACS). The same CAP can be varied online to one partition (ACS) and offline to another partition (ACS).

See also:

| For information about...                     | See...                         |
|----------------------------------------------|--------------------------------|
| Displaying the status of a library component | <a href="#">query commands</a> |

## Command Area Messages

This section discusses the command area messages.

### Success Messages:

- `Vary: type identifier Varied state`  
Explanation: ACSLS changed the state of the specified library component.  
Variable:

- The `type` is the library component type.
- The `identifier` is the library component identifier.
- The `state` is the new state of the library component.

**Intermediate Messages:**

None.

**Error Messages:**

- `In-transit volume recovery incomplete.`

Explanation: The LSM robot cannot dispose of in-transit volumes while the LSM or its ACS is varied online. If you are varying an ACS online, the ACS state is changed to online immediately, but any LSMs that are unable to complete in-transit volume recovery remain offline.

- `Vary: Vary type identifier failed, Drive in use.`

Explanation: ACSLS cannot change the state of the specified library component because the command involved an in-use drive.

Variable:

- The `type` is the library component type.
- The `identifier` is the library component identifier.

- `Vary: Vary type identifier failed, Incorrect lockid`

Explanation: ACSLS cannot change the state of the specified library component because the `drive lock_id` does not match your `lock_id`.

Variable:

- The `type` is the library component type.
- The `identifier` is the library component identifier.

- `Vary: Vary type identifier failed, State unchanged.`

Explanation: ACSLS cannot change the state of the specified library component because the component is already in the requested state.

Variable:

- The `type` is the library component type.
- The `identifier` is the library component identifier.

- `Vary: Vary type identifier failed, Vary disallowed.`

Explanation: ACSLS cannot change the state of the specified library component because a `vary diagnostic` request was issued from a client application.

Variable:

- The `type` is the library component type.
- The `identifier` is the library component identifier.

- `Vary: Vary type identifier failed, Incorrect state.`

Explanation: ACSLS cannot change the state of the specified library component because you tried to `vary`:

- A CAP online when the LSM or ACS is in the diagnostic or offline state.
- A CAP diagnostic when the LSM or ACS is offline.
- An LSM online when the ACS is in the diagnostic or offline state.
- An LSM diagnostic if the ACS is offline.

Variable:

- The `type` is the library component type.
- The `identifier` is the library component identifier.
- Vary: Vary type identifier failed, Vary in progress.

Explanation: ACSLS cannot change the state of the specified library component because:

- A CAP, its LSM, or its ACS is in a transitory (recovery or offline-pending) state.
- An LSM, its ACS, or a subordinate CAP is in a transitory state.
- An ACS, a subordinate LSM, or a subordinate CAP is in a transitory state.
- A vary offline/force request overrode the attempt to change an ACS, LSM, or CAP to the diagnostic or online state.
- A drive is in a transitory state of recovery.

Variable:

- The `type` is the library component type.
- The `identifier` is the library component identifier.
- Vary: Vary type identifier failed, State unchanged.

Explanation: ACSLS cannot change the state of the specified library component because you tried to vary an ACS, LSM, or CAP offline and a vary offline/force request overrode the attempt.

Variable:

- `type` is the library component type.
- `identifier` is the library component identifier.
- Vary: Vary type identifier failed, ACS online.

Explanation: ACSLS cannot change the state of the specified library component because you tried to vary the last port offline in an online ACS.

Variable:

- The `type` is the library component type.
- The `identifier` is the library component identifier.
- Vary: Vary type identifier failed, No ports online.

Explanation: ACSLS cannot change the state of the specified library component because there are no ports online for an ACS.

Variable:

- The `type` is the library component type.
- The `identifier` is the library component identifier.

## Display Area Messages

One of the following messages appears when an ACS, LSM, drive, or port changes state.

- ACS `acs_id`: state
- LSM `lsm_id`: state
- Drive `drive_id`: state
- Port `port_id`: state

The following message appears if, during recovery of in-transit volumes, a cartridge label is unreadable or a duplicate. The cartridge is moved to the CAP.

CAP `cap_id`. Remove volumes from CAP.

- `cap_id`  
is the CAP containing the cartridges.

# 13

## Display Command Reference

In this chapter you will learn how to use the `display` command.

If you think in terms of SQL and look at the options, you will realize the power and flexibility in the `display` command. The `display` command is a powerful tool for reporting information from the ACSLS database. Like a SQL `SELECT` statement, `display` allows you to specify:

- the database table (ACSL object) you want to display.
- wildcards.
- conditions that objects must meet to be selected.
- fields to report and the order in which they will be reported.
- the sort order in which output will be displayed.
- that only the count of the number of objects meeting the selection criteria be displayed.

The result is a `query` facility that lets you customize what you want displayed and how it should be presented. Much more information is available through the `display` command than is possible through ACSLS `query` commands. For example, `display volume` gives access to volume information that was only available through a custom `volrpt`. `display lsm` and `display drive` provide LSM and tape drive serial numbers.

The `display` commands are:

- [display cap](#)  
Displays specific CAP information based on the options selected.
- [display cell](#)  
Displays specific cell information based on the options selected.
- [display drive](#)  
Displays specific drive information based on the options selected.
- [display lock](#)  
Displays specific lock information based on user ID.
- [display lsm](#)  
Displays specific LSM information based on the options selected.
- [display panel](#)  
Displays specific library panel information based on the options selected.
- [display pool](#)  
Displays specific scratch pool information based on the options selected.
- [display port](#)  
Displays specific port information based on the options selected.
- [display volume](#)  
Displays specific volume information based on the options selected.



## Using display Command Options

The `display` command displays various fields of information for different library components. The `display` command allows you to select the order and format in which the output is displayed.

The information displayed can include multiple columns and long lines. To prevent lines from wrapping, you can re-size the terminal window used for display. A `cmd_proc -lq` window will probably be your best choice.

Wildcards, numeric or alpha ranges, and selection lists can be used with the `display` command.

## Using Wildcard Characters

- The wildcard character, `*`, can be used to substitute: for certain `display` operands; for one or more subfields in compound display operands; or at the beginning or end of a volume ID. It must match one or more characters.
- Many of the ACSLs library components have a primary key composed of multiple fields.

For example, the primary key of:

- an LSM is `acs,lsm`
- a drive is `acs,lsm,panel,drive`

- When you wildcard a primary key that contains multiple fields, you can either specify an asterisk (`*`) for all fields in the key, or you can wildcard individual key fields. If you specify (`*`) some key fields and wildcard others, specify all of the key fields.
- Do not use an asterisk (`*`) for a `vol_id`. Use an underscore (`_`) to match a single character in a `vol_id`. The underscore is positional, and represents a single character. It may be used more than once in a `vol_id` to represent more than one character.

## Examples

Examples of valid wildcards using the `*` character:

- `display drive *` (display all drives)
- `display drive 0,1,*,*` (displays all drives in LSM 0,1)
- `display volume *100` (displays all volumes ending in 100)
- `display drive * -volume RB0001` (displays the drive(s) that contain volume RB0001)

Examples of invalid wildcards using the `*` character:

- `display drive 0,1,*` (Asterisks for both the panel and drive positions must be specified when selecting all drives in LSM 0,1)
- `display cap * *` (Only one asterisk is allowed when selecting all CAPs)

- `display volume 1*111` (The embedded asterisk is not allowed. Use `display volume 1_111`).

## Using Ranges

- A range is denoted with a '-' (hyphen or dash).
- Rules for using ranges, are:
  - Negative numbers are not allowed, except for volume home LSM.
  - <n and >n are allowed. <n is all numerics less than n. >n is all numerics greater than n.
  - Multiple numbers or ranges can be used, separated by a white space.
  - Volume ranges can be alphanumeric.

## Examples

Examples of valid ranges, are:

- `display volume * -drive 0,1,2,2-5` (displays all volumes in acs 0, lsm 1, panel 2, drives 2 through 5)
- `display lsm * -state online -free_cells >50 -type 9730` (displays all 9730 lsms in an online state with a free cell count greater than 50)

Examples of invalid ranges, are:

- `display drive 0,1,1,1-*` (\* is not valid in a range)
- `display volume * -drive 0,1,?,1-5` (? is not valid in a range)

## Format

```
display type arg ... [ -selection ... ] [ [ -c ] | [ -f field ... ] [ -s
sort_field ... ] [ -n n ] ]
```

## Options

- `type`  
specifies the object to be displayed. Valid types are: cap, cell, drive, lock, LSMs, panel, pool, port, and volume.
- `arg`  
(argument) specifies the identifiers for the object type. Only objects that match one or more of the identifiers are returned.
- `-selection`  
specifies additional selection criteria. To be selected, a component must match at least one of the selection identifiers for each selection criteria.
- `-c`  
(count) causes the request to display only the number of objects that satisfy the `arg` and `selection` criteria. This option is mutually exclusive with the `-f field`, `-s sort_field`, and `-n n` options.

- `-f field`  
specifies the fields of information to be returned, and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If `-f` is specified, the default fields are not displayed unless they are key fields or have been specified. If `-f` is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the `-f` option.  
  
The list of fields is separated by spaces.
- `-s sort_field`  
sorts the objects returned by the field(s) selected in the order specified. The default sort order for each display command is discussed under each command. Sorting is based on internal database values and may not always display in alphanumeric order.  
  
All returned fields for a type are valid `sort_fields`.
- `-n n`  
specifies the maximum number of objects to be displayed.

## Start `cmd_proc` in Line Mode when Displaying Lines Longer than 80 Characters

The `cmd_proc` command started in curses mode cannot display lines longer than 80 characters, and the `cmd_proc` window hangs if it attempts to display a line longer than 80 characters.

Because of this, it is a good idea to start `cmd_proc` in line mode (with the `-l` option) when displaying many optional fields. Example:

```
display drive * -f volume type state serial num wwn
```

using a `cmd_proc` started as `cmd_proc -l`.

## Commands

This section discusses the display commands.

### display cap

The `display cap` command displays specific CAP information based on the options selected.

#### Note:

On SL3000s and SL8500s with 4.70 and later firmware. To prevent CAP IDs from changing when additional modules are added to the SL3000 or additional CAPs are installed, the library returns all possible CAP locations in its configuration information. CAPs that are not installed will be reported as "not installed" in the CAP Status.

ACSLS only adds database records for CAP locations that actually have CAPs installed.



#### Note:

See [Using display Command Options](#) for details about specifying display selection criteria and selecting the data to be displayed.

## Format

```
display cap cap_id ... ][ -availability cap_availability ...] [ -status
cap_status ... ][-priority cap_priority ... ] [ -state cap_state ... ] [ -manual
| -automatic ] [ -condition cap_condition ... ] [ [ -c ] | [ -f field ... ] [ -s
sort_field ... ] [ -n n ] ]
```

## Fields

The fields for `display cap`, are:

- **key fields:** `acs`, `lsm`, `cap`
- **default fields:** `acs`, `lsm`, `cap`, `status`, `priority`, `state`, `mode`, `size`, `availability`
- **available fields:** `acs`, `lsm`, `cap`, `status`, `priority`, `state`, `desired_state`, `mode`, `size`, `condition`, `availability`

## Options

- `cap_id`  
specifies the CAP(s) to display. A `cap_id` is of the format: `acs`, `lsm`, `cap`.  
Wildcard character, `*`, can be used in any `cap_id` subfield or used to represent all subfields. However, if it represents all `cap_ids` (example: `display cap *` or `display cap *,*,*`), additional `cap_ids` (either numeric or `*`) are not allowed. Numeric ranges apply to any subfield of `cap_id`.
- `availability cap_availability`  
specifies one or more cap availabilities as additional selection criteria. CAP availabilities are: `lib_not_partn`, `cap_shared`, `cap_dedicated`, `dedicated_other`, `not_installed`.  
CAPs that are `dedicated_other` or `not_installed` are not defined in the ACSLS database. These values are only seen when a CAP's availability has changed (since it was configured). Reconfigure to remove these CAPs.
- `-status cap_status`  
specifies one or more CAP statuses as additional selection criteria. CAP statuses are: `audit`, `available`, `eject`, `enter`, `insert_magazines`.
- `-priority cap_priority`  
specifies one or more CAP priorities. Numeric range rules apply. CAP numeric ranges: 0-16.
- `-state cap_state`

specifies one or more CAP states. CAP states are: `diagnostic`, `offline`, `offline_pending`, `online`, `recovery`.

- `-manual`  
selects CAPs in manual mode.
- `-automatic`  
selects CAPs in automatic mode.
- `-condition cap_condition`  
specifies one or more CAP conditions from which to select. Valid conditions are: `inoperative`, `maint_required`, or `operative`.

 **Note:**

An `inoperative` or `maint_required` condition is only cleared when a drive, LSM, or ACS is varied online. So, a CAP condition of `inoperative` or `maint_required` may be inaccurate.

- `-c`  
(count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the `-f` field, `-s` sort\_field, and `-n n` options.
- `-f field`  
specifies the fields of information to be returned and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If `-f` is specified, the default fields are not displayed unless they are key fields or have been specified. If `-f` is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the `-f` option.

The list of fields is separated by spaces.

The following is a new CAP field that is not reported by `query cap`.

— `desired_state`

The `desired state` for an ACS, port, LSM, drive, or CAP is the availability that you want for the component. ACSLS sets desired state when an explicit `vary` operation is performed. (This is a `vary` from a `cmd_proc` or `ACSAPI client` command. It is not a `vary` that ACSLS internally generates because of a library status change.)

The current state (specified as "state") for an ACS, port, LSM, drive, or CAP is the current availability of the component limited by desired state. When a drive is varied online, its desired state is online. However, its current state could be offline because the drive is inoperative or is in an LSM that is offline or not ready. The current state of an ACS, port, LSM, drive, or CAP is also referred to as the "state" of the component in the results of `query` and `display` commands.

The current state of a CAP, based on availability, is not maintained at this time.

The availability of logical libraries and tape drives in logical libraries is also governed by the desired state that you set for both the underlying physical library and the logical library. If the desired states for both the physical library and logical library are online, the current states of logical libraries and logical tape drives reflect the current state of the underlying physical libraries and drives.

- `-s sort_field`  
sorts the objects returned by the field(s), selected in the order specified. Sorting is based on internal database values and may not always display in alphanumeric order.  
All returned fields for a type are valid `sort_fields`.
- `-n n`  
specifies the maximum number of objects to be displayed.

## Examples

- To display all CAP data:  
`display cap *`
- To display CAP data for ACS 1, LSM 1, CAP 1:  
`display cap 1,1,1`
- To display all manual mode CAPs in LSM 1, 0:  
`display cap 1,0,* -manual`
- To display all CAPs in an offline pending state:  
`display cap * -state offline_pending`

## display cell

The `display cell` command displays specific information about a cell based on the options selected.



### Note:

See "[Using display Command Options](#)" for details about specifying display selection criteria and selecting the data to be displayed.

## Format

```
display cell cell_loc ... [ -status cell_status ... ] [ [ -c ] | [ -f
field... ] [ -s sort_field ... ] [ -n n ] ]
```

## Fields

The fields for `display cell`, are:

- **key fields:** acs, lsm, panel, row, column,
- **default fields:** acs, lsm, panel, row, column, status

- **available fields:** `acs, lsm, panel, row, column, status`

## Options

- `-status cell_status`  
specifies one or more cell statuses as additional selection criteria. Valid cell statuses: `empty, full, inaccessible, reserved`.  
The wildcard character, `*`, can be used in any `cell_id` subfield or used to represent all subfields. However, if it represents all `cell_ids` (example: `display cell *` or `display cell_id *,*,*`), additional `cell_ids` (either numeric or `*`) are not allowed. Numeric ranges apply to any subfield of `cell_id`.
- `-c`  
(count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the `-f field`, `-s sort_field`, and `-n n` options.
- `-f field`  
specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If `-f` is specified, the default fields are not displayed unless they are key fields or have been specified. If `-f` is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the `-f` option.  
The list of fields is separated by spaces.
- `-s sort_field`  
sorts the objects returned by the field(s), selected in the order specified. Sorting is based on internal database values and may not always display in alphanumeric order.  
All returned fields for a type are a valid `sort_field`.
- `-n n`  
specifies the maximum number of objects to be displayed.

## Example

To display information for cells with a reserved status:

```
display cell * -status reserved
```

## display drive

The `display drive` command displays specific drive information based on the options selected.

**Note:**

See [Using display Command Options](#) for details about specifying display selection criteria and selecting the data to be displayed.

## Format

```
display drive drive_id ... [ -status drive_status ... ][-state drive_state ... ]
[ -type drive_type ... ][-volume vol_id ... ] [ -lock lock_id... ] [ -serial
drive_serial_num ... ] [ -condition drive_condition ...][ [ -c ] | [ -f
field ... ] [ -s sort_field ... ] [ -n n ] ]
```

## Fields

The fields for `display drive` are:

- **key fields:** `acs`, `lsm`, `panel`, `drive`  
**default fields:** `acs`, `lsm`, `panel`, `drive`, `status`, `state`, `volume`, `type`
- **available fields:** `acs`, `lsm`, `panel`, `drive`, `status`, `state`, `volume`, `type`, `lock`, `desired_state`, `serial_num`, `condition`, `wwn`, `last_dismount_time`, `error`

## Options

- `drive_id`  
displays the `drive_id` in the following format: `acs`, `lsm`, `panel`, `drive`.  
Wildcard character,\*, can be used in any `drive_id` subfield or used to represent all subfields. However, if it represents all `drive_ids` (example: `display drive *` or `display drive_id*,*,*`), additional `drive_ids` (either numeric or \*) are not allowed. Numeric ranges apply to any subfield of `drive_id`.
- `-status drive_status`  
displays one of the following statuses as additional selection criteria: `available`, `in_use`, `reserved`.  
Possible scenario: You perform a `query drive all` and get a message saying that the drive is in use. You then perform a `display drive *` and get a message saying that the drive is reserved. This means: the reserved status for the drive implies that a mount request is in progress and that a volume is on its way to the drive. At the same time, the drive is considered to be in use.
- `-state drive_state`  
displays the following states for one or more drives: `diagnostic`, `online`, `offline`, `recovery`.
- `-type drive_type`  
displays drives by transport type.
- `-volume vol_id`  
displays volume(s) mounted on the drive(s) selected. A `vol_id` can be a 6-alphanumeric string or wildcard string that specifies a tape volume or range of volume IDs.



- `-lock lock_id`  
displays drive information on locked drives by lock ID.
- `-serial drive_serial_num`  
specifies the drive serial number.
- `-condition drive_condition`  
displays condition of the drive(s) specified. Valid conditions are: `operative`, `inoperative`, or `maint_required`.

 **Note:**

An `inoperative` or `maint_required` condition is only cleared when a drive, LSM, or ACS is varied online. So, a drive condition of `inoperative` or `maint_required` may be inaccurate.

- `-c`  
(count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the `-f field`, `-s sort_field`, and `-n n` options.
- `-s sort_field`  
sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in alphanumeric order.  
  
All returned fields for a type are valid `sort_fields`.
- `-f field`  
specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If `-f` is specified, the default fields are not displayed unless they are key fields or have been specified. If `-f` is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the `-f` option.

The list of fields is separated by spaces. The following is a list of new tape drive field descriptions that are not reported by `query drive`.

— `desired_state`

The desired state for an ACS, port, LSM, drive, or CAP is the availability that you want for the component. ACSLS sets desired state when an explicit `vary` operation is performed. (This is a `vary` from a `cmd_proc` or ACSAPI `client` command. It is not a `vary` that ACSLS internally generates because of a library status change.)

The current state (specified as "state") for an ACS, port, LSM, drive, or CAP is the current availability of the component limited by desired state. When a drive is varied online, its desired state is online. However, its current state could be offline because the drive is `inoperative` or is in an LSM that is offline or not ready. The current state of an ACS, port, LSM, drive, or CAP is also referred to

as the "state" of the component in the results of `query` and `display` commands.

The availability of logical libraries and tape drives in logical libraries is also governed by the desired state that you set for both the underlying physical library and the logical library. If the desired states for both the physical library and logical library are online, the current states of logical libraries and logical tape drives reflect the current state of the underlying physical libraries and drives.

– `serial_num`

ACSLs can only report drive serial numbers when the drive reports its serial number to the library and the library reports its serial number to ACSLS. Only modern libraries and drives report the drive serial numbers to ACSLS. This includes T9840, T10000, and LTO drives. SL150, SL3000 and SL8500, and later libraries report serial numbers to ACSLS.

ACSLs reports the serial number on the tape drive "brick," not the serial number on the drive tray.

The serial number on the drive brick is the serial number reported by the drive to the library and to you through SL Console and ACSLS. It is also the serial number reported over the SCSI/Fibre data path (such as, to the backup application).

The drive tray is labeled with its own serial number. The serial number on the drive tray label licenses service and so on, through StorageTek. The number on the drive tray is used for activating, so drives (drive "bricks") can be replaced without needing to update the activated serial number.

There is minimal communication between the drive tray and drive brick. This is displayed by way of a few ready/not ready indicators. The drive tray is basically a power supply for the drive without any smarts.

– `wnn`

The drive's World Wide Name. The WWN is 8 bytes (binary). It is displayed as 16 hex characters (0-9 and A-F) with dot separators between the pairs of hex characters (bytes).

The WWN displayed is the WWN of the node. The WWN of the first port (Port A) is one number greater. If the drive has two ports, the WWN of Port B is two numbers greater.

The drive's WWN is available for SL3000 libraries and for SL8500 libraries with 3.94 and later firmware.

– `last_dismount_time`

The date and time of the last time a cartridge was dismounted from the drive. ACSLS uses this to order the drives reported to `query mount`. The drives are listed primarily by LSM closeness (least pass-thrus) to the volume being mounted and secondarily by earliest `last_dismount_time`. Ordering the drives by earliest `last_dismount_time` evens out the use of drives.

These statistics are reported to ACSLS when a cartridge is dismounted from a drive. The following statistics fields are only reported for these libraries and drives:

Libraries:

- \* All SL3000s.
- \* SL8500s with 4.13 and later firmware.

Tape drives:

- \* T9840C and T9840D with 1.42 and later drive firmware.
  - \* T10000A
- error - (drive maintenance required)

This is currently only reported for T10000 drives. Values are:

- \* none - No drive error reported.
- \* maint\_reqd - Maintenance is required for the tape drive.

This is only reported when the tape drive has determined that the drive is to be avoided and should be taken offline. You should call for service.

Primary drive error scenarios:

In these scenarios, the drive and library waits for manual intervention.

**Table 13-1 Drive and Library waiting for Manual Intervention**

| Unload failures                | The cartridge is stuck in the drive                                 |
|--------------------------------|---------------------------------------------------------------------|
| Drive Leader Off Hook failures | The drive won't indicate a drive error until the media is extracted |
| Drive IPL failures             | The drive stops communicating with the library.                     |

- `-s sort_field`  
sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in alphanumeric order. For example, `drive type` is sorted by an internal numeric drive type and not by the literal that is display.  
All returned fields for a type are valid `sort_fields`.
- `-n n`  
specifies the maximum number of objects to be displayed.

## Examples

- To display all drives in panel 0,1,1.  
`display drive 0,1,1,*`
- To display all drives in LSM 0,1.  
`display drive 0,1,*,*`
- To display a maximum of 56 rows for all drives in ACS 1, sorted by panel and drive:  
`display drive 1,*,*,* -s panel drive -n 56`
- To display drive data for ACS 1, LSM 1, with status available:  
`display drive 1,1,*,* -status available`
- To display drive types, serial numbers, and world wide names (wwn) for all drives:  
`display drive * -f type serial_num wwn`
- To display all drives desired state:

```
display drive * -f state desired_state
```

## display lock

The `display lock` command displays specific lock information by `lock_id` based on the options selected.



### Note:

See [Using display Command Options](#) for details about specifying display selection criteria and selecting the data to be displayed.

## Format

```
display lock lock_id ... [ -user user_id ... ] [ [ -c ] | [ -f field ... ] [ -s
sort_field ... ] [ -n n ] ]
```

## Fields

The fields for `display lock`, are:

- key fields: `lock_id`
- default fields: `lock_id, user_id`
- available fields: `lock_id, user_id`

## Options

- `lock_id`  
displays the numeric `lock_id`.  
Wildcard character, `*`, can be used in any `lock_id` subfield or used to represent all subfields. However, if it represents all `lock_ids` (example: `display lock *` or `display lock_id *,*,*`), additional `lock_ids` (either numeric or `*`) are not allowed. Numeric ranges apply to any subfield of `lock_id`.
- `user_id`  
specifies the user name (example: `acsss`) that has locked the volume or transport. The `user_id` wraps after 80 characters.
- `-c`  
(count) causes the request to display only the number of objects that satisfy the `arg` and `selection` criteria. This option is mutually exclusive with the `-f field`, `-s sort_field`, and `-n n` options.
- `-f field`  
specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If `-f` is specified, the default fields are not displayed unless they are key fields or have been specified. If `-f` is not specified, then the default fields are returned. Key fields for each

type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the `-f` option.

The list of fields is separated by spaces.

- `-s sort_field`

sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in alphanumeric order.

All returned fields for a type are valid sort fields.

- `-n n`

specifies the maximum number of objects to be displayed.

## Example

- To display lock information for `lock_id 2`:

```
display lock 2
```

## display lsm

The `display lsm` command displays specific LSM information based on the options selected.



### Note:

See [Using display Command Options](#) for details about specifying display selection criteria and selecting the data to be displayed.

## Format

```
display lsm lsm_id ... [ -status lsm_status ... ] [-state lsm_state ... ]
[ -free_cells cell_count ... ] [ -type lsm_type ... ] [ -serial
lsm_serial_num ... ] [ -condition lsm_condition ] [ -door_open | -
door_closed ] [ [ -c ] | [ -f field ... ] [ -s sort_field ... ] [ -n n ] ]
```

## Fields

The fields for `display lsm`, are:

- **key fields:** `acs`, `lsm`
- **default fields:** `acs`, `lsm`, `status`, `state`, `free_cells`
- **available fields:** `acs`, `lsm`, `status`, `state`, `desired_state`, `free_cells`, `type`, `serial_num`, `activity`, `condition`, `door_status`
- `lsm_id`

displays the lsm id in the following format: `acs, lsm`.

Wildcard character, `*`, can be used in any `lsm_id` subfield or used to represent all subfields. However, if it represents all `lsm_ids` (example: `display lsm * or`

display *lsm\_id* \*,\*), additional *lsm\_ids* (either numeric or \*) are not allowed. Numeric ranges apply to any subfield of *lsm\_id*.

- `-status lsm_status`  
specifies one or more LSM statuses from which to select. Valid statuses are: `audit`, `normal`
- `-state lsm_state`  
specifies one or more LSM states from which to select. Valid states are: `diagnostic`, `online`, `offline`, `offline_pending`, `recovery`.
- `-free_cells cell_count`  
selects LSMs by the number of free cells.
- `-type lsm_type`  
displays LSMs by type.
- `-serial lsm_serial_num`  
displays serial numbers for selected LSMs.  
With SL8500 6.0+ and SL3000 3.0+ firmware, the library reports the licensed serial number. (To get the SL8500 library number, add 1 to the LSM number and divide by 4.)
- `-condition lsm_condition`  
displays condition of the selected LSMs. Valid conditions are: `operative`, `inoperative`, or `maint_required`.

 **Note:**

An `inoperative` or `maint_required` condition is only cleared when a drive, LSM, or ACS is varied online. So, an LSM condition of `inoperative` or `maint_required` may be inaccurate.

- `-door_open`  
displays LSMs with open doors.
- `-door_closed`  
displays LSMs with closed doors.
- `-c`  
(count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the `-f` field, `-s` *sort\_field*, and `-n` *n* options.
- `-f field`  
specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If `-f` is specified, the default fields are not displayed unless they are key fields or have been specified. If `-f` is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the `-f` option.

The list of fields is separated by spaces. The following are new LSM fields that are not reported by `query lsm`.

– `desired_state`

The desired state for an ACS, port, LSM, drive, or CAP is the availability that you want for the component. ACSLS sets desired state when an explicit `vary` operation is performed. (This is a `vary` from a `cmd_proc` or `ACSAPI client` command. It is not a `vary` that ACSLS internally generates because of a library status change.)

The current state (specified as "state") for an ACS, port, LSM, drive, or CAP is the current availability of the component limited by desired state. When a drive is varied online, its desired state is online. However, its current state could be offline because the drive is inoperative or is in an LSM that is offline or not ready. The current state of an ACS, port, LSM, drive, or CAP is also referred to as the "state" of the component in the results of `query` and `display` commands.

The availability of logical libraries and tape drives in logical libraries is also governed by the desired state that you set for both the underlying physical library and the logical library. If the desired states for both the physical library and logical library are online, the current states of logical libraries and logical tape drives reflect the current state of the underlying physical libraries and drives.

– `serial_num`

The serial number that ACSLS reports for the LSM is the library's serial number that is recorded in the HBC card of the SL3000 and SL8500 libraries (before SL8500 6.00 firmware or SL3000 3.0 firmware.). SL8500 and SL3000 libraries with later firmware report the frame serial number which is the activated serial number.

SL8500 and SL3000 libraries are activated through the serial number for the base module frame. This serial number is on the UL label on the frame and is recorded in the HBC card in the library. (Manufacturing can re-generate replacement HBC cards.) These serial numbers start with 516 for SL8500s and 571 for SL3000s. The frame serial number can be displayed through the SL Console (System Detail -> Properties -> General).

In the future, library firmware will report this frame serial number to ACSLS. Whenever an LSM serial number changes, ACSLS automatically updates the serial number when an LSM is varied online.

• `-s sort_field`

sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in alphanumeric order.

All returned fields for a type are valid sort fields.

• `-n n`

specifies the maximum number of objects to be displayed.

## Examples

- To display information for all LSMs in the library for type 9714:

```
display lsm * -type 9714
```

- To display types all LSMs:

```
display lsm * -f type
```

- To display library serial numbers:

```
display lsm * -f serial_num
```

- To display all LSMs states and desired\_state:

```
display lsm * -f state desired_state
```

## display panel

The `display panel` command displays specific library panel information based on the options selected.



### Note:

See [Using display Command Options](#) for details about specifying display selection criteria and selecting the data to be displayed.

## Format

```
display panel panel_id ...[ -type panel_type ... ] [ [ -c ] |[ -f field ... ]
[ -s sort_field ... ] [ -n n ] ]
```

## Fields

The fields for `display panel`, are:

- **key fields:** `acs`, `lsm`, `panel`
- **default fields:** `acs`, `lsm`, `panel`, `type`
- **available fields:** `acs`, `lsm`, `panel`, `type`

## Options

- `panel_id`

displays the panel id in the following format: `acs`, `lsm`, `panel`.

Wildcard character, `*`, can be used in any `panel_id` subfield or used to represent all subfields. However, if it represents all `panel_ids` (example: `display panel *` or `display panel_id *,*,*`), additional `panel_ids` (either numeric or `*`) are not allowed. Numeric ranges apply to any subfield of `panel_id`.

- `-type panel_type`  
specifies one or more valid panel types.
- `-c`



(count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the `-f field`, `-s sort_field`, and `-n n` options.

- `-f field`  
specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If `-f` is specified, the default fields are not displayed unless they are key fields or have been specified. If `-f` is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the `-f` option.  
The list of fields is separated by spaces.
- `-s sort_field`  
sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in numeric order.  
All returned fields for a type are valid sort fields.
- `-n n`  
specifies the maximum number of objects to be displayed.

## Example

Display panel data for all panels:

```
display panel *
```

## display pool

The `display pool` command displays specific scratch pool information based on the options selected.



### Note:

See [Using display Command Options](#) for details about specifying display selection criteria and selecting the data to be displayed.

## Format

```
display pool pool_id... [ -low_water low_water_mark... | -high_water
high_water_mark... ] [-overflow | -no_overflow ] [ [ -c ] | [ -f field... ]
[ -s sort_field... ] [ -n n ] ]
```

## Fields

The fields for `display pool`, are:

- **key fields:** `pool_id`
- **default fields:** `pool_id`, `low_water`, `high_water`, `overflow`

- available fields: `pool_id`, `low_water`, `high_water`, `overflow`

## Options

- `pool_id`  
displays the pool id of a scratch pool.  
Wildcard character, `*`, can be used in any `pool_id` subfield or used to represent all subfields. However, if it represents all `pool_ids` (**example**: `display pool *` or `display pool_id *,*,*`), additional `pool_ids` (either numeric or `*`) are not allowed. Numeric ranges apply to any subfield of `pool_id`.
- `-low_water low_water_mark`  
specifies the minimum number of scratch cartridges in the pool. If a `low_water` mark is displayed with a hyphen (-) appended to it, the number of cartridges in the library for the selected scratch pool is within the designated `low_water` mark.
- `-high_water high_water_mark`  
specifies the maximum number of scratch cartridges in the pool. If a `high_water` mark is displayed with a hyphen (-) appended to it, the number of cartridges in the library for the selected scratch pool is within the designated `high_water` mark.
- `-overflow`  
selects pools wherein if a scratch mount is requested and if the pool is empty, the common pool satisfies the request.
- `-no_overflow`  
selects pools wherein if a scratch mount is requested and if the pool is empty, the mount fails without checking the common pool for available tapes.
- `-c`  
(count) causes the request to display only the number of objects that satisfy the `arg` and `selection` criteria. This option is mutually exclusive with the `-f field`, `-s sort_field`, and `-n n` options.
- `-f field`  
specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If `-f` is specified, the default fields are not displayed unless they are key fields or have been specified. If `-f` is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the `-f` option.  
The list of fields is separated by spaces.
- `-s sort_field`  
sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in numeric order.  
All returned fields for a type are valid sort fields.
- `-n n`  
specifies the maximum number of objects to be displayed.

## Examples

- To display all information for pool 5:

```
display pool 5
```

- To display all pools with overflow:

```
display pool * -overflow
```

## display port

The `display port` command displays specific port information based on the options selected.

### Note:

See [Using display Command Options](#) for details about specifying display selection criteria and selecting the data to be displayed.

## Format

```
display port port_id ... [ -online | -offline ] [ -name port_name ... ]  
[ [ -c ] | [ -f field ... ] [ -s sort_field ... ] [ -n n ] ]
```

## Fields

The fields for `display port` are:

- key fields: `acs`, `port`
- default fields: `acs`, `port`, `name`, `state`
- available fields: `acs`, `port`, `name`, `state`, `desired_state`

## Options

- *port\_id*

displays the port id in the following format: `acs`, *port*.

Wildcard character, `*`, can be used in any `port_id` subfield or used to represent all subfields. However, if it represents all `port_ids` (example: `display port *` or `display port_id *,*,*`), additional `port_ids` (either numeric or `*`) are not allowed. Numeric ranges apply to any subfield of `port_id`.

– `online`

selects and displays ports in an online state.

– `offline`

selects and displays ports in an offline state.

– name *port\_name*

specifies the port name.

- `-c`  
(count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the `-f field`, `-s sort_field`, and `-n n` options.
- `-f field`  
specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If `-f` is specified, the default fields are not displayed unless they are key fields or have been specified. If `-f` is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the `-f` option.  
  
The list of fields is separated by spaces. The following is a new port field that is not reported by `query port`.  
  
— `desired_state`  
  
The desired state for an ACS, port, LSM, drive, or CAP is the availability that you want for the component. ACSLS sets desired state when an explicit `vary` operation is performed. (This is a `vary` from a `cmd_proc` or ACSAPI `client` command. It is not a `vary` that ACSLS internally generates because of a library status change.)  
  
The current state for an ACS, port, LSM, drive, or CAP is the current availability of the component limited by desired state. When a drive is varied online, its desired state is online. However, its current state could be offline because the drive is inoperative or is in an LSM that is offline or not ready. The current state of an ACS, port, LSM, drive, or CAP is also referred to as the “state” of the component in the results of `query` and `display` commands.  
  
The availability of logical libraries and tape drives in logical libraries is also governed by the desired state that you set for both the underlying physical library and the logical library. If the desired states for both the physical library and logical library are online, the current states of logical libraries and logical tape drives reflect the current state of the underlying physical libraries and drives.
- `-s sort_field`  
sorts the objects returned by the field(s) selected in the order specified. Sorting is based on internal database values and may not always display in alphanumeric order  
  
All returned fields for a type are valid sort fields.
- `-n n`  
specifies the maximum number of objects to be displayed.

## Example

- To display all ports in an online state:

```
display port * -online
```

## display volume

The `display volume` command displays specific volume information based on the options selected.

**Note:**

See [Using display Command Options](#) for details about specifying display selection criteria and selecting the data to be displayed.

## Format

```
display volume vol_id ... [ -home acs,lsm,panel,row,column...] [ -drive
drive_loc ... ] [ -data | -scratch | -clean ] [ -media media_type ... ] [ -
pool pool_id... ] [ -standard | -virtual ] [ -status vol_status ... ] [ -
entry entry_date ... ] [ -access access_date ... ] [ -lock lock_id ... ]
[ [ -c ] | [ -f field ... ] [ -s sort_field ... ] [ -n n ] ] [ -max_use max_use] [ -
lock_time lock_time]
```

## Fields

The fields for display volume, are:

- **key field:** *vol\_id*
- **default fields:** *vol\_id, acs, lsm, panel, row, column, pool, status, media, type*
- **available fields:** *vol\_id; acs; lsm; panel; row; column; drive\_lsm, drive\_panel, drive (display all three to display where a cartridge is mounted); type; media; pool; label\_type; status; entry\_date; access\_date; access\_count; max\_use; lock; lock\_time; recording\_format\_family; recording\_format\_model; encrypt\_status; volsafe\_status; media\_status; warranty\_life; end\_of\_life; load\_limit\_alert*

## Options

- *vol\_id*  
displays the volume(s) selected.  
A *vol\_id* can be one or more 1-16 alphanumeric strings that specifies a tape cartridge.  
The wildcard character, \*, can be used to represent all *vol\_ids* (example: *display vol\**) or to represent a group of cartridges (example: *display vol\_id VAP\** displays all cartridges starting with VAP).  
The *vol\_id* may contain any valid volume ID characters and \*.
- *home home\_loc*  
specifies volume home location(s) from which to retrieve volume information in the following format: *acs,lsm,panel,row,column*. It cannot be used with *-drive*.  
*home\_loc* subfields can use numeric ranges. All *home\_loc* subfields must be specified. A wildcard character, \*, can be used in place of one or more *drive\_loc* subfields. Numeric range rules apply to any subfield or *drive\_loc*.
- *acs <acs\_id>*  
specifies the ACS for which volume information should be retrieved.

- `drive drive_loc`  
specifies drive location(s) from which to retrieve volume information in the following format: `acs,lsm,panel,drive`. It cannot be used with `-home`  
*drive\_loc* subfields can use numeric ranges. All *home\_loc* subfields must be specified. A wildcard character, `*`, can be used in place of one or more *drive\_loc* subfields. Numeric range rules apply to any subfield or *drive\_loc*.
- `data`  
selects data cartridges.
- `scratch`  
selects cartridges in scratch mode.
- `clean`  
selects cleaning cartridges.
- `spent_clean`  
selects cleaning cartridges reported as spent (used-up) by drive type.
- `media media_type`  
selects only cartridges with the specified media type.
- `pool pool_id`  
selects and displays cartridges from scratch pools. Cartridges retain the ID of the scratch pool even after they have changed to data cartridges.
- `standard`  
selects a tape(s) with a scannable volume ID *label\_type*. It cannot be used with `-virtual`.
- `virtual`  
selects a tape(s) without an external volume ID *label\_type* that was entered through the `venter` command. It cannot be used with `-standard`.
- `status vol_status`  
valid entries are: `dismount, eject, enter, mount, home, in_drive, move, missing, absent, ejected`.
- `entry entry_date`  
specifies the date and time the cartridge was entered into the library. It is entered and displayed in ISO standard format (`yyyy-mm-dd`).
- `access access_date`  
specifies the date and time the cartridge was last used. It is entered and displayed in ISO standard format (`yyyy-mm-dd`).
- `lock lock_id`  
specifies the lock ID for the volume.
- `max_use max_use`  
specifies the maximum number of uses for a cleaning cartridge. Non-cleaning cartridges have a `max_use` value of zero.
- `lock_time lock_time`

specifies the date and time the lock was set. It is displayed according to the system default format.

- `recording_format_family drive_family -recording_format_model drive_model`  
specifies the recording format drive family and drive model. **Example:** T10000 and T10000C

- `c`  
(count) causes the request to display only the number of objects that satisfy the *arg* and *selection* criteria. This option is mutually exclusive with the `-f field`, `-s sort_field`, and `-n n` options.

- `-f field`  
specifies the fields of information and the order in which they are to be returned for each object selected. Each display type has a list of available, key, and default fields. If `-f` is specified, the default fields are not displayed unless they are key fields or have been specified. If `-f` is not specified, then the default fields are returned. Key fields for each type are normally displayed first. However, you can change the order in which the key fields are displayed by specifying them after the `-f` option.

The list of fields is separated by spaces. The following is a list of new volume fields that are not reported by `query volume`. These statistics fields are only reported for these libraries and drives:

The following statistics fields are only reported for these libraries and drives:

Libraries:

- All SL3000s.
- SL8500s with 4.13 and later firmware.
- Tape drives:
- T9840C and T9840D with 1.42 and later drive firmware.
- T10000A and T10000B with 1.38 and later drive firmware.
- T10000C and T10000D

- `recording_format_family`  
This shows the drive family that last recorded data on this cartridge.
- `recording_format_model`  
This shows the drive model that last recorded data on this cartridge. This information is useful to:
  - avoid mounting a cartridge on a drive that cannot read the data on a cartridge (such as, A T10000A cannot read data recorded by a T10000B drive) or
  - append data to a cartridge (such as, A T10000B cannot append data to a cartridge written by a T10000A).
- `encrypt_status`  
This shows the encryption status of a cartridge, if known.
- `volsafe_status`  
This shows whether this is a `volsafe` cartridge. `Volsafe` is StorageTek's WORM (Write Once Read Many) cartridge.

- `media_status`  
When the tape drive reports a cartridge as “questionable,” it should be examined.
- `warranty_life` and `end_of_life`  
The volumes' `warranty_life` and `end_of_life` percentages are the volumes' use as a percentage of its `end_of_life` setting. These are values from 000.0% to over 100.0%. (The `warranty_life` is less than the `end_of_life`.)  
This information can be used to identify cartridges that need to be retired after migrating their data to new cartridges.
- `load_limit_alert`  
The volumes' `load_limit_alert` is a boolean value, where: True=1 and False=0. When a volume that reports an `end_of_life` percentage reaches 100%, its `load_limit_alert` is set to 1. Some drive types only report a `load_limit_alert`.
- `-s sort_field`  
sorts the objects returned by the field(s), selected in the order specified. Sorting is based on internal database values, and may not always display in alphanumeric order. For example, `media_type` is sorted by an internal numeric media type and not by the literal that is `display`.  
All returned fields for a type are valid sort fields.
- `-n n`  
specifies the maximum number of objects to be displayed.

## Examples

- To display volumes based on location:  
This example displays all volumes with home locations in panel 0,1,9.  

```
display volume * -home 0,1,9,*,*
```
- To display all 9840 data volumes.  

```
display volume * -media STK1R
```
- To display 9840 data volumes with their recording formats (if known):  

```
display volume * -media STK1R -f media recording_format_family  
recording_format_model
```
- To display a cleaning volumes by ACS and media type:  

```
display volume * -home acs,*,*,* -media media_type -f vol_id acs lsm media  
max_use access_count
```
- To display volumes by date:
  - To display all volumes entered in 2011:  

```
display volume * -entry 2011
```
  - To display all volumes entered in January, 2011  

```
display volume * -entry 2011-01
```
  - To display all volumes entered on January 27, 2011  

```
display volume * -entry 2011-01-27
```



- To display all volumes entered on January 27, 2011 between 10:00 a.m. and 11:00 a.m.  

```
display volume * -entry 2011-01-27:10
```
- To display all volumes entered between January 2011 and April 2011.  

```
display volume * -entry 2011-01-2011-04
```
- To display all volumes entered between 6:33 a.m. and 6:57:45 April 10 (for example, do this after an outage).  

```
display volume * -entry 2011-04-10:06:33:00-2011-04-10:6:57:45
```
- To display all volumes entered since February 25, 2011.  

```
display volume * -entry >2011-02-25
```
- To display T10000 data volumes sorted by `end_of_life` usage:  

```
display volume * -media T10000T1 T10000TS -f media end_of_life  
warranty_life -s end_of_life
```
- To display all volumes warranty life and end-of-life and sorted by `end_of_life` usage  

```
display volume * -f warranty_life end_of_life -s end_of_life
```
- To display spent (used-up) cleaning cartridges:  

```
display volume * -spent_clean
```

# 14

## lib\_cmd

This chapter describes the optional command-line interface (CLI), `lib_cmd`. This tool is primarily intended for use for administering logical libraries in ACSLS, but some commands may be useful as an alternative to the ACSLS GUI or to `cmd_proc` in certain specific contexts. The `lib_cmd` CLI supports eject operations, including the ability to specify long (greater than 42) volume lists.

### Introduction

ACSLs introduced the concept of logical libraries in Version 8.0. Logical Library client operation is accessed using SCSI commands through the SCSI Media Changer Emulation (SMCE) over a fibre-channel connection. Clients can `mount`, `dismount`, `enter` and `eject` tapes, and manage their own tape inventory using the SMCE interface, operating as if the library were directly connected to the client application.

Until ACSLS 8.2, logical library administrative setup functions were accomplished through the ACSLS Graphical User Interface (also introduced in ACSLS 8.0). Such tasks include creating a logical tape library with a desired number of storage cells and tape-drive slots, establishing a client initiator-target relationship, assigning physical drives to the library, and assigning tape volumes.

These administrative tasks can also be accomplished through a UNIX command-line interface (CLI). The CLI extends the usability of logical libraries, providing a faster path to critical administrative functions, and providing the means to administer logical libraries using batch processing UNIX shell scripts.

### Command

```
lib_cmd
```

### Pathname

```
$ACS_HOME/bin/lib_cmd
```

### Synopsis

```
lib_cmd
lib_cmd [-f infile]
lib_cmd assign drive drive_id lib_id
lib_cmd assign volume [vol_id | vol-range] lib_id
lib_cmd unassign volume [ vol_id | vol-range ] lib_id
lib_cmd unassign drive lib_id logical_drive_id
lib_cmd create library lib_name backing_acs cell_capacity drive_capacity
lib_cmd create mapping initiator_id target_id library_id
lib_cmd edit initiator initiator_id alias
lib_cmd edit library lib_id [-n name ]
                             [-c capacity ]
                             [-d drive_slots ]
                             [-f volume_label_format [6|8p|8s|all]]
```

```

                                [-x imp/exp_cell_count ]
lib_cmd edit target target_id alias
lib_cmd refresh initiator
lib_cmd refresh target
lib_cmd delete initiator initiator_id (y/n)
lib_cmd delete library lib_id (y/n)
lib_cmd delete mapping lib_id (y/n)
lib_cmd delete target target_id (y/n)
lib_cmd display drive [drive_id ]
                                [-p acs_id | all ]
                                [-l lib_id | all ]
                                [-t drive_type ]
                                [-u acs_id ]
lib_cmd display initiator
lib_cmd display library [ -p acs_id | all ]
                                [ -l lib_id | all ]
lib_cmd display mapping [ lib_id | all ]
lib_cmd display target
lib_cmd display volume [ vol_id ] | vol-range ]
                                [ -p acs_id | all ]
                                [ -l lib_id | all ]
                                [ -u acs_id ]
lib_cmd eject cap <cap_id> [-verbose] volume <vol_id...vol_id> | file
<path_to_volume_list>
lib_cmd vary library lib_id [online|offline|diagnostic]
lib_cmd vary drive lib_id drive_id [online|offline|diagnostic]
lib_cmd [ exit | quit | log ]

```

## Using lib\_cmd

The `lib_cmd` command-line utility is an alternative to the ACSLS GUI for managing and monitoring resources in ACSLS logical libraries. With `lib_cmd`, the ACSLS administrator can create, edit, or delete a logical library; assign or un-assign volumes or tape drives to a logical library; refresh initiators or targets; create initiator-target mappings for client communication; display the status of logical libraries; or display the status of volumes or drives within the configured physical or logical libraries.

In addition to logical library administration, `lib_cmd` offers a small set of general library commands that can be used in lieu of `cmd_proc` for displaying the contents of the library or for ejecting large amounts of volumes to a single cap (such as the SL3000 AEM).

These operations can be performed by any user in the ACSLS group, including `root`.

The `lib_cmd` can be operated in interactive mode or batch mode. Interactive mode is invoked by the command alone, without arguments.

```
lib_cmd
```

Interactive mode supports 'help' functions which list all of the possible sub-commands and arguments. Within the context of each sub-command, a list of possible arguments and options can be displayed upon request. When unsure of possible valid input strings, you can press `<Enter>` to get a list of all possible arguments for the immediate context.

No inter-activity is offered in batch mode to better support the use of this utility with user-generated scripts. In batch mode, all of the desired options and arguments must be submitted with the command. Otherwise, an error message is returned.

## Option

- `-f` (Command file input specification)

When the `-f` option is requested, the utility executes the set of commands it finds in the specified input file. The input file must contain only fully-formed requests.

## Subcommands

This section describes the sub-commands.

- `assign drive drive_id lib_id`

Assigns the specified drive to the specified library. The `drive_id` is expressed by the standard notation for drives in a physical library (`acs, lsm, panel, drive`). The argument `lib_id` is the assigned ID of the logical library. Logical library IDs begin at 1001 and increase incrementally as libraries are created.

- `unassign drive lib_id logical_drive_id`

Removes the specified logical drive from the logical library configuration. The drive is specified by its logical address (starting at address 500). The logical library ID is a four digit number (starting at 1001).

- `assign volume vol_id | vol-range lib_id`

Assigns a volume for exclusive use within a logical library. The volume is expressed by its standard six or bar-code label format. The library is expressed by its numerical logical library ID. The volume must be 'assignable'. That is, it is not already assigned, is un-owned, and is not mounted, reserved, in transit, or otherwise in use.

A range is specified by a low ACSII value string and high ASCII value string separated by a dash. When a volume range is requested, `lib_cmd` searches the backing ACS for volumes in the specified range. Any assignable volume in the range is assigned to the specified logical library.

- `unassign volume vol_id | vol-range lib_id`

Removes a volume from the logical library inventory. The volume must reside in its home cell and cannot be reserved for a pending move operation.

A range is specified by a low ACSII value string and high ASCII value string separated by a dash. When a volume range is requested, `lib_cmd` searches the specified logical library for volumes in the specified range. Any assigned volumes in the range that are not in use are unassigned from the library.

- `create library lib_name backing_acs cell_capacity drive_capacity`

Creates a new logical library. Following `create library`, four arguments are required in this order: library alias name (string); backing ACS ID (integer); number of storage cells (integer); and number of drive slots (integer).

- `create mapping initiator_id target_id library_id`

Establishes an initiator-target (client-server) relationship for the specified library.

- `edit library lib_id [-n name ] [-c capacity ] [-d drive_slots ] [-f volume_label_format [6|8p|8s|all] [-x imp/exp_cell_count ]`

Alters the configuration of an existing logical library. The specific library is expressed by its four-digit numerical ID. Any single option may be supplied and only that single field is changed using the argument supplied with the option. Options include:

- -n library alias name
- -c logical storage cell capacity
- -d number of logical drive slots
- -f volume label format
- -x number of logical import/export (CAP) cells

Volume label format may be expressed with any of the following:

- 6 six character legacy volume labels
- 8p eight character prefixed with media type code
- 8s eight character suffixed with media type code
- all accept all label formats

- refresh initiator

Retrieves all initiators currently known to ACSLS.

- refresh target

Retrieves all targets currently known to ACSLS.

- delete initiator initiator\_id (y/n)

Removes the specified initiator from the configuration. Confirmation (y) is required in order to commit the specified deletion.

- delete library lib\_id (y/n)

Removes the specified logical library from the ACSLS configuration. This creates a cascading delete, disassociating any assigned volumes and drives from the library. The library must be taken offline before the deletion is attempted. Removal fails if one or more of the logical volumes in the library are currently in use. Confirmation (y) is required in order to commit the specified deletion.

- delete mapping lib\_id (y/n)

Removes all initiator-target mappings from the specified logical library. The operation disables any current client connections. Confirmation (y) is required in order to commit the specified deletion.

- delete target target\_id (y/n)

Removes the specified target from the configuration. Confirmation (y) is required in order to commit the specified deletion.

- display drive drive\_id [-t drive\_type ] [-p acs\_id | all ] [-l lib\_id | all ] [-u acs\_id ]

Generates a summary of the specified drive, or all drives in the specified library.

The display includes the drives' physical and logical address, drive type, drive state, drive status, and if drive\_status shows in\_use, the VOLSER of the mounted cartridge is displayed.

```
----- Example -----
$ lib_cmd display drive 0,0,10,1
Physical   Logical   Drive    Drive    Drive    Mounted
Location   Address   Type     State    Status   Volume
0,0,10,1   1001:502  9840     Online   in use   RIFF21
-----
```

- If display drive all is submitted without options, the utility generates a listing of all volumes contained in each physical ACS.
- If the -t option is passed, only drives of the specified drive type are displayed.
- The -p option limits the display to drives in the specified physical ACS.
- If the -l option is included, then only the drives assigned to the specified logical library are displayed. If all is specified following the -l option, then a display is generated showing all of the drives associated with each of the configured logical libraries.
- The -u option displays only unassigned drives in the associated ACS. This option must be followed with an argument specifying the desired ACS.

- display initiator

Generates a list of all initiators identified by ACSLS. Each initiator is listed by its world-wide name and the alias name.

- display library [ -l lib\_id | all ] [ -p acs\_id | all ]

Displays the attributes of the requested library ID. Specify -l for logical libraries or '-p' for physical ACSs.

- If -l is specified, the utility generates a summary of the logical library configuration, as follows:

Numeric id, name, State (Desired state)

Backing ACS ID, State (Desired state).

Number of allocated import/export cells

Number of allocated drive slots

Number of assigned drives

Number of allocated storage cells

Number of free cells

Number of assigned volumes (of these, how many are inaccessible)

Volume Label Format

```
-----Example-----
ACS 1001 'logLib01' Offline (Desired Online)
  Backing ACS 0 Offline (Desired Online)
  2 import/export cells
  5 Tape Drive Slots
  2 Tape Drives
  999 Storage Cells
  993 Free Cells
```

```
106 Volumes (6 Inaccessible):
Volume Label Format: Six character (classic)
```

-----

Inaccessible volumes are assigned volumes that are out of reach from the logical library, for any of the following reasons:

- The volume has been ejected from the physical library.
- The volume has been marked for `eject` by the client application.
- The assigned volume resides in the physical library, but the logical library exceeds its storage capacity.
- If the argument `all` is specified in lieu of the numeric `lib_id`, then the display repeats for each configured logical library. In batch mode, passing `-1` with no argument has the same effect as if the argument `all` were used.
- When the option `-p` is specified with a specified ACS ID, the utility generates a display summarizing the library configuration for that ACS:

----- Example -----

```
$ lib_cmd display lib -p 0
Physical ACS 0 Online 56 Drives:
1 LSM:
LSM Library LSM LSM Drive Vol Free Cell
ID Type Status State Count Count Count
0,0 SL3000 Normal Online 56 62 4321
10 CAPs:
ID Mode State Status Condition Size Availabilit
y
0,0,1 Manual Online Available Operative 26 shared
0,0,2 Manual Online Available Operative 26 shared
0,0,3 Manual Online Available Operative 26 shared
0,0,4 Manual Online Available Operative 26 shared
0,0,5 Manual Online Available Operative 26 shared
0,0,6 Automatic Online Available Operative 26 shared
0,0,7 Manual Online Available Operative 26 shared
0,0,8 Manual Online Available Operative 26 shared
0,0,9 Manual Online Available Operative 26 shared
0,0,10 Manual Online Available Operative 26 shared
-----
-
```

- If the argument `all` is passed with the `-p` option, the display is repeated for each configured physical ACS.

The `all` argument is the default display for requests in batch mode. The `-p` option with no argument has the same effect as if `all` had been requested.

- `display mapping [ lib_id | all ]`

Generates a list of initiator-target mappings for each library ID (or all libraries).

- `display volume [ vol_id ] | vol-range [ -p [ acs_id | all ] [ -1 [ lib_id | all ] [ -u acs_id ]`

Generates summary information for the specified volume or volume set:

```
-----Example -----
$ lib_cmd display volume ST0212
Volume media current physical logical
```

| ID     | type  | status | location  | address  |
|--------|-------|--------|-----------|----------|
| ST0212 | STK1R | Home   | 0,0,2,6,0 | 1001:100 |

- If the volume is mounted, the logical address and physical location reflects the address of the drive to which the volume is mounted. Otherwise, the volumes' logical and physical home addresses are listed. The logical address is populated only in cases where the volume has been assigned to a logical library.
- A volume range can be specified by a low ASCII value string, and high ASCII value string separated by a dash. The display shows a status summary for each volume in the specified range.
- If `display volume all` is submitted without options, the display shows all of the volumes contained in each physical ACS.
- If the physical ACS ID is passed with the `-p` option, the list is limited to the volumes that reside in that specified physical ACS. If `all` is submitted with `-p`, the volumes contained in all physical ACSs are displayed.
- If the numeric id of a logical library (`lib_id`) is passed with the `-l` option, only the set of the volumes associated with that specific logical library are displayed. If the word `all` is passed with the `-l` option, the utility displays all of the volumes associated with each configured logical library.
- The `-u` option generates a display that is limited to unassigned volumes contained in the specified physical ACS.

```
eject cap <cap_id> [-verbose] volume <vol_id...vol_id> | file <path_to_volume_list>
```

The `lib_cmd eject` supports eject operations, including the ability to specify long (greater than 42) volume lists. Volumes can be entered as user input, or by specifying the path to a text file containing a volume list. Multiple volumes can be entered on a line, in upper or lower case.

#### Note:

`lib_cmd eject` does not provide feedback on when to empty the CAP. If you desire this feedback, use `ejecting.sh` utility command, or alternatively a simple `cmd_proc` for simple eject jobs.

- `<cap_id>`
  - It must be a specific and existing CAP in the library configuration.
  - Wild cards are not supported.
  - A list of CAPs is available.
- `[-verbose | -v]`
  - The `verbose` option specifies that all volumes be displayed.
  - The default is NOT verbose and displays up to 10 volumes plus a count.
- `<vol_id>`
  - One or more volume identifiers can be specified per line.
  - When running in interactive mode, press `<Return>` to end the list.



- <path\_to\_volume\_list>
  - Specify as relative or full path to the text file.
  - Multiple volume identifiers can be specified per line.
  - All comment lines (beginning with "#") or blank lines are ignored.

There is no known limit to the number of volumes that can be specified in either form. A single eject operation is carried out by ACSLS once all volumes have been supplied, and it keeps filling the specified CAP as needed until all possible volumes have been ejected.

The output display includes a reporting of volume counts (volumes specified, volumes ejected and/or not ejected). The `verbose` option produces output showing all volumes (by default, a 10-volume preview is produced for each list). For any volumes not ejected, a line indicating the volume identifier and the reason is displayed.

```
vary drive lib_id drive_id [online|offline|diagnostic]
```

Changes the desired state of the logical drive to the specified (online, offline, diagnostic) state.

```
vary library lib_id [online|offline|diagnostic]
```

Changes the desired state of the logical library to the specified (online, offline, diagnostic) state.

## Using lib\_cmd in Batch Mode

Using `lib_cmd` in batch mode allows you to perform more complex administrative tasks. For example, operations that assign or un-assign multiple volumes or multiple drives are often most efficiently accomplished in batch mode.

In this example, we will assign all of the 9840 drives on rail-2 of an SL8500 module to logical library 1002.

1. Using `lib_cmd`, list all of the unassigned drives in `acs-0` and, using `grep`, filter only the 9840 drives that reside on rail-2 (`acs-0, lsm-1`).

```
$ lib_cmd display drive -u 0 | grep 9840 | grep 0,1,..
```

All of the unassigned drives on rail-2 will be listed in a five-field display:

| Physical Location | Logical Address | Drive Type | Drive State | Drive Status | Mounted Volume |
|-------------------|-----------------|------------|-------------|--------------|----------------|
| 0,1,1,0           |                 | 9840       | Online      | available    |                |
| 0,1,1,1           |                 | 9840       | Online      | available    |                |
| 0,1,1,2           |                 | 9840       | Online      | available    |                |
| 0,1,1,3           |                 | 9840       | Online      | available    |                |
| 0,1,1,4           |                 | 9840       | Online      | available    |                |

For our purposes, only the drive id is required from this display, so we print only the first field, using `awk`.

```
$ lib_cmd display drive -u 0 | grep 9840|grep 0,1,..|awk '{print $1}'
0,1,1,0
0,1,1,1
0,1,1,2
0,1,1,3
0,1,1,4
```

2. Within the same `awk` print statement, we can add text to construct the necessary commands that assign the drives to logical library 1002.

```
$ lib_cmd display drive -u 0 | grep 9840|grep 0,1,..|awk '{print "assign drive
"$1"1002}'
assign drive 0,1,1,0 1002
assign drive 0,1,1,1 1002
assign drive 0,1,1,2 1002
assign drive 0,1,1,3 1002
assign drive 0,1,1,4 1002
```

3. Satisfied with the command text, we can now direct the output to a text file:

```
$ lib_cmd display drive -u 0 | grep 9840 | grep 0,1,.. | awk '{print "assign
drive "$1" 1002"}' > /tmp/assignDr
```

4. Finally, we can edit any changes to the command file, then execute the file using `lib_cmd`:

```
$ lib_cmd -f ./tmp./assignDr
--ACSL 8.2.0--
Copyright (c) 2012 Oracle and/or its affiliates. All rights reserved.
Drive 0,1,1,0 now assigned to logLib02 at 1002:500
Drive 0,1,1,1 now assigned to logLib02 at 1002:501
Drive 0,1,1,2 now assigned to logLib02 at 1002:502
Drive 0,1,1,3 now assigned to logLib02 at 1002:503
Drive 0,1,1,4 now assigned to logLib02 at 1002:504
```

5. Using the `lib_cmd display` command, we confirm the assignment:

```
$ lib_cmd display drives -l 1002
ACS 1002 LOG LIB02:20 Drive Slots 5 Drives
Physical Logical Drive Drive Drive Mounted
Location Address Type State Status Volume
0,1,1,0 1002:500 9840 Online available
0,1,1,1 1002:501 9840 Online available
0,1,1,2 1002:502 9840 Online available
0,1,1,3 1002:503 9840 Online available
0,1,1,4 1002:504 9840 Online available
```

# A

## ACSL S Backup and Recovery Tools

This appendix:

- Outlines and explains each utility, what they are used for, and why they are important.
- Provides a high level view of disaster recovery scenarios.

### ACSL S Backup Tools

ACSL S offers three robust and distinctly different methods for backing up both its database and ACSL S control files. Each utility performs different functions, and all methods play an important role in a complete disaster recovery plan.

### Automatic Backups

ACSL S provides automated database protection services. These automated protection services safeguard the daily operation of the ACSL S database against changes that can produce either unintended consequences, or from database corruption.

As a result of these automated backup protection services, you have the ability to restore your database back to any backup time from the present to the end of your retention period. The restore tools are discussed later in this appendix.

This section discusses the automated backup methods and why they are used.

- **ACSL S default backup directory**  
During the initial installation of ACSL S, you were asked to supply the name of the directory to use for backups (`/export/backup` by default). It is in this directory that backup activity occurs.
- **A complete database backup is performed and placed in the directory, using a date naming convention:**  
`/export/backup/yyyy-mm-dd-hh:mm:ss.tar`.

The time at which the daily backup is performed can be modified by changing the "Automatic Backup Variables" within `acsss_config`.

Refer to [Setting Variables that Control ACSL S Behavior](#) for information on changing default backup behavior.

- **Database retention period**  
Another configurable parameter within ACSL S that affects automatic backups is the database retention period. This is defined as the amount of time ACSL S retains the backups.

The default for the retention period is 8 days.

Refer to [Setting Variables that Control ACSL S Behavior](#) for information on changing default backup behavior.

The retention period can also be modified by using `acsss_config`.

## Manual Backups

ACSL S provides a utility called `bdb.acsss` that backs up the critical ACSLS data using the command line. This is also the method used to restore the ACSLS database where the environment consists of the same or identical hardware, OS level, and ACSLS version. Refer to [bdb.acsss](#).

Used without any command line options, `bdb.acsss` provides the ability to create a database backup and store it in the default backup directory. All critical ACSLS database and ACSLS control files are backed up to a single file. This file can then be used to restore ACSLS to its previous state on the same or identical hardware for scenarios such as an internal disk or motherboard failure.

The `rdb_acsss` utility allows a `-f` option that can be used to either specify a file and location (`rdb.acsss -f /path/my_file`) or a tape device (`-f /dev/rmt/0mn`). When using a tape device, you do not provide a file name on the tape device.

## Manual Database Exports

ACSL S provides a utility called `db_export.sh` to export the ACSLS database, ACSLS control files, and any customized dynamic variables. The `db_export.sh` utility is responsible for dumping the ACSLS database to comma separated ACSII files, making a copy of the ACSLS control files, and making a copy of the dynamic variables. This is the method used to migrate to newer versions of ACSLS and is not recommended for daily backup operations because both ACSLS and the database must be down before performing the export.

The `db_export.sh` command line utility is the preferred method for migrating the database between different levels of server hardware, OS versions, and different releases of ACSLS. Without options, it can be used with the local default tape device such as `/dev/0mn`. This tape can then be moved to any location, and ACSLS and its associated ACSLS control files can be restored into any OS version or level of ACSLS.

 **Note:**

Although any tape device can be selected, a *no-rewind* device should be used. The `db_export` utility creates two files. If a rewind device is selected, the first file (datafiles) would be overwritten when the second file is created.

As in the `bdb.acsss` utility, the `-f` option can be used to specify a tape device other than the system default. Simply execute `db_export.sh /dev/0mn` or any attached tape device to use this option.

The `-f` option also allows the database to be exported to the named file. When using this method, you will notice that two files are created - one file that you named and also another file with a `.misc` extension. Both files must be transferred to the server where the import will take place to ensure a successful import.

When executing the `db_export.sh` utility either with the `-f` option or without, you will be prompted to choose the version of ACSLS to which you are exporting.

The menu selections in `db_export.sh` are:

```
1: ACSLS 7.3
2: ACSLS 8.0, 8.0.1, 8.0.2, 8.1
3: ACSLS 8.2 or 8.3
4: ACSLS 8.4
5: ACSLS 8.5
E: Exit
Please select by number (or E to exit):
```

## ACSL Recovery Tools

ACSL uses two different recovery tools to restore all backups and exports. Both of the tools offer a menu driven user interface and easily selectable options. The two utilities are:

- `rdb.acsss` - the recovery tool for both automated and manual backups
- `db_import.sh` - restores an exported database and/or ACSL control files from the same version of ACSL, a different version of ACSL, or from a different hardware platform. This option also allows the recovery of customized dynamic variables.

### Using `rdb.acsss`

The `rdb.acsss` utility restores the ACSL database and the ACSL Control files using a backup created by either the automatic backup function or the `bdb.acsss` utility. The ACSL Control files are located in `$ACS_HOME/data`, and define several different environmental variables for ACSL. They specify Access Control settings, scratch preferences, Extended Store LSMs, custom `volrpt` settings, and volume attributes (for `watch_vols` utility), and so forth.

See [rdb.acsss](#) for options and procedures.

### Using `db_import.sh`

ACSL provides a `db_import.sh` utility to restore an exported database from the same version of ACSL, a different version of ACSL, or it could even be from a different hardware platform. Like `rdb.acsss`, it offers an easy to read menu driven user interface, allowing you to select the task that you want to perform.

The `db_import.sh` utility can work without options, or you can supply the `-f` option with a path and file name as an argument. Executing `db_import.sh` from the command line without any options causes the utility to look for the exported database on the local tape device. It first checks for the existence of the exported database, verifies that it is a valid database export file, and then displays a menu with four options.

#### Note:

You can also supply an `-f` option with a tape device (`-f /dev/rmt/0mn`) for a non-default device. Although you can supply any valid tape device, it is a requirement that you supply a no-rewind device. The `db_import.sh` utility uses two files - one for data and one for control files. If you use a rewind device, after data files are recovered, the tape would be rewound and the control files would fail.

If you provide the `-f` option with a path and file name, `db_import.sh` uses the supplied file name as the exported database file. As with the local tape device, it first checks to see if the

file exists, and then validates that the supplied file name is an exported database file. If the supplied file is a valid export, it displays a menu. The menu options, are:

- Option 1 - Import database tables, control files, and dynamic variables for the exported file.  
This option brings in the library database plus all of the customized updates that were preserved from the exported version.
- Option 2 - Import only database tables from the exported file.  
This option brings in the complete library configuration and volume data set, but does not apply any system customizations that were made in the exported version.
- Option 3 - Import only control files from the exported file.  
This option does not alter the current library database and it brings in only the customizations that had been exported from the previous version.
- Option 4 - Merge customized dynamic variables from the exported file.  
This option merges any customized settings from the exported version with the current version. See [Setting Variables that Control ACSLS Behavior](#).

## Disaster Scenarios

This section discusses disaster scenarios.

### Database becomes corrupted

1. As user `acsss`, stop ACSLS before running the recovery

```
$ acsss db
$ rdb.acsss
```
2. Select option 2. Refer to [rdb.acsss](#).
3. When recovery is complete, start ACSLS: `acsss enable`.

### Ran `acsss_config` against the wrong library

1. Select option 2. See [rdb.acsss](#).
2. Start ACSLS and test according to database backup and restore procedures.

### Server failure – rebuilding the same server with new hardware

1. Install the operating system.
2. Configure the new server and OS with the settings from the previous server.
3. Install ACSLS.
4. Insert the backup tape or FTP backup file onto the server.
5. Start the `rdb.acsss` utility.
6. Select option 2. Refer to [rdb.acsss](#).
7. Exit `rdb.acsss`.
8. Start ACSLS and test according to database backup and restore procedures.

## Server failure – rebuilding another ACSLS server with new hardware

1. Install the operating system.
2. Install ACSLS.
3. Place the ACSLS server-to-server backup files in the proper location.
4. Enter `rdb.acsss`. Refer to [rdb.acsss](#) .
5. Select option 3.
6. When the recovery utility completes, start ACSLS and test according to database backup and restore procedures.

# B

## Enterprise Library Connection Options

This chapter contains the following:

- [Overview](#)
- [Dual TCP/IP Support](#)
- [Multi TCP/IP Support](#)
- [Redundant Electronics](#)

### Overview

There are multiple options for connecting ACSLS to SL8500 and SL3000 libraries. These options can be used independently or together, for communication between ACSLS and an SL8500, or SL3000.

In a string of connected SL8500s, you can implement Dual TCP/IP and/or Multi-library TCP/IP, and/or Redundant Electronics.

In an SL3000 or SL8500, you can implement Dual TCP/IP and/or Redundant Electronics (RE). You can connect to an SL3000 or SL8500 through IPv4.

The following summarizes the connections options:

- **Dual TCP/IP**

Dual TCP/IP provides two separate and independent TCP/IP connections between ACSLS and a Library Controller card. If one of these communication paths fail, ACSLS automatically uses the second path for communication.

To implement Dual TCP/IP support, routing tables on both the ACSLS server and the library must be defined and managed using the "route" command. These routing tables force communication between a pair of ports on the ACSLS server and the library to use a defined network communication path.

Both the SL8500 and the SL3000 support Dual TCP/IP communication with the library.
- **Multi TCP/IP Support**

Multi TCP/IP support allows the ACSLS server to connect to multiple SL8500 libraries in a string of connected SL8500s. If communication with one library fails, ACSLS automatically sends library communication to the connections with the other libraries. The libraries automatically forward the messages to the other libraries.

Configuring and managing Multi TCP/IP communication is simpler than Dual TCP/IP because routing tables do not need to be defined on the ACSLS server or the SL8500 libraries. However, Multi TCP/IP requires a string of connected SL8500 libraries. It does not apply to single, stand-alone SL8500 or SL3000 libraries.
- **Redundant Electronics (RE)**

RE uses a redundant set of Library Controller cards. At any given time, one set is active and the other set is standby. The active Library Controller can failover to the standby in



response to a command from ACSLS or the SL Console. Automatic failover can be initiated by the library if a library card fails.

RE enables minimally disruptive library firmware (microcode) downloads. Within a string of connected SL8500s, RE can be implemented on a per library basis. You can implement RE in any or all libraries within a complex.

To support RE in the library, ACSLS 7.3.1 or 8.0.2 or later is required.

## Displaying the status of ACSLS Communication with Libraries

Use the `query lmu` command to view and monitor the status of ACSLS communication with the libraries it manages. The `query lmu` command also shows the status of ACSs and port connections to libraries.

## Dual TCP/IP Support

Dual TCP/IP is an option that can be purchased for the SL8500 and SL3000 libraries (herein known as the library). It provides two TCP/IP connections to the library. However, you can continue to use the library with only one of the two connections operational.

The purpose of dual TCP/IP is to automatically recognize and avoid a failing communication path. Since this is automated, there is no need for you to manually switch from an inoperative connection.

To use dual TCP/IP support on the library, the routing tables on both of the ACSLS server and the library must be managed using the "route" command. This forces a route to the defined network interfaces on the library which in essence, creates a one-to-one relationship between interfaces. The Customer Systems Administrator (CSA) changes the routing tables on the ACS server and the Customer Systems Engineer (CSE) updates the routing tables on the library. For further information on the UNIX "route" command, refer to the man pages on your ACSLS server.

## Requirements

- Coordinate with both your system administrator and network administrator to understand your current network environment, and to identify all necessary IP addresses in advance.
- Coordinate with your system administrator to either configure your network interface, or to validate that it is configured properly.

## Configuration

It is recommended that ACSLS keep two connections to the library open since ACSLS uses all active connections. If one connection is inoperative, ACSLS uses the remaining operative connection while continuing to try to re-establish communication on the failing connection.

The preferred configuration for dual TCP/IP implementations would be two network interfaces on two separate subnets for the ACSLS server, as shown in Scenario 1. This provides maximum throughput and minimum resource contention, with regard to network communications while adding a second physical connection, improving reliability.

To configure two TCP/IP connections to a single library, use the `acsss_config` utility or Dynamic Configuration (`config`). Enter the number (2) of connections that there are to the library and the IP addresses of the network devices. The SL3000 supports IPv4 connections.

The following scenarios provide examples for configuring the ACSLS server. For instructions on configuring the library dual TCP/IP feature, refer to the appropriate *library System Dual TCP/IP Feature* document.

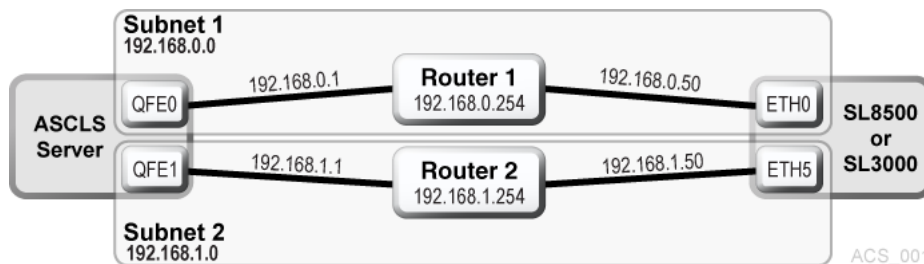
The following scenarios use private subnet IP addresses and will not be the same in your environment. These scenarios assume that your network devices have been configured and are functioning properly.

## Scenario 1 - Preferred Configuration

Scenario 1 is the preferred configuration for the dual TCP/IP feature.

In this configuration, the ACSLS server contains two network interfaces that reside on two separate subnets. The SL8500 or SL3000 has two network interfaces on the same two subnets as the ACSLS server.

**Figure B-1 Preferred Configuration**



In this scenario, the libraries use a one-to-one relationship with the network interfaces on the ACSLS server in which the:

- `qfe0` interface on the ACSLS server only communicates with the `eth0` interface on the SL8500 or SL3000.
- `qfe1` interface on ACSLS only communicates with `eth5` on the SL8500 or SL3000.

Using the UNIX “`route`” commands, you force this relationship.

- For Solaris: as user `root`, type the following commands:

```
route -p add 7.0.50 -ifp qfe0 192.168.0.254
route -p add 192.168.1.50 -ifp qfe1 192.168.1.254
```

The first route command routes any communication with `192.168.0.50` to go through `qfe0` on the ACSLS server and then go through Router 1.

The second command routes any communication with `192.168.1.50` to go through `qfe1` on the ACSLS server and then go through Router 2.

You can validate that the routes are in the routing table by typing:

```
# netstat -r
```

The first two entries are the ones that were just added. All communication with 192.168.0.50 will go through QFE0, and communication with 192.168.1.50 will go through QFE1.

**Remember:** Configure the libraries' routing tables according to the instructions in the *StorageTek SL8500 Modular Library System Dual TCP/IP Feature* document.

**Example B-1 IPv4 Routing Table**

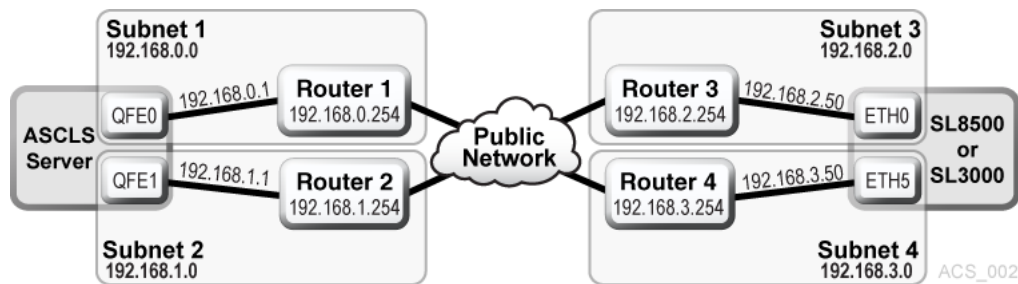
| Destination            | Gateway       | Flags | Ref | Use | Interface |
|------------------------|---------------|-------|-----|-----|-----------|
| 192.168.0.50           | 192.168.0.254 | UGH   | 1   | 0   | qfe0      |
| 192.168.1.50           | 192.168.1.254 | UGH   | 1   | 0   | qfe1      |
| 192.168.0.0            | 192.168.0.1   | U     | 1   | 7   | qfe0      |
| 192.168.1.0            | 192.168.1.1   | U     | 1   | 0   | qfe1      |
| BASE-ADDRESS.MCAST.NET | 192.168.0.1   | U     | 1   | 0   | qfe0      |
| default                | 192.168.0.254 | UG    | 1   | 33  |           |
| localhost              | localhost     | UH    | 4   | 77  | lo0       |

## Scenario 2

Scenario 2 shows:

- ACSLS server with two interfaces on separate subnets from the library
- SL8500 or SL3000 library with two network interfaces on separate subnets from ACSLS
- Both ACSLS and SL8500 or SL3000 using a public network

**Figure B-2 ACSLS and SL8500 or SL3000 Using a Public Network**



Using the UNIX "route" commands, you force this relationship.

- For Solaris: as user root, type the following commands:

```
#route add 192.168.2.50 -ifp qfe0 192.168.0.254
#route add 192.168.2.50 -ifp qfe0 192.168.0.254
```

The default routes for the ACSLS remain the same. The routes within the subnets will know how to route communication to the libraries through the public LAN and you are still forcing the one to one relationship with the interfaces. Again, this is seen using the following command:

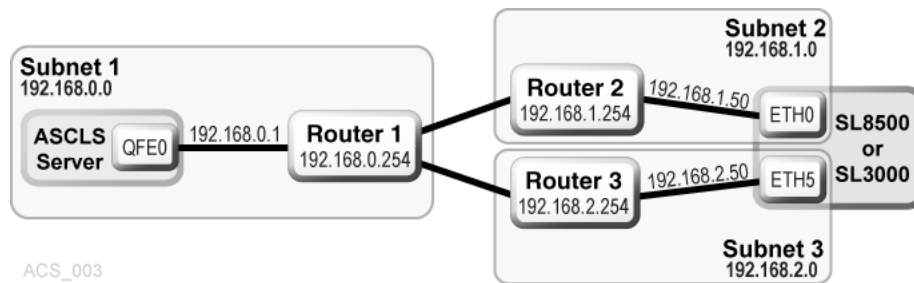
```
# netstat -r
```

**Remember:** Configure the libraries' routing tables according to the instructions in the *StorageTek SL8500 or SL3000 Modular Library System Dual TCP/IP Feature* document.

## Scenario 3

In this scenario, there is one ACSLS server with one network interface on a separate subnet. The SL8500 or SL3000 library has two network interfaces on two subnets that are separate from the ACSLS server.

**Figure B-3 SL8500 or SL3000 with Two Network Interfaces**



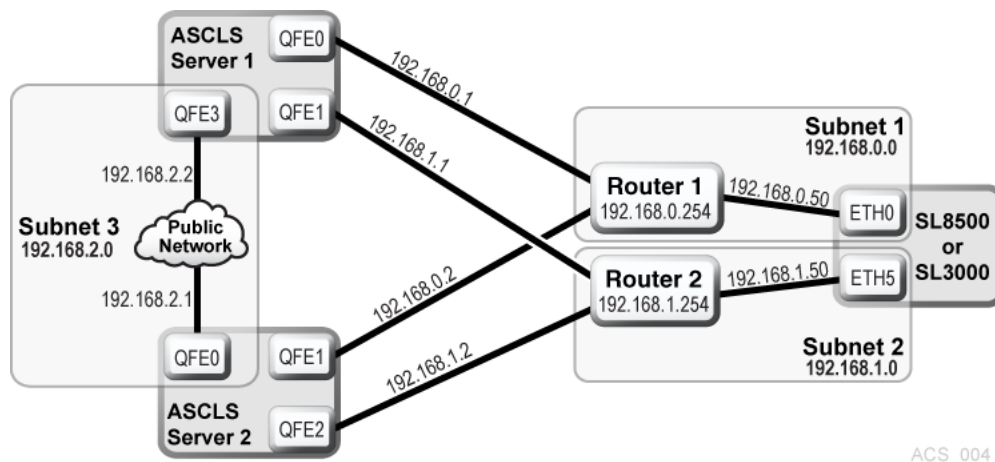
**Remember:** Configure the library routing tables according to the instructions in the StorageTek SL8500 or SL3000 Modular Library System Dual TCP/ IP Feature document.

## Scenario 4

Scenario 4 shows:

- Two Highly Available (ACSLS HA) servers, both with three network interfaces, two separate private subnets with the SL8500 or SL3000, and a third public network.
- One SL8500 or SL3000 library with two network interfaces on the same two private subnets as the ACSLS servers.

**Figure B-4 ACSLS HA**



In this scenario, ACSLS HA uses two different servers with each using different network interfaces. This means that custom route entries must be added to both ACSLS servers.

For the Solaris user:

- On ACSLS server 1, you would type:

```
route add 192.168.0.50 -ifp qfe0 192.168.0.254
```

```
route add 192.168.1.50 -ifp qfe1 192.168.1.254
```

- On ACSLS Server 2, you would type:

```
route add 192.168.0.50 -ifp qfe1 192.168.0.254
```

```
route add 192.168.1.50 -ifp qfe2 192.168.1.254
```

You must add the IP addresses for both servers to the libraries' configuration. Refer to the *StorageTek SL8500 or SL3000 Modular Library System Dual TCP/IP Feature* document.

It is important that you separate the libraries' network interfaces over two different subnets when on ACSLS HA. The purpose of a Highly Available environment is to build in redundancy and eliminate single points of failure.

**Remember:** Configure the libraries' routing tables according to the instructions in the StorageTek SL8500 or SL3000 Modular Library System Dual TCP/ IP Feature document

## Retaining Customized Routing Table Entries after a Reboot

Any customized routing table entries are lost after a system reboot. This is the nature of system routing tables and is expected behavior.

In order to support the Dual TCP/IP feature on the SL8500 or SL3000, it is necessary to add custom entries to the routing tables on the ACSLS server. When the ACSLS server is rebooted, all routing table entries are flushed and any necessary routes to the libraries are removed. Since this is the nature of the operating system, there are a couple of different ways to handle this situation.

## Creating Scripts

You can create scripts to add custom routes to be initialized at boot time. See [Adding Custom Route To Be Initialized At Boot Time](#) for procedures.

These scripts can then be placed in the `rc` directory structure for automatic execution at boot time. Refer to your system documentation for details on the best way to implement this.

Use the ACSLS startup scripts to add your custom routing entries at boot up time. The startup scripts check for a file that contains customized route table entries. Any entries found are added to the routing table automatically using the UNIX `route` command. For standalone ACSLS installations, this is a desirable method to maintain route entries that are necessary for library support.

**Important:** This solution **will not work** if the ACSLS installation is a Highly Available ACSLS (ACSLs HA) environment. In this case, you must use the first method to maintain routing tables.

ACSLs HA handles system initialization differently than a standalone ACSLS server because it relies on Solaris Cluster to manage its clustered resources, which means ACSLS cannot be automatically started by way of the system RC mechanism at boot

time. This is handled strictly by the Solaris Cluster agents with the S87ACSLs startup scripts never being used. Add a script with the appropriate "route add" commands and locate it within the `/etc/rc2.d` directory structure. It is highly recommended that anyone with an ACSLS HA environment engage Oracle Advanced Customer Support – preferably the same consultant that originally installed the ACSLS HA system.

## Adding Custom Route To Be Initialized At Boot Time

To add custom routing entries:

1. cd to the following directory:

```
$ACS_HOME/data/external/ custom_routing.
```

This directory contains the template file `custom_routing_tables.tpl`.

2. Copy this file and change the file name to `custom_routing_tables`.

```
# cp custom_routing_tables.tpl custom_routing_tables
```

3. Edit (vi) the `custom_routing_tables` file and add your entries.

The file contains three fields.

- The IP address for the SL8500 or SL3000.
  - The name of the interface on the ACSLS server that you want to establish the one-to-one relationship.
  - The IP address of the default route for your subnet.
4. Follow the instructions in the `custom_routing_tables` comment section for the format.

 **Note:**

Make sure that there are no blank lines.

When your server reboots, ACSLS is automatically initialized, and your custom routes are added to the routing table.

5. Verify all routes in the routing table with the following command:

```
# netstat -r
```

Refer to your UNIX man pages for complete documentation on both the `route` and `netstat` commands.

## Removing routing commands

Use the `route` command to remove any special routing commands that have been added erroneously or are no longer needed to the earlier configuration.

Example: As user `root`, type the following commands:

```
# route delete 192.168.0.50 192.168.0.254
```

This says to remove the route to `192.168.0.50` (the SL8500 or SL3000) using the default route of `192.168.0.254`. The route is then removed.

## Multi TCP/IP Support

When SL8500 3.97 or higher firmware is installed, ACSLS can connect to more than one SL8500 in an ACS (library complex).

ACSLs supports up to fifteen connections to an ACS. For example, this can be: fifteen connections to four SL8500s; two connections to each of two SL8500s; two connections to one SL8500 and two connections to two other SL8500s; three connections to two or three libraries, and so on.

When ACSLS is connected to more than one library, the connections should be through different subnets for redundancy. If one subnet fails, communication between ACSLS and the library still continues over the other subnet(s).

When ACSLS has two connections to one SL8500 HBC card, you should configure the SL8500 and ACSLS server routing tables as described in [Dual TCP/IP Support](#). If you have only a single connection between the ACSLS server and each SL8500 HBC card, configuring the ACSLS and SL8500 routing tables is **not** necessary.

To optimize library performance, and minimize inter-library communication among SL8500s, define your first connection (port 0) to the library with the most activity.

Configuring and managing Multi TCP/IP communication is simpler than Dual TCP/IP because routing tables do not need to be defined on the ACSLS server or the SL8500 libraries. However, Multi TCP/IP requires a string of connected SL8500 libraries. It does not apply to single, stand-alone, SL8500 or SL3000 libraries.

For more information, refer to the *StorageTek SL8500 Modular Library System Technical Brief - Host to Library Communications*.

The following figures show an ACSLS with Multi TCP/IP configuration and an ACSLS with Multi TCP/IP and Dual TCP/IP configuration.

**Figure B-5 ACSLS with Multi TCP/IP**

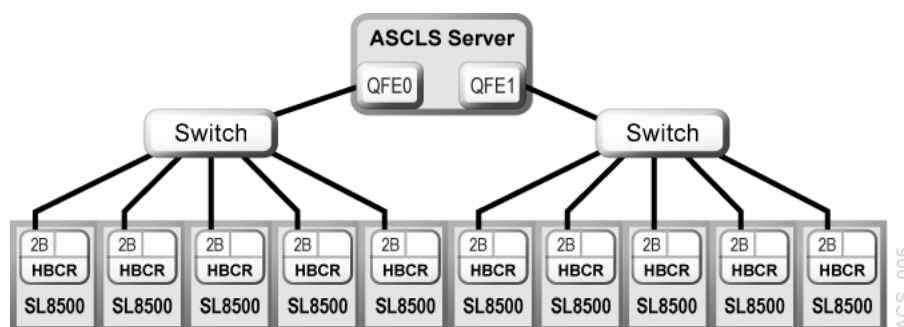
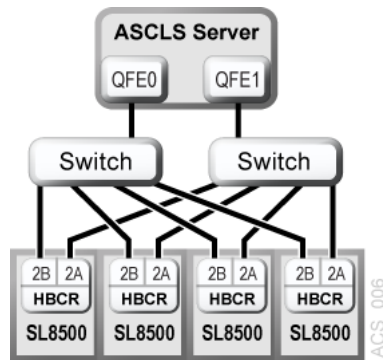


Figure B-6 ACSLS with Multi TCP/IP and Dual TCP/IP



## Redundant Electronics

The optional SL8500 or SL3000 Redundant Electronics (RE) feature provides failover protection in enterprise libraries. If the library controller experiences errors, it automatically switches operations to an alternate library controller, with minimal disruption to library and host operations. This allows your Oracle support representative to replace the faulty card while the library continues normal operations.

RE also provides minimal disruption of library operations during firmware upgrades.



### Note:

The libraries offer redundancy in a variety of components, including robots and power systems. The term “Redundant Electronics” refers specifically to redundancy in the library and drive controller components.

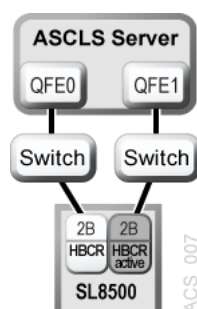
RE requires all of the following hardware components:

- Active library controller (HBC or HBCR) paired with the active drive controller (HBT)
- Standby HBC or HBCR paired with the standby HBT
- Other redundant components

For more information, refer to the *StorageTek SL8500 or SL3000 User's Guide*.

The following figure shows ACSLS with RE in a single library.

Figure B-7 ACSLS with RE



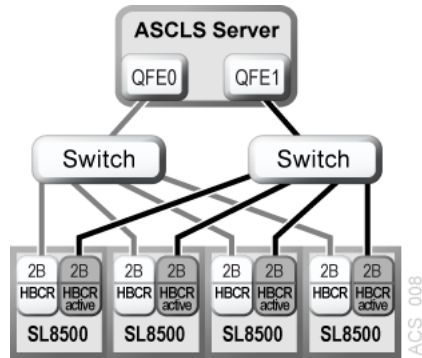


## ACSL S Support for RE

ACSL S handles a mix of active and standby SL8500 Library Controller (LC) cards within a single library complex (an ACS of libraries connected by pass-thru).

As shown in the following figure, either of the HBCR cards in each SL8500 can be the active controller card.

**Figure B-8 ACSLS with RE and Multi TCP/IP**



Each library in a string of connected SL8500s can now have its own pair of redundant Library Controllers. In a library complex, some libraries can have a pair of library controller cards, with RE enabled, while other libraries only have a single library controller. ACSLS should be able to communicate with all of the active LCs at the same time.

The following figures show ACSLS support for RE with Dual TCP/IP or with Dual and Multi TCP/IP.

**Figure B-9 ACSLS with RE and Dual TCP/IP**

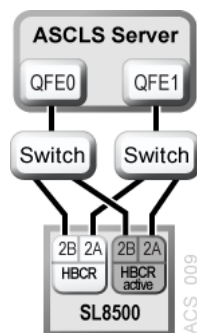
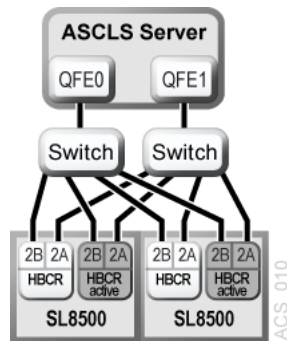


Figure B-10 RE with Dual TCP/IP and Multi TCP/IP



## Query and Retry of Mounts and Dismounts

To support RE, ACSLS implemented the Query and Retry of mounts and dismounts during temporary library and drive outages. For more information, see [Queue and Retry Mounts and Dismounts when Library is Temporarily Unavailable](#).

## switch lmu for Only a Single Library

The `switch lmu` command can be used to force a switch between library controllers in an SL3000 or a single SL8500 library. The `switch lmu` command cannot be used to switch one SL8500 that is connected to other SL8500s in a library complex.

# C

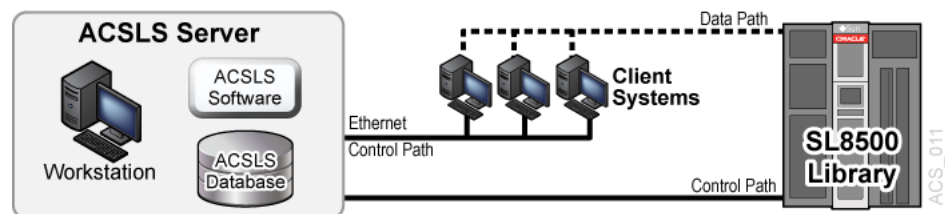
## ACSLs Support of the SL8500

This appendix discusses the following topics:

- [Connecting to Multiple SL8500s Using Multi TCP/IP](#)
- [Understanding SL8500 Internal Addresses and ACSLS Addresses](#)
- [Using SL Console to Translate Addresses](#)
- [Moving Cartridges Before Removing Cells from a Partition](#)
- [SL8500 CAPs](#)
- [Minimizing Elevator and PTP Activity](#)
- [Configuring Tape Drives to Support Your Workloads](#)
- [Managing Cartridge Locations](#)
- [Finding Missing Cartridges](#)
- [Varying the SL8500 Offline](#)
- [Using the Dynamic Configuration \(config\) utility](#)
- [Expanding the SL8500](#)
- [Connecting SL8500s with Pass-Thru-Ports](#)

The following figure shows the SL8500 library with an ACSLS server:

**Figure C-1 SL8500 with an ACSLS Server Before Configuring ACSLS for the SL8500**



The SL8500 library uses TCP/IP protocol over an Ethernet physical interface to manage and communicate with the host and ACSLS. This interface enables ACSLS to connect to and communicate with the SL8500. Before you configure ACSLS for the SL8500:

- Connect one or more SL8500s to ACSLS
- Verify that all the components of the SL8500 are operational.

ACSLs builds its library configuration from the information reported by the library. If SL8500 components are not operational, the library information may not be reported to ACSLS, and the ACSLS configuration of the SL8500 will be incomplete.

 **Note:**

If the components like drives or CAPs are not operational, it is easy to use Dynamic Configuration (`config acs`, `config lsm`, or `config drives`) to add or update them while ACSLS is running and the library is online.

## Connecting to Multiple SL8500s Using Multi TCP/IP

When SL8500 3.97 or higher firmware is installed, ACSLS can connect to more than one SL8500 in an ACS (library complex).

ACSLs supports up to fifteen connections to an ACS. For example, this can be: fifteen connections to four SL8500s; two connections to each of two SL8500s; two connections to one SL8500 and two connections to two other SL8500s; three connections to two or three libraries, and so on.

When ACSLS is connected to more than one library, the connections should be through different subnets for redundancy. If one subnet fails, communication between ACSLS and the library still continues over the other subnet(s).

When ACSLS has two connections to one SL8500 HBC card, configure the SL8500 and ACSLS server routing tables, as described in [Overview](#). If you have only a single connection between the ACSLS server and each SL8500 library, configuring the ACSLS and SL8500 routing tables is **not** necessary.

To optimize library performance and minimize inter-library communication among SL8500s, connect to the libraries with the most activity. Make the first connection that you specify in `acsss_config` or `config acs new` to the SL8500 with the most activity.

For more information, refer to the *SL8500 Modular Library System Technical Brief - Host to Library Communications*.

## Verifying that all SL8500 Components are Operational

To verify that all the components of the SL8500 are operational:

1. Logon to the StorageTek Library Console (SL Console).  
You can use either the console on the SL8500 or a remote Library Console.
2. Select `Tools -> System Detail`.
  - All SL8500 components should be green.  
**Exception:** Drives that are yellow can be configured now or later, using dynamic configuration. See [bdb.acsss](#).
  - Missing components can be added using the Dynamic Configuration (`config acs` or `config lsm`) utility.
  - **IMPORTANT:** Before configuring the SL8500, the elevators (Elevator Folder) **must** be green. If the elevators are not green, **do not** configure the SL8500 to ACSLS. The elevators are the logical pass-thru-ports (PTPs). Without PTPs, ACSLS will not know that the SL8500 rails are connected.

3. Once the SL8500 components are operational, configure SL8500 to ACSLS, as discussed in [Setting CSI Tuning Variables](#) or under the [The acsss Macro](#).

## Understanding SL8500 Internal Addresses and ACSLS Addresses

There are differences in the internal addresses of the SL8500 and other libraries supported by ACSLS and HSC.

- The SL8500 is *one's-based* (1) and uses negative numbers.
- Other libraries use a *zero-based* (0) with *no* negative numbers.
- The SL8500 uses *five* parameters: library, rail, column, side, and row.
- Legacy StorageTek libraries (such as 9310) use: ACS, LSM, panel, row, and column (HLI-PRC).

**Table C-1 SL8500 Addressing Descriptions**

| HLI-PRC                                  | SL8500                                | Description                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------------------------------|---------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ACS                                      | Library                               | Number of the specific SL8500 library in a library complex. An ACS is a SL8500 library complex. There can be multiple SL8500s in a library complex.                                                                                                                                                                                                                                                                    |
| LSM<br>LSM 0-3                           | Rail<br>Rail 1-4                      | The SL8500 library has four rails that the HandBots travel, which are numbered from top to bottom 1–4 (one's-based).<br><br>ACSLs considers each rail to be a separate LSM, numbered from top to bottom 0–3 (zero-based).                                                                                                                                                                                              |
| Panel<br>Panel 0<br>Panel 1<br>Panel 2-n | Column<br>CAP Drives<br>Storage slots | Columns indicate the horizontal location in the library. As viewed from the front of the library column and panel numbers start at the center of the drive panel (1) and sweep forward with increasing numbers. (The SL8500 does not use panels as an address.)<br><br>An HLI panel spans across the width of the library to include both sides (left and right) and both walls (inner and outer) for <i>each</i> LSM. |
| -                                        | Side                                  | Wall location:<br>Outer wall<br>Inner wall<br><br>HandBot number:<br>Left (-)<br>Right (+)                                                                                                                                                                                                                                                                                                                             |
| Row                                      | Row                                   | Rows indicate the vertical location of a tape cartridge and are numbered from the top—down.                                                                                                                                                                                                                                                                                                                            |

**Table C-1 (Cont.) SL8500 Addressing Descriptions**

| HLI-PRC | SL8500 | Description                                                                                                                                                                                                                                                                                                                                                                                                                           |
|---------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Column  | -      | <p>Rows for the HLI address are:<br/>Storage panels start at 2 with Column 0 = left and Column 1 = right<br/>Rows 0–12 outer walls<br/>Rows 13–26 inner walls<br/>Each column in a normal storage panel has 27 rows.<br/>For a total capacity of 54 cartridges per panel.</p> <p>Rows for the SL8500 address are:<br/>Storage slots start at Column -3 = left Column +3 = right<br/>Rows 1–13 outer wall<br/>Rows 1–14 inner wall</p> |

- Zero-based numbering (as with HLI) starts numbering at 0.
- One-based numbering (as with the SL8500) starts numbering at 1.
- This is an important difference in the numbering sequences between software (ACSLs or HSC) and hardware (physical SL8500 addresses)

## Using SL Console to Translate Addresses

Use the SL Console Search utility to translate between the SL8500 internal address and the ACSLS or HSC panel, row and column. To locate a cartridge:

1. Log in to the SL Console.
2. Select Tools > Diagnostics > Search.
3. Select Location.
4. Select one of the following operations in the `Location` field:

|                         |                                                                                               |
|-------------------------|-----------------------------------------------------------------------------------------------|
| <code>contains</code>   | Example: 1,1,-9 lists the contents in Library 1, Rail 1, Column -9 for all rows on both sides |
| <code>endsWith</code>   | Example: 1,5 lists the slot contents for all rails and columns for Side 1, Row 5              |
| <code>equals</code>     | Example: 1,1,-9,1,1 lists the contents in that specific location (L,R,C,S,W)                  |
| <code>startsWith</code> | Example: 1, 3 lists the slot contents for all columns, sides, and rows in Library 1, Rail 3   |

5. Select one of the following from the `Requestor` pull-down menu.
  - `default`  
The physical location inside the library (cell, drive, CAP).  
If you know the physical location (the internal address), and need to find the HLI-PRC address, enter that address in the `location` and pick `default` as the requester.
  - `hli#`

This selects the HLI-PRC address of the cartridge from the library management software where # is one of the following:

- hli0 for a non-partitioned.
- hli1-8 for a partitioned library, where the number is the partition number.

This option displays both the Internal Address and the hli# Requester.

6. Click the `Search` button in the top right corner of the SL Console.

The search result lists the location by slot-type (cell, drive, or CAP).

7. Click the `Details (...)` field.

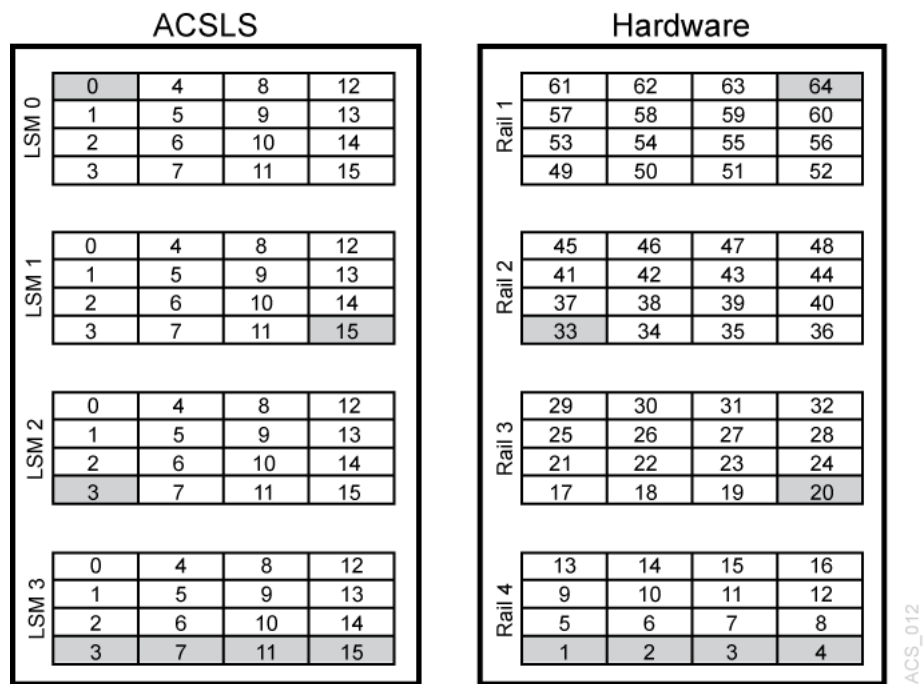
A pop-up window provides more information, such as: VOLID, media and cartridge type (LTO, SDLT tape, and T-Series, data, cleaning, or diagnostic) for cartridges, and shows both the internal and HLI addresses for the location.

## Tape Drive Locations

The tape drives are associated with, and belong to an *LSM*. To mount a cartridge tape in a different LSM, the cartridge must go through an internal pass-thru operation (in this case, *the elevator*) to the drive.

The following figure shows the **internal**—software—mapping (viewed from looking *inside* the library at the tape drives), and the **external**—physical numbering of the drives (looking *outside* at the rear of the Drive and Electronics Module).

**Figure C-2 Software and Physical Drive Numbering**



The highlighted drives show a matching drives. **For example:**

- Internal/software LSM 0 Drive 0 matches with external/physical Drive 64.

- Internal LSM 1 Drive 15 matches with external/physical Drive 33.
- Internal LSM 2 Drive 3 matches with external physical Drive 20.

## Moving Cartridges Before Removing Cells from a Partition

The SL8500 can partition down to the drive and cell array level with enhanced partitioning. For more information, see [Moving Cartridges Before Removing Cells from a Partition](#).

## SL8500 CAPs

Beginning with ACSLS 8.4, two SL8500 CAP types are supported in ACSLS. The legacy Rotational type may have one or two 39-cell CAPs installed on each SL8500 library. The newest CAP type, the Bulk CAP, has eight 36-cell CAPS installed on each library.

## Bulk CAP

Newer SL8500 libraries are designed for faster, more efficient enter and eject operations for data centers where large and frequent vaulting activities are common. There are eight Bulk CAPs on each SL8500 library, with one CAP on each side of each rail. Each CAP contains three removable 12-slot magazines.

ACSLs uses Bulk CAPs to efficiently enter and eject cartridges. Volumes entered from a CAP are moved to a slot on the same side and on the same rail as the CAP. If that side is full, an empty slot on the other side is selected. If the rail is full, an adjacent rail is selected. This strategy minimizes robotics movement and prevents contention between robots. Similarly, if `ejecting.sh` ejects a list of volumes, each volume is ejected to the nearest CAP among the CAPS that you specified. See "[ejecting.sh](#)".

To reserve an open slot in each magazine to serve as a handle during ejects, set the dynamic variable, `BULK_CAP_EJECT_HANDLE`, to **TRUE** using `dv_config`.

```
$ dv_config -p BULK_CAP_EJECT_HANDLE -u
```

When ejecting cartridges to an SL8500 Bulk CAP, leave a slot in each CAP magazine empty so it can be used as a handle. (TRUE/FALSE) [FALSE]: TRUE

Updating configuration file.

When this variable is set to **TRUE**, eleven storage cells in each magazine are used for eject operations. The bottom cell in each magazine on the top three rails remains empty, and the top cell in each magazine on the bottom rail is empty. This enables you to use the empty slot as a handle. This setting does not affect behavior during enter operations.

With eight CAPs per module, there can be eight Bulk CAPs in a ten-string SL8500 configuration. In larger SL8500 complexes, CAP operations could delay mount and dismount operations when multiple enters and ejects are in progress. To mitigate this problem, the dynamic variable, `LIMIT_CAP_CONCURRENT_MOVES`, can limit the number of concurrent enter and eject robotic moves, allowing mounts and dismounts to proceed. To engage this feature, set the dynamic variable, `LIMIT_CAP_CONCURRENT_MOVES`, to **TRUE** using `dv_config`.



```
$ dv_config -p LIMIT_CAP_CONCURRENT_MOVES -u
```

When using large numbers of CAPs for ejects and/or enters in an ACS with multiple libraries, limit the number of concurrent moves to/from CAPs to reserve library resources for mounts and dismounts. (TRUE/FALSE). [FALSE]: TRUE

Updating configuration file

## Upgrading the SL8500 to Handle Bulk CAPs with ACSLS

To update ACSLS when installing Bulk CAPs in one or more SL8500s:

1. Install ACSLS 8.5.  
This can be done far in advance of installing the Bulk CAPs.
2. The Oracle Field Service Engineer (FSE) must load and activate the SL8500 firmware that supports Bulk CAPs onto the affected SL8500s.  
The minimum SL8500 firmware level is 8.50.
3. Before installing the Bulk CAP hardware, use ACSLS `cmd_proc` to vary the libraries offline where Bulk CAPs are installed.
  - If you are installing Bulk CAPs in a stand-alone SL8500 or installing Bulk CAPs in all of the SL8500s in a string, vary the entire ACS (library complex) offline.
  - If you are installing Bulk CAPs in only some of the SL8500s in a library complex, you only need to vary the LSMs involved offline.
4. During this step the FSE installs the Bulk CAP hardware on the affected libraries.
  - a. Before the FSE installs Bulk CAP hardware, you must remove the cartridges in the three cell array columns closest to the service door and save them outside the library. (You will reenter the cartridges after the installation is complete.)  
Reason: To install Bulk CAPs, two columns of system cell arrays must be removed plus the three-pack array. Most of the storage cells in the third column becomes system cells which are no longer accessible to ACSLS.
  - b. The FSE installs the Bulk CAP hardware on the libraries.
5. After all SL8500s have been re-booted and the library hardware audit is complete, use ACSLS `cmd_proc` to vary the SL8500 ACS to diagnostic mode.  
Diagnostic mode prevents ACSLS clients from accessing these libraries while the ACSLS configuration is being updated and the libraries are audited
6. With ACSLS running, use the `config acs acs_id` utility to add the Bulk CAPs to the ACSLS configuration recorded in the database.

### Note:

You can also disable ACSLS, and run `acsss_config`, Option 8, to update the configuration. If you do this, run `query lmu all` from `cmd_proc` and save the output before shutting ACSLS down. Then specify the ACSs to `acsss_config` with the same ACS numbers and with the same port connections. After `acsss_config` is done, enable ACSLS

7. Display and verify the cap status and type from an ACSLS `cmd_proc` with:

```
display cap * -f state mode status size type
```

## Sample output:

```

0 0 0   online   automatic   available   36   SL8500-Bulk
0 0 1   offline  manual     available   36   SL8500-Bulk
0 1 0   online   automatic   available   36   SL8500-Bulk
0 1 1   offline  manual     available   36   SL8500-Bulk
0 2 0   online   automatic   available   36   SL8500-Bulk
0 2 1   offline  manual     available   36   SL8500-Bulk
0 3 0   online   automatic   available   36   SL8500-Bulk
0 3 1   offline  manual     available   36   SL8500-Bulk

```

8. Audit the libraries where Bulk CAPs were installed using `cmd_proc`. You can either
  - Audit the entire ACSLS:
 

```
audit <cap_id> acs <acs_id>
```
  - Audit just the LSMs where Bulk CAPs were installed:
 

```
audit <cap_id> lsm <lsm_id> <lsm_id> <lsm_id> <lsm_id> ...
```
9. Vary the ACS and LSMs online to ACSLS. using the `vary` command. ACSLS clients can now use the Bulk CAPs.
10. Use a CAP to reenter any cartridges that were taken out of the library in Step 4 above.

## Custom SL Console Messages Showing Purpose of Enters and Ejects

The SL Console can display custom operator messages on the CAP Status screen that show the purpose for enters to and ejects from Bulk CAPs. These messages can also report the partition into which cartridges are being entered or the partition from which cartridges are ejected.

These operator messages are optional. They do not affect the underlying enter and eject processing, and they are only supported for SL8500 Bulk CAPs.

To use custom operator messages:

1. Define the message to be displayed for an `opmsg` number using the SL Console. Select the following options.

```

Tools
  Configuration
    CAP Usage Message

```

Define a message number, from 4 through 99, and an associated message. If possible, limit the message to 20 characters, so it fits in the available space.

2. Enter the optional operator message number on manual enters or ejects:
 

```
enter <cap_id> [opmsg <opmsg_nbr>]
```

```
eject <cap_id> [opmsg <opmsg_nbr>] vol_id | vol_range ...
```

Custom `opmsg` numbers cannot be specified for ejects from ACSAPI clients, the ACSLS GUI, or `lib_cmd eject`. In these cases the default messages are displayed.

The `opmsg` message displays on the System Details, CAP Status page after a CAP is unlocked for cartridges to be entered, to be inserted, or cartridges being ejected to be removed.

Example: To specify custom operator panel message number 55 when entering cartridges through Bulk CAP 1,2,1:

```
enter 1,2,1 opmsg 55
```

## Rotational CAP

The SL8500 Rotational CAP spans three rails (2,3,4) corresponding to LSMs 1, 2, and 3. The base configuration includes one CAP per SL8500 module, and a second CAP that can be installed as an option.

Each Rotational CAP has three magazines with 13 cells each. The magazines are situated each on a different rail, accessible only to the handbots on that rail. During an enter, ACSLS attempts to move cartridges from each magazine to the adjacent LSM (rail). Only if the adjacent rail is full will the volume be moved to a different rail. Similarly, volumes on a given rail are ejected to the adjacent magazine on that rail.

Since the top rail (LSM 0) does not have an adjacent CAP magazine, an elevator move automatically comes into play with volumes ejected from the top rail. On enters, the top rail is not populated until a lower rail is filled to capacity. Volumes mounted to drives on the top rail eventually migrate to the top LSM when they are dismounted. Otherwise, an explicit move operation is needed to situate volumes in the LSM on the top rail. Such an extra move can be handled automatically after an enter with the `watch_vols` utility. See [watch\\_vols](#).

Because the single Rotational CAP serves multiple LSMs, the Rotational CAP state is not tied to the online or offline state of any LSM. The CAP can remain online whether any one or all of the adjacent LSMs are offline. Conversely, if the CAP is offline, it is not brought online automatically when any LSM is varied online.

Even though multiple LSMs access the CAP, the SL8500 Rotational CAP is addressed as though it was in LSM 1 (such as 1,5,9,13). In a partitioned library where each partition is assigned to a different host, users must be aware that the CAP is a shared resource. A Rotational CAP becomes reserved immediately upon an enter or eject operation. Operators in shared environments should promptly fill or empty the CAP and close the door upon completion of the CAP operation. See [Library Partitioning](#).

Older releases of the SL8500 had reported the optional CAP as present but not operational if the second CAP was not actually installed. As a work-around, ACSLS users were instructed to keep the desired state of the non-existent CAP offline. This is no longer an issue beginning with library firmware level 6.07 and above.

## Enter or Eject Operations

During an `enter`, ACSLS will always try to move the cartridge to an LSM (rail) adjacent to the CAP magazine. For ejects, ACSLS will always try to eject cartridges to a CAP cell adjacent to the LSM containing the cartridge.

If these two operations are not possible, the library controller takes care of moving the cartridge through the elevator to another LSM. This requires movements between two handbots and the elevator.

## Enter, Eject, and Audit Operations for Some ACSLS Clients

Unlike other libraries, the SL8500 does not have CAPs defined for each LSM ID in an SL8500 library. The CAPs on an SL8500 contain LSM 1 in their CAP IDs. There are no CAPs in an SL8500 with LSM IDs 0, 2, or 3. Partitioning complicates this problem because LSM 1

(the LSM ID in the SL8500 CAP ID) may not be assigned to your partition. (Remember, the CAPs are still available to all partitions as a shared resource).

Some ACSLS clients do not *query* ACSLS to identify which CAPs exist and are available before selecting a CAP for an *enter*, *eject*, or *audit*. They may specify *cap\_ids* that do not exist, or CAPs that are not online. For example, some ACSAPI clients assume CAPs exist for all LSM IDs. They may automatically specify a CAP with the same LSM ID as the location of cartridges or drives that they manage. Enters, ejects, or audits that specify non-existent CAP IDs will fail.

You must use the ACSLS `cmd_proc` to:

- *enter* and *eject* cartridges for clients that specify non-existent CAP IDs.
- run audits for any ACSs and partitions used by these clients.

After entering, ejecting, or auditing, you must re-synchronize the client application's database with the ACSLS database.

## Minimizing Elevator and PTP Activity

There are several things you can do to minimize elevator and PTP activity.

**Table C-2 Recommendations to Minimize Elevator and PTP Activity**

| Action              | Recommendation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Mounting cartridges | Whenever possible, when mounting a tape, use cartridges and tape drives that are in the same LSM. LSM refers to a <b>single rail</b> within the SL8500 library. Each SL8500 contains 4 LSMs.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Using float         | Take advantage of the ACSLS "float" option (enabled by default by ACSLS) by maintaining some free cells within each LSM. Cartridge float is a feature that allows ACSLS to place a dismounted tape cartridge in an empty slot in the same LSM or a closer LSM to the tape drive if the tape originally came from a different LSM using a pass-thru operation. When cartridges are dismounted, ACSLS tries to avoid elevator (pass-thru) activity among LSMs by assigning a new home cell whenever the cartridges' old home cell is in a different LSM. ACSLS attempts to put the cartridge away: <ul style="list-style-type: none"> <li>• in the same LSM as the tape drive from which it was dismounted</li> <li>• or to the closest LSM (with free storage cells) to the drive</li> </ul> |
| Entering cartridges | <i>Enter</i> cartridges into an LSM that has compatible tape drives for the media being entered.<br>Example: You have only LTO drives on LSMs 2 and 3, and you want the LTO cartridges to be located in these LSMs. When entering these cartridges, you should place them in the CAP magazines adjacent to LSMs 2 and 3. ACSLS then makes every effort to put the cartridges in the LSM that is adjacent to that CAP magazine.                                                                                                                                                                                                                                                                                                                                                              |
| Scratch cartridges  | Make sure that scratch cartridges are available in sufficient quantity in each LSM where they will be used. For an SL8500, this means having scratch cartridges available on each rail (LSM) of the library.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Free cells          | Make sure that there are adequate free cells in each LSM.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

## Configuring Tape Drives to Support Your Workloads

How tape drives are configured in the SL8500 can minimize both elevator and PTP activity, while supporting your tape workloads. Strategies to use in determining where tape drives are located in the SL8500 include:

- Cluster cartridges by workload, with enough drives to support the maximum drives needed for the workload. Separate the cartridges used by each workload on separate rails, and ensure the rail(s) dedicated to a workload has enough drives to meet the maximum concurrent mounts for the peak usage of the workload. Ensure that the rail has not only the tape cartridges for the workload, but also the scratch cartridges that will be needed.
- Allocate separate rail(s) to each major application workload. This is because some applications, such as Symantic NetBackup and Tivoli, can use their own media and drives.
- Clustering drives and media on a single rail works until the mounts per hour threshold is reached, all drives are in use, or there are too many active cartridges to fit on a rail. When the resources needed for a workload exceeds the capacity of a rail, spread the cartridges and drives over two or more rails.
- Cluster drives by type, placing drives that use different media types on separate rails (LSMs). For example, place T9840 drives on one rail and T10000 drives on a different rail.
- Configure your heavy tape applications so they will not exceed the performance limits of your library configuration.
- Configure the SL8500 with 8 HandBots (two HandBots per rail) to provide redundancy. This allows you to always access the cartridges and drives that support a workload.

## Managing Cartridge Locations

How cartridges are originally entered in the library, or their status in the library, can have an affect on ACSLS performance.

**Table C-3 Recommendations for Managing Cartridges**

| Action              | Recommendation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|---------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Entering cartridges | <p>Enter cartridges through the CAP.</p> <p>When manually placing cartridges in the library with the front access door open, library operations cease and ACSLS must perform a full audit to update the library database to match the actual contents of the library.</p> <p>To maximize performance, enter cartridges through the cartridge access port (CAP). During an enter, the library stays online, mounts can continue, and the library management software always tries to move the cartridge to an LSM adjacent to the CAP magazine—minimizing pass-thru activity.</p> <p>If this is not possible, the library controller moves the cartridge through the elevator to another LSM—which requires additional movement between two HandBots and the elevator.</p> |

**Table C-3 (Cont.) Recommendations for Managing Cartridges**

| Action                       | Recommendation                                                                                                                                                                                                                |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Clustering cartridges        | Cluster cartridges by workload on separate rails with enough tape drives to support the maximum activity—peak usage—for that workload.                                                                                        |
| Using float                  | When <code>float</code> is on, ACSLS selects a new home cell for a cartridge that is in an LSM as close to the drive as possible on a dismount. This option automatically clusters cartridges by the drives for the workload. |
| Supplying scratch cartridges | Make sure each rail has the correct amount and type of data cartridges plus enough scratch cartridges to support the workload.                                                                                                |

## Finding Missing Cartridges

If a cartridge is out of place or unaccounted for by ACSLS:

1. Perform a physical audit of the SL8500 using the SL Console.

The physical audit of the SL8500 is performed as a background task in between handling mount and other library operation requests.

### ▲ Caution:

If the SL8500 contents are out of sync with ACSLS due to manual operations such as loading cartridges directly, it is not advisable to attempt continued operations.

If you want to manually add tapes, adding them to a particular LSM within the SL8500 is a better approach. Adding tapes to a particular LSM and auditing only the affected LSM is a quicker and more reliable solution.

You must `vary` the affected LSM to a diagnostic state to ACSLS while the audit is in process. After the SL8500 library audit is performed, `vary` the LSM online to ACSLS.

2. Run an ACSLS `audit` to update the ACSLS database to match the actual inventory of library cartridges.

## Varying the SL8500 Offline

`Vary` SL8500 components offline to ACSLS before they are powered off, if they are inoperative, and before you open an SL8500 access door. This notifies ACSLS that they are unavailable. Once they are available, `vary` them back online.

## Use ACSLS to vary SL8500 components offline, not SL Console

`Vary` SL8500 components (ACSSs, LSMs, and CAPs) offline to ACSLS, not the SL Console.

ACSLs allows outstanding requests to complete before taking components offline, unless it is a vary offline force. The SL Console has no knowledge of outstanding requests to ACSLS.

Varying components offline using SL Console may cause requests in progress to fail.

## When to vary SL8500 components offline to ACSLS

This section discusses when to vary SL8500 components offline to ACSLS.

### Before Opening the Access Door

Before opening the SL8500 access door, vary the ACS or all four LSMs offline.

- For a standalone SL8500, vary the ACS offline using the following command:

```
vary acs acs_id offline
```

- For a SL8500 connected through PTPs, vary all four LSMs (in the SL8500 whose access door will be opened) offline using the following command four times (once for each of the four LSMs):

```
vary lsm lsm_id offline
```

#### Note:

If any CAPs in the SL8500 are in automatic mode, you must:

1. Set them to manual mode before opening the access door.
2. Set them back to automatic mode after you close the access door and the SL8500 comes back online.

### If a CAP is inoperative

If the CAP is inoperative, vary it offline using the following command:

```
vary cap cap_id offline
```

### When closing the Service Safety Door

Whenever replacing hardware requires using the Service Safety Door, it is advisable to keep that Service Safety Door closed for the minimum amount of time possible. The Service Safety Door blocks other hardware components (elevators, CAPs, and cells) to which access may be required for completing specific requests.

- Before closing the Service Safety Door on either the left or right-side of the SL8500, vary the elevator on that side offline through the SL Console.

After the Service Safety Door is opened, vary the elevator on that side back online through the SL Console.

- When the Service Safety Door is closed on the right-side, it will block access to the CAP.
- Before closing the Service Safety Door on the right-side of the SL8500, vary the CAP offline through ACSLS.
- After the Service Safety Door is opened, vary the CAP online through ACSLS.

 **Note:**

When the SL8500 Service Safety Door is closed to separate a service bay from the rest of the library, the CSE can open the access door on that side without taking the LSM or ACS offline.

## When using the Service Safety Door do not use these ACSLS commands and utilities

There are some ACSLS commands and utilities that should not be in progress or initiated when the Service Safety Door is being used. These commands, are:

When the Service Safety Door is closed on either side, do not use the following utilities:

- `acsss_config`
- `config` (`config drives is OK`)

When the Service Safety Door is closed on the right (CAP) side, do not use the following commands:

- `enter`
- `eject`
- `set cap mode auto <cap_id>`

When the Service Safety Door is closed on the right (CAP) side, the following commands can be used, but special considerations apply:

- `audit`  
The audit command can be used. However, if there is a need to `eject` cartridges as a result of the audit (because the audit encounters duplicates or unreadable labels), the audit will complete and update the ACSLS database, but the cartridges will **not** be ejected.
- `vary acs` and `vary lsm`

These commands will succeed, but messages will be displayed on `cmd_proc` and the Event Log reporting CAP failures and inoperative CAPs.

## Using the Dynamic Configuration (config) utility

The dynamic configuration (`config`) utility allows you to implement configuration changes to ACSLS libraries (and components) while ACSLS remains online and running. These configuration changes are recorded in the `acsss_config.log` file.

The following dynamic configuration utilities are supported:

- `config acs`
- `config drives`
- `config lsm`
- `config ports`



Using the `config` utility provides the following benefits:

- ACSLS can continue running, allowing you to perform `mount` requests to unaffected library components.
- Allows you to reconfigure specified library components while all other configuration information remains unchanged. For example, when specifying:
  - An ACS, the configurations of other ACSs are not affected.
  - An LSM, the configurations of other LSMs are not affected.
  - A drive panel, the drives on a panel, mounts and dismounts to all existing drives are not affected.

## Expanding the SL8500

Storage Expansion Modules (SEMs) are added to the SL8500 to increase its capacity. The SEMs are inserted between the Customer Interface Module (CIM), which includes the CAP, and the SEM or Robotics Interface Module (RIM) that is currently attached to the CIM.

When an SL8500 is expanded:

- The library configuration changes, and you must reconfigure ACSLS using either the ACSLS Dynamic Configuration utility while ACSLS is up, or run `acsss_config` while ACSLS is down.
- Cartridges must be removed to expand the SL8500. **When placing these cartridges back into the library, leave the cell arrays that they were formerly in, empty.**
- After the expansion is complete, the SL8500 must be rebooted twice. First to discover the new configuration, and second, to restart all library components with the updated configuration.
- The library must update the locations of all cartridges with a physical audit.

The library begins a physical audit automatically when the access door is closed. The library's physical audit completes when all robot movement stops for at least a minute.

- After the library's physical audit is complete, use the procedure [Auditing an Expanded SL8500](#) to update the ACSLS database.

### Note:

Do not start automated library operations until the ACSLS audits are complete, and the ACSLS database has been updated with the new cartridge locations.

## What is Involved

Physically expanding an SL8500 involves the following:

- A new SEM(s) is inserted between the existing SEMs or RIM and the CIM.  
The panel numbers on the new SEM(s) and the CIM are now higher than the panel numbers on the existing SEMs and RIM.
- Because the CIM must move out, new and higher panel numbers are assigned to the three cell panels (columns) on the CIM. When the cell panels on the CIM are assigned higher panel numbers, the addresses of all cartridges on the CIM change.

- Many cartridges must be removed to expand the SL8500. Cell arrays must be removed to both unbolt existing rails, and bolt-in new ones, to jack-up the CIM.
- After the new rails are installed and the SL8500 is expanded, the cartridges that were removed may be loaded back into the library. **Leave the cell arrays that were temporarily removed to expand the library, empty.**

After expansion, `audit` the library so ACSLS can update its database with the new addresses of these cartridges. When cartridges are placed in new locations, ACSLS' attempts to find the cartridges that were previously in the cells can seriously degrade library performance during an audit. To optimize both ACSLS and library performance, complete the following procedure:

1. Let the library's physical audit complete before starting an ACSLS audit. The combined audits will finish faster if the library is able to report cartridge locations to ACSLS from its database. Otherwise, the library must re-verify cartridge locations before responding to ACSLS' `audit` requests.
2. For the cartridges that were removed to expand the library, use one of these strategies to re-insert them back into the library:
  - a. Remove the cartridges from the library, `audit` the library from ACSLS, then `enter` the cartridges that were removed back into the library.
  - b. Insert the cartridges only into the panel numbers that were added to the library, then `audit` these panels, first.

If you are not using ACSLS scratch pools to track scratch cartridges, no special procedure is required, if you have absent volume retention enabled (the default). When you `audit` the library, the cartridges that were moved are often marked absent, but `audit` finds them in their new locations and reactivates them. The location of the cartridges is updated, and no important information is lost.

If you are managing scratch cartridges by assigning cartridges to ACSLS scratch pools, and you do not want to have the cartridges' scratch status cleared, place the cartridges in the newly added panels, and `audit` these panels, first.

- Insert the cartridges that were removed into the panels that were added to the library. This means, only insert them into panels (columns) beyond the three panels that are closest to the drives in the first new SEM. (The first three panels in the first SEM have the panel numbers that were formerly assigned to the three panels in the CIM.
- Then, `audit` the panels in the ACS or LSMs from the highest panel number down, through all of the new panel numbers that were added:  
Use `audit cap_id panel panel_id` commands.
- Finally, `audit` the rest of the ACS or LSMs:  
Use the `audit cap_id acs acs_id` or `audit cap_id lsm lsm_id` commands.

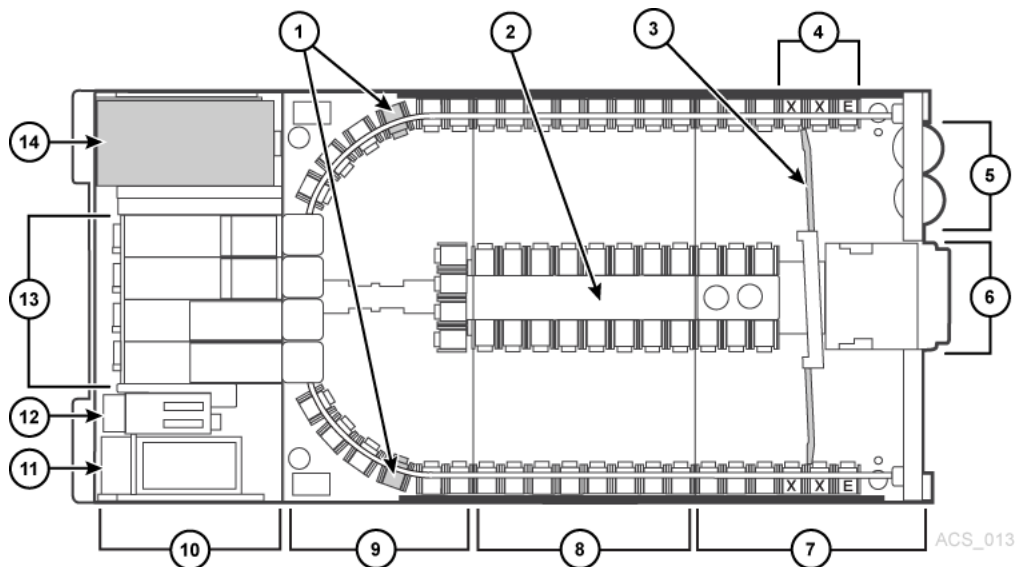
This procedure is described in full detail in [Auditing an Expanded SL8500](#).

## Diagram of SL8500 Modules:

In the following diagram, the three cell panels in the Customer Expansion Module (#5) are the three columns of cartridges on both the outside and inside walls on the drive

side of the CEM. These are the only cell addresses in the CIM that HandBots can access when the service safety door is activated.

**Figure C-3 SL8500 Customer Expansion Module**



Legend:

1. Cartridge Access Ports (2) Caps
2. Facade Operator Panel (Optional) Icey Pad
3. Customer Interface Module
4. Storage Expansion Module
5. Robotics Interface Module
6. PTP (Pass Thru Port)
7. Drive Electronics Module
8. AC Power Supplies Electronic Control Module
9. DC Power Supplies
10. Tape Drives
11. Accessory Racks
12. Inner wall
13. Service Door
14. Reserved Columns
  - E = End Stop
  - S = System Cell

## Auditing an Expanded SL8500

Before expanding the SL8500:

1. If you want to preserve the cartridges' scratch status:
  - a. Determine the highest panel number for the LSMs in the library by running the following command:

```
display panel acs,lsm,*
```

Where:

    - `acs` is the ACS
    - `lsm` is one of the LSMs. All rails (LSMs) on the SL8500 have the same number of panels.
    - `*` displays all panels in the LSM.
  - b. Record the highest panel number.
2. Vary the LSMs to a diagnostic state until the configuration is updated and cartridge addresses have been updated with `audit`.
3. Backup the ACSLS database.
4. Expand the SL8500:

During the expansion, cartridges must be removed to install new or expanded rails, and to add the Storage Expansion Modules.

 **Note:**

Do not insert these cartridges back into the cell arrays that were temporarily removed to expand the library. This avoids a severe degradation in library performance when these locations are first audited.

5. Insert the cartridges that were removed into the panels that were added to the library.

This means, you should only insert these cartridges into panels (columns) beyond the three panels that are closest to the drives in the first new Storage Expansion Module. (The first three panels in the first Storage Expansion Module retain the numbers formerly assigned to the panels in the CIM.)

After physical expansion of the SL8500 is complete:
6. Reboot the library twice.
7. Leave the LSMs in a diagnostic state.
8. Update the configuration in the ACSLS database using one of the following methods:
  - ACSLS is running:

```
config acs acs_id
```

or, for each LSM (rail) in the SL8500:

```
config lsm lsm_id
```
  - Shutting down ACSLS:
    - Bring down ACSLS:

```
acsss disable
```
    - Update the ACSLS configuration:

```
acsss_config
```

– Bring up ACSLS:

```
acsss enable
```

If you want to preserve the cartridges' scratch status, `audit` the library using steps 9 through 11. If you are not tracking the cartridge scratch status using ACSLS scratch pools, skip to step 12.

- Determine the new highest panel number for the LSMs in the library, and record this number using the following command:

```
display panel acs,lsm,*
```

Where:

- `acs` is the ACS.
  - `lsm` is one of the LSMs. All rails (LSMs) on the SL8500 have the same number of panels
  - `*` displays all panels in the LSM.
- Audit the panels that were added during the expansion, from the new highest panel number to the old highest panel number+1. Audit these panels in descending order using the following command:

```
audit cap_id panel panel_id
```

Where:

- `cap_id` is the CAP where cartridges with duplicate `vol_ids` or unreadable labels should be ejected.
  - `panel_id` is the panel being audited (`acs,lsm,panel`)
- Perform a separate audit for each added panel in each LSM.

 **Note:**

You can only run as many concurrent audits as you have CAPs in your library complex. Each concurrent audit needs a separate CAP. If the audit ejected any cartridges to the CAP, they must be removed before the audit finishes.

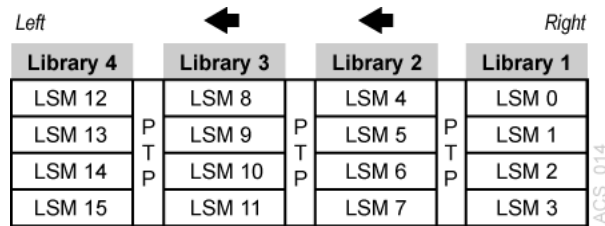
- Audit each entire LSM in the expanded SL8500 (or audit the entire ACS).
- Backup the ACSLS database.
- Vary the LSMs online.
- Resume automated library operations.

## Connecting SL8500s with Pass-Thru-Ports

You may need to `audit` SL8500(s) to update cartridge addresses when adding SL8500 libraries to an existing SL8500 library complex.

LSMs in the SL8500 complex are numbered from top-down and right-to-left when viewed from the CAP end, as shown in the following figure.

**Figure C-4 Four connected SL8500 Libraries**



Depending on how your site is set up, you may be adding a new SL8500 to the right or left of an existing SL8500. As a result:

- When a new SL8500 is added on the left, the new LSMs have higher numbers, and existing LSMs are not renumbered. The disruption is minimal.
- When a new SL8500 is added on the right, all existing LSMs are renumbered. This causes the home cell addresses of all existing cartridges to change.

## Before You Install SL8500 PTP Connections

1. Apply all relevant maintenance to ACSLS before installing SL8500 connections.
2. Enable absent cartridge support.  
When absent cartridge support is enabled, ACSLS will not delete cartridges in an LSM that is deleted or when `audit` does not find them in their old address. These cartridges are marked absent, preserving all information about them, except for the scratch status. When these cartridges are later found by `audit`, they are reactivated.
  - Absent cartridge support is enabled when the `ABSENT_VOLUME_RETENTION_PERIOD` variable is **not** zero. The default value is 5 days.
  - Use `acsss_config` (option 3). For procedures, refer to [Setting CSI Tuning Variables](#) and the variable `ABSENT_VOLUME_RETENTION_PERIOD`.
3. Update your configuration by one of the following methods:
  - With ACSLS up, use the following command:

```
config acs acs_id
```
  - Shutting ACSLS down, use the following command:
    - Bring down ACSLS: `acsss disable`
    - Update the ACSLS configuration: `acsss_config`
    - Bring up ACSLS: `acsss enable`

ACSLs cannot be running when you run `acsss_config`.

For more information, refer to [Setting CSI Tuning Variables](#) or under the [The acsss Macro](#).

## Adding New SL8500s

When additional SL8500s are added to an existing SL8500 library complex, the new ACSLS configuration must be updated. If the addition of new SL8500s causes the LSMs in the existing SL8500s to be renumbered, the cartridge addresses in those LSMs must be updated.

The cartridge addresses should be updated without losing the other information associated with the cartridges. This includes cartridge information, such as scratch status, pool, locks, and ownership and cleaning cartridge usage counts, and the date and time that cartridges were entered.

LSMs in connected SL8500s are numbered from top-to-bottom and right-to-left (as viewed from the CAP end). See [Figure C-4](#).

## Adding a New SL8500 to the Left

In the case of the non-disruptive addition of a new SL8500 on the left (from the CAP end) of the original SL8500(s), use the host software (ACSLs) to dynamically configure the additional drives and libraries. You do not need to recycle ACSLS if Dynamic Configuration is installed. `Mount` requests continue as normal to the existing SL8500s during this phase.

If cartridges were placed in cells in the new SL8500, an ACSLS `audit` must be run on the LSMs of the new SL8500 to add these cartridges to the ACSLS database.

The LSMs in the pre-existing SL8500s can be online during the audit.

## Dynamically Configuring the New ACSLS Configuration

1. Add new SL8500(s) to the library complex.
2. Backup ACSLS before you make the configuration change.
3. Update the ACSLS configuration dynamically, use the following command:

```
config acs acs_id
```

You can also update the ACSLS configuration when ACSLS is down, use the following command:

```
acsss_config
```

4. Backup ACSLS after you make the configuration change.

### Note:

If there are cartridges in the new SL8500, `audit` the LSMs (rails) in the new SL8500 to add these cartridges to the ACSLS database.

## Adding a New SL8500 to the Right

If you want to add a new SL8500 to the right, the existing LSMs will be renumbered, as shown in the following figures.

## Considerations When Adding an SL8500 to the Right

When new SL8500s are added to the right, all existing LSMs will be renumbered and cartridge addresses will change. The change to LSM numbers causes all existing cartridge addresses to change. If ACSLS attempts to `mount` a cartridge whose address changed, the `mount` will fail because ACSLS cannot find the cartridge.

- Quiesce mount activity until all cartridge addresses have been updated.
- `Vary` all LSMs in the ACS to diagnostic state to prevent mounts.
- Update the cartridge addresses by auditing the LSMs in the existing and new SL8500s in a specific sequence.

## Dynamically Configuring the New ACSLS Configuration

To a update the ACSLS Configuration Dynamically after adding a new SL8500(s):

1. `Vary` existing LSMs in diagnostic state (`vary lsm_id diag`).

### **Caution:**

These LSMs must remain in diagnostic state until they have been audited. Otherwise, the following problems will occur:

- Mounts will fail because cartridges cannot be found in their last-known addresses.
  - Until the audits update the ACSLS map of the empty cells in the renumbered LSMs:
    - Enters of new cartridges will collide with existing cartridges.
    - Movements of cartridges to existing (renumbered) LSMs will collide with cartridges already in the cells.
2. Add the new SL8500(s) to the library complex.
  3. Backup ACSLS (before making any configuration changes).
  4. Update the ACSLS configuration dynamically using the following command:

```
config acs acs_id
```

### **Note:**

As an alternative, update the ACSLS configurations while ACSLS is down. First, bring ACSLS down using the `acsss disable` command. Change the configuration with `acsss_config`, and finally, bring ACSLS back up with `acsss enable`.



 **Note:**

New LSMs are added when in an online state. Vary these LSMs to a diagnostic state until they have been audited.

5. Backup ACSLS (after the configuration change).
6. Audit the library to update cartridge addresses (since they were renumbered).

Use the following sequence:

- a. Audit each LSM in the pre-existing SL8500s that were renumbered.
  - Audit the pre-existing SL8500s in a careful sequence to avoid losing cartridges.
 

Audit each LSM (rail) in sequence from the highest numbered LSM to the lowest numbered LSM ID. The audit finds all of the cartridges in their (renumbered) LSM addresses.
  - For ACSLS, you must audit each LSM separately, and allow the audit of one LSM to finish before auditing the next LSM.

Audit checks the cartridges' former home cell addresses (in the LSM to the right), and then updates the cartridges' addresses. The audits for the LSMs in the left most SL8500 will take a long time. The audits in all of the other SL8500s will be faster.

The reason that the audits of the left of most LSMs take longer is that they will trigger a cascade of Cartridge Recovery requests. When audit finds a cartridge that has a different home cell address, audit checks the cell address recorded in the database. If that cell has a different cartridge, Cartridge Recovery investigates that cartridge, and so on. Because all of the LSM addresses have changed, these recoveries will cascade across the SL8500 library complex.

- Keep the LSMs in diagnostic state until they have been audited. After they have been audited, you can vary them online. Audit has updated the addresses of cartridges in this LSM, so you can resume automated mount activity with the audited LSMs' cartridges.
- b. Last, audit the LSMs in the newly added SL8500 (that have the lowest LSM IDs).
 

After these LSMs have been audited, they can be varied online, and cartridges in them can be mounted on tape drives, as shown in the following figure:

**Figure C-5 Auditing Order for Renumbered LSMs**

| First four audits        |             | Second four audits       |             | Third four audits        |             | Last four audits         |
|--------------------------|-------------|--------------------------|-------------|--------------------------|-------------|--------------------------|
| Now LSM 12<br>(Audit #4) | P<br>T<br>P | Now LSM 8<br>(Audit #8)  | P<br>T<br>P | Now LSM 4<br>(Audit #12) | P<br>T<br>P | New LSM 1<br>(Audit #16) |
| Now LSM 13<br>(Audit #3) |             | Now LSM 9<br>(Audit #7)  |             | Now LSM 5<br>(Audit #11) |             | New LSM 2<br>(Audit #15) |
| Now LSM 14<br>(Audit #2) |             | Now LSM 10<br>(Audit #6) |             | Now LSM 6<br>(Audit #10) |             | New LSM 3<br>(Audit #14) |
| Now LSM 15<br>(Audit #1) |             | Now LSM 11<br>(Audit #5) |             | Now LSM 7<br>(Audit #9)  |             | New LSM 4<br>(Audit #13) |

7. Backup ACSLS after the audits have completed.

## Procedures for Merging ACSs

SL8500 PTP supports the merging of multiple, separate SL8500s into a single ACS. To minimize the loss of information about cartridges and operational downtime, follow these recommended procedures.



### Note:

When the ACSLS configuration is updated, global cartridge address changes **are not** made

For ACSLS, two scenarios are described below. In the first, the ACSs to be merged are numbered from right-to-left (as you face them from the CAP-side). In the second, they are numbered from left-to-right. It is assumed that the resulting ACS will use the lower/lowest ACS ID.

### Merging ACSs Numbered from Right to Left

In the following scenario, the ACSs to be merged are numbered from right-to-left (as you face them from the CAP side).

**Figure C-6 ACSs to be Merged**

| ACS 2   | ACS 1   | ACS 0   |
|---------|---------|---------|
| LSM 2,0 | LSM 1,0 | LSM 0,0 |
| LSM 2,1 | LSM 1,1 | LSM 0,1 |
| LSM 2,1 | LSM 1,2 | LSM 0,2 |
| LSM 2,3 | LSM 1,3 | LSM 0,3 |

ACS\_017

**Figure C-7 Desired Configuration: Single ACS**

| ACS 0    |   |         |
|----------|---|---------|
| LSM 0,8  |   | LSM 0,0 |
| LSM 0,9  | P | LSM 0,1 |
| LSM 0,10 | T | LSM 0,2 |
| LSM 0,11 | P | LSM 0,3 |
|          |   | LSM 0,4 |
|          |   | LSM 0,5 |
|          |   | LSM 0,6 |
|          |   | LSM 0,7 |

ACS\_018

### Procedure for Merging ACSs Numbered from Right-to-Left

1. Vary all ACSs, except the right most ACS being merged, offline.  
This prevents mounts and dismounts while cartridge addresses are being updated.
2. Stop ACSLS: `acsss disable`
3. Backup ACSLS before making any configuration changes.
4. Update the ACSLS configuration (while ACSLS is down) using `acsss_config`

5. Bring up ACSLS: `acsss enable`
6. Vary the new LSMs (that were added to the ACS) to a diagnostic state.  
These LSMs were added while in online state.
7. Backup ACSLS after the configuration change.
8. Audit the LSMs that were added to the ACS.  
The order in which they are audited does not matter. The entire ACS, or all LSMs can be audited at once.  
The cartridges will be re-activated.
9. Backup ACSLS after the audits are complete.
10. Vary the new LSMs online, and resume normal automated processing.

## Merging ACSs Numbered from Left-to-Right

In the following scenario, the ACSs to be merged are numbered from left-to-right.

**Figure C-8 Existing Configuration: Three Separate ACSs**

| ACS 0   | ACS 1   | ACS 2   |
|---------|---------|---------|
| LSM 0,0 | LSM 1,0 | LSM 2,0 |
| LSM 0,1 | LSM 1,1 | LSM 2,1 |
| LSM 0,1 | LSM 1,2 | LSM 2,2 |
| LSM 0,3 | LSM 1,3 | LSM 2,3 |

ACS\_019

**Figure C-9 Desired Configuration: Single ACS**

| ACS 0    |                  |         |                  |         |
|----------|------------------|---------|------------------|---------|
| LSM 0,8  | P<br>T<br>P<br>P | LSM 0,4 | P<br>T<br>P<br>P | LSM 0,0 |
| LSM 0,9  |                  | LSM 0,5 |                  | LSM 0,1 |
| LSM 0,10 |                  | LSM 0,6 |                  | LSM 0,2 |
| LSM 0,11 |                  | LSM 0,7 |                  | LSM 0,3 |

ACS\_018

## Procedure for Merging ACSs Numbered from Left-to-Right

1. Vary all ACSs offline.  
This prevents mounts and dismounts while cartridge addresses are being updated.
2. Stop ACSLS: `acsss disable`.
3. Backup ACSLS before making any configuration changes.
4. Update the ACSLS configuration (while ACSLS is down) using `acsss_config`.
5. Bring up ACSLS: `acsss enable`.
6. Vary the new LSMs (that were added to the ACS) to a diagnostic state.  
(These LSMs were added while in online state).
7. Backup ACSLS after the configuration change.

8. Vary the new LSMs that were added to the left, to a diagnostic state. You do not want to perform mounts and dismounts until you have audited these LSMs.
9. Audit the newly configured LSMs. The order in which they are audited does matter. Audit in this order:
  - a. First, audit the LSMs in the SL8500(s) that retains the same ACS number first.
    - You want `audit` to find these cartridges in their new LSM addresses before you `audit` the LSMs that now are assigned to these cartridges' old addresses.
    - The LSM addresses will be updated for all cartridges in these SL8500(s).
    - For ACSLS, `audit` each LSM separately, and allow the audit of one LSM to finish before auditing the next LSM.
    - Keep the LSMs in diagnostic state until they have been audited.

After they have been audited, you can `vary` them online. `Audit` has updated the cartridges' addresses, so you can resume automated `mount` activity with the audited LSMs' cartridges.
  - b. Last, `audit` the SL8500s that were merged into the lowest ACS. The cartridges in these SL8500s will be re-activated.
    - All of these LSMs can be audited at the same time.
    - The order in which these LSMs are audited does not matter.
10. Backup ACSLS (after the audits are complete).
11. Vary all LSMs online and resume normal automated processing.

## Removing PTPs and Splitting ACSs

You may decide to remove the PTP mechanisms that connect two SL8500s, and split a single ACS into two separate ACSs. This configuration change is the reverse of merging two ACSs into a single ACS.

It is much easier to add the new ACS with the SL8500(s) on the left-hand side of the split. This assigns the higher numbered LSMs to the new ACS, and avoids renumbering the LSMs that remain in the existing ACS.

## Splitting ACSs, with the new ACS Created from SL8500(s) on the Left - Possible Scenarios

**Figure C-10 Existing Configuration: One ACS**

| ACS 0    |     |          |     |         |     |         |
|----------|-----|----------|-----|---------|-----|---------|
| LSM 0,12 | PTP | LSM 0,8  | PTP | LSM 0,4 | PTP | LSM 0,0 |
| LSM 0,13 |     | LSM 0,9  |     | LSM 0,5 |     | LSM 0,1 |
| LSM 0,14 |     | LSM 0,10 |     | LSM 0,6 |     | LSM 0,2 |
| LSM 0,15 |     | LSM 0,11 |     | LSM 0,7 |     | LSM 0,3 |

ACS\_020

**Figure C-11 Desired Configuration: Two ACSs**

| ACS 1   |             |         | ACS 0   |             |         |
|---------|-------------|---------|---------|-------------|---------|
| LSM 1,4 | P<br>T<br>P | LSM 1,0 | LSM 0,4 | P<br>T<br>P | LSM 0,0 |
| LSM 1,5 |             | LSM 1,1 | LSM 0,5 |             | LSM 0,1 |
| LSM 1,6 |             | LSM 1,2 | LSM 0,6 |             | LSM 0,2 |
| LSM 1,7 |             | LSM 1,3 | LSM 0,7 |             | LSM 0,3 |

## ACSLs Procedure for Splitting an ACS

With Dynamic Configuration, ACSLS can add ACSs while it is running. However, Dynamic Configuration does not make global cartridge address changes.

1. Backup ACSLS before the configuration change.
2. Vary the LSMs that will be moved to the new ACS offline.  
This prevents mounts and dismounts while cartridge addresses are being updated.  
The LSMs that are remaining in the existing ACS can remain online.  
Mounts and dismounts in these LSMs can continue.
3. Remove the four PTP mechanisms that connect the SL8500s that you want to separate.
4. Reconfigure the existing ACS once the pass-thru ports have been removed.  
This will remove the LSMs that are moving to the new ACS. Use the following command:

```
config acs acs_id
```

5. Add the new ACS by using the following command:

```
config acs new
```

Alternatively, you can update the ACSLS configurations when ACSLS is down:

```
acsss_config
```

New LSMs are added in an online state.

6. Vary these LSMs in a diagnostic state until they have been audited.
7. Vary the new ACS to diagnostic state.
8. Backup ACSLS after the configuration change.
9. Audit the LSMs in the new ACS.  
You do not want to perform mounts and dismounts until you have audited these LSMs.  
The order in which they are audited does not matter. They can all be audited at once.  
The cartridges will be re-activated.
10. Backup ACSLS after the audits are complete.
11. Vary the new LSMs online and resume normal automated processing.  
LSMs can be varied online as soon as the `audit` of that LSM completes.

## Adding the New ACS on the Right Hand side of the Split

If the new ACS is added on the right-hand of the split, all cartridge addresses will be re-mapped.

As this is not recommended, a detailed procedure is not provided.

However, some considerations, are:

- When the LSMs in the right-hand SL8500(s) are removed from the existing ACS, the cartridges in these LSMs will be marked absent (if absent cartridge retention is active). They will be reactivated when the new ACS is audited.
- The LSMs in pre-existing ACS must be audited one LSM at a time to update the addresses of the cartridges in these LSMs. Audit the highest numbered LSM first, followed by the next lower LSM. The audit of one LSM must finish before the next audit is started.

# D

## ACSL S Support of the SL3000

The SL3000 offers the following benefits:

- Storage capacity from 200 to 4500 storage cells.
- Performance from 1 to 56 tape drives.
- Support of mixed media and a variety of tape drive.
- Ability to partition an SL3000 at the storage cell and drive level.
- An option (with ACSLS 7.3 PUT0801 or later) to install an Access Expansion Module (AEM). This option is discussed under [Using the AEM](#) .

### ▲ Caution:

You must `audit` an SL3000 after it is configured or reconfigured to ACSLS. You must also `audit` an SL3000 after increasing or changing the activated capacity of the library.

The library reports the potential cell locations that are not accessible to ACSLS when ACSLS audits it. Inaccessible cell locations include: locations where CAPs, drives, and operator panels are installed; cell locations that the robots cannot access; cell locations that are not activated; and locations that are not in this partition.

## ACSL S Support

ACSL S provides support for the following SL3000 features:

- New LSM and panel types, and panel maps for the SL3000 Library
- Up to eight (8) partitions.

The SL3000 can partition down to the drive and cell level. If cells are reassigned from one partition to a different partition, cartridges in those cells will be orphaned, and they will no longer be accessible by the partition that they were in before. To avoid this, before re-partitioning an SL3000, move cartridges to cells that will remain in your partition.

- New maximum numbers of CAPs (12), drives per panel (32), and cell rows (52) for the SL3000.
- Ability to dedicate a CAP to a partition within the SL3000.

CAPS are dedicated to a partition using the SL Console. The library reports dedicated CAPs to ACSLS.

CAPS **dedicated** to a partition managed by ACSLS can be set to automatic mode.

- SL3000 CAP Aliasing

The SL3000 can have up to 12 CAPs, but some backup applications currently only support a maximum of 3 CAPs per LSM. CAP aliasing lets you alias CAPs 3-10 as CAPs 0, 1, or 2.

With PUT0801, you are no longer able to alias CAP 0. CAP 0 is reserved for the Access Expansion Module (AEM) and has special characteristics. Because of this, only CAP numbers 1 and 2 can be used as aliases.

- ACSLS 7.3 with PUT0801 supports AEMs for bulk enters and ejects.

## Before Configuring ACSLS for the SL3000

Before you configure ACSLS for the SL3000:

- Connect the SL3000 to ACSLS
- Verify that all of the components of the SL3000 are operational.

ACSLs builds its library configuration from the information reported by the library. If SL3000 components are not operational, the library information may not be reported to ACSLS, and the ACSLS configuration of the SL3000 could be incomplete.

## Connecting the SL3000 to ACSLS

The SL3000 library uses TCP/IP protocol over an Ethernet physical interface to manage and communicate with the host and ACSLS. This interface enables ACSLS to connect to, and communicate with the SL3000.

Using TCP/IP connection(s) to the SL3000 provide these advantages:

- ACSLS can send multiple concurrent requests to the library, and the library can process them in parallel. This improves library performance.
- When connected using TCP/IP, the SL3000 supports Dual TCP/IP and Redundant Electronics.
- The TCP/IP Host/Library Interface is a richer interface than the SCSI Media Changer over Fibre Channel. More information is available to ACSLS about library status, errors, and so forth.

It is preferable to have the SL3000 library on a separate subnet, or a controlled network to protect it from floods of ARP broadcasts.

## Verifying that all SL3000 Components are Operational

To verify that all of the components of the SL3000 are operational, logon to the StorageTek Library Console (SL Console).

Refer to the *SL3000 User's Guide* for procedures.

## Configuring SL3000 to ACSLS

Once the SL3000 components are operational, configure SL3000 to ACSLS. You can use either `acsss_config` (see [Setting CSI Tuning Variables](#)) or Dynamic Configuration (see [Using the Dynamic Configuration \(config\) utility](#)).



## Auditing the SL3000

Audit an SL3000 after it is configured or reconfigured to ACSLS. Auditing informs ACSLS which cells are part of the partition that it is managing in this library and displaying the cells that are actually present on panel drives or CAPs.

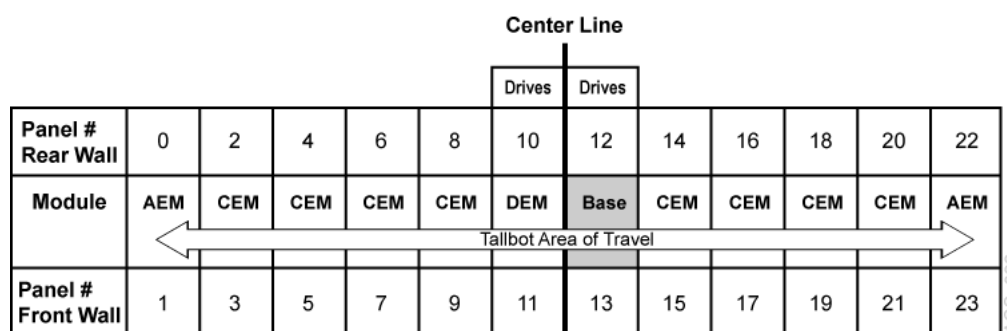
- First, configure or reconfigure an SL3000 to ACSLS.
- Audit the ACS.
  - `free_cells.sh -a`
  - `get_license_info`
  - `display cell * -f status`
- Look at the drives using either:
  - `query drive all`
  - `display drive * -f type serial_num wwn`

## SL3000 Addressing

An SL3000 can be expanded without changing any of the ACSLS addresses for drives and storage cells. Since ACSLS addresses map the largest possible SL3000 library, you can add modules without changing the addresses of existing cells, drives, and CAPs.

If modules are not installed, their panels are simply mapped as "not installed". If panels are installed, the addresses on these panels become available. In this way, when you install a new module, addresses of the panels on existing modules do not change.

**Figure D-1 SL3000 Addressing**



The smallest SL3000 configuration is a single Base Drive Module (BDM). In a single BDM configuration, panels 12 and 13 are installed, and all other potential panel locations are not installed. When you add a Drive Expansion Module (DEM), panels 10 and 11 are also installed. If you add a Cartridge Expansion Module (CEM) to the right of the BDM, panels 14 and 15 are now installed. The addresses of the cells and drives on the BDM never change. This ensures that as you increase your SL3000 capacity. The ACSLS addresses for drives and storage cells do not change.

## CAP Numbering

CAPs are numbered, as follows:

- CAP 0 is reserved for the AEM on the left
- CAPs 1-5 - in the CEMs on the left and the DEM
- CAP 6 - in the Base (the only required CAP)
- CAPs 7-10 - in the CEMs on the right
- CAP 11 is reserved for the AEM on the right

If CAPs are not present, ACSLS reports them as “not installed”. This keeps CAP locations from changing in SL3000 when new CAPs are added.

CAPs can be dedicated to a partition.

## SL3000 Modules

There are five types of modules in an SL3000 library:

- Base module, one, *required*

The Base Module consists of a single frame. This module centralizes the infrastructure for all other modules in the library, which includes the power configurations, robotics (TallBot), electronic modules, cartridge access port, storage cells, tape drives, and operator controls.

One Base Module, and only one, is required for every library installation.

On the front of this module is:

- A single, 26 cartridge-dual magazine-cartridge access port (CAP).
  - A service door for library access.
  - A front panel with three LEDs: Library Active, Service Required, and Wait.
  - An optional feature for a touch screen operator panel or window.
- Drive expansion module (DEM) one, *on the left-side, only*

A Drive Expansion Module (DEM) may be attached adjacent to the Base Module on the left-side, only. This module allows further expansion of tape drives and provides additional data cartridge capacity.
  - Cartridge expansion module (CEM) *left or right-side*

The Cartridge Expansion Module (CEM) provides additional cartridge cell capacity and growth—no tape drives are in this module. A maximum of four CEMs are supported on a single library in addition to the Base Module (required) and optional drive expansion module.
  - Parking expansion module (PEM) *leftmost* and the *rightmost side* modules.

The Parking Expansion Module (PEM) is the same as the cartridge expansion module, with six columns of arrays (three on the front wall and three on the rear wall) that are inaccessible to allow the library to park a defective robot. In the left PEM, it is the three leftmost columns (front and back). In the right PEM, it is the three rightmost columns.

 **Note:**

An SL3000 with dual robots has either AEMs or PEMs

- Access Expansion Module (AEM)  
AEM is discussed under [Using the AEM](#) .

## New Panel Types

Instead of having specific panel definitions in the SL3000 that define each possible cell location, there are generic panel definitions.

The three generic panel types map to a variety of specific panel maps. They are:

- A single SL3000 cell panel type.  
To find out which cells are present in this library and which are part of the activated capacity and are assigned to this partition, you must `audit` the library.
- A drive panel type.  
ACSLs identifies which drives are present by configuring the library. Before you do this, you must make sure all drives are powered-on and ready before configuring the library.
- Not installed.  
This is a placeholder for the front and rear panels on modules that are not yet installed.

## Understanding Internal SL3000 Addressing

There are differences in the internal addresses of the SL3000 and other libraries supported by ACSLS and HSC.

- The SL3000 is *one's-based* (1) and uses negative numbers.
- Other libraries use a *zero-based* (0) with *no* negative numbers.
- The SL3000 uses *five* parameters: library, rail, column, side, and row.
- Other libraries use: ACS, LSM, panel, row, and column (HLI-PRC).

The modules of the library consists of walls, columns, and rows that house cartridges, tape drives, cartridge access ports, and robotic units.

You can use the SL Console to translate between the SL3000 internal address and the ACSLS panel, row, and column.

Refer to the *SL3000 User's Guide* for detailed information and procedures.

## Using the AEM

The AEM (access expansion module) is essentially a very large CAP that has all of the characteristics of a CAP, such as online/offline states and the ability to be shared by partitions.

The AEM allows you to perform bulk loads or unloads of up to 234 cartridges at a time, to or from the library. It also allows for the non-disruptive maintenance of a library TallBot.

AEMs share many of the features and functions as rotational CAPs, but they also have some unique characteristics. They are:

- Access Door
- Safety Door
- CAP Operations
- Non-Disruptive Maintenance

For detailed information on the AEM, refer to the *StorageTek SL3000 Modular Library System User's Guide*.

## Access Door

The AEM access door is the external door you open to bulk load or unload cartridges to or from the library.

## Safety Door

The safety door is the internal “garage door,” which lowers to separate the AEM from the rest of the library. This door allows for safe access to the inside of the AEM, either for the library operator to load or unload cartridges, or for a Storage CSE to perform service on the Tallbot or other AEM components.

## CAP Operations

The advantages of the AEM is that it allows you to perform the following:

- Bulk enters

Bulk enters enables to you to enter up to 234 cartridges at a single time.

You must start a normal enter specifying the AEM `cap_id`. This can either be a manual enter or you can set the AEM to automatic enter mode.

In a partitioned library, you can only set a CAP to automatic enter mode if the CAP is dedicated to your partition. Dedicating an AEM to only your partition makes it unavailable to other partitions. Consider other SL3000 users before doing this.

- Bulk ejects

When ejecting more that 42 volumes to the AEM, the preferred tool for larger ejects is `lib_cmd` or `ejecting.sh`. Any other user interface, including `cmd_proc`, `xject`, or an ACSAPI eject will limit the eject load to 42 cartridges at a time. See [Using lib\\_cmd](#) and [ejecting.sh](#).

- Non-disruptive robot maintenance

### **Caution:**

When performing a bulk eject using the AEM, ensure all CAP magazines are present. The `eject` fails if there are missing CAP magazines at locations where ACSLS is attempting to `eject` cartridges.

 **Note:**

The AEM should not be used for entering or ejecting small numbers of cartridges. It should only be used for bulk enters and ejects. Using the AEM for small enters and ejects is much slower since the service safety door must be lowered when the AEM is accessed to enter or remove volumes, and raised when access to the AEM is completed.

For smaller numbers, use CAPs 1-10. See [CAP Numbering](#).

Since the AEM should not be used for small enters and ejects, the maximum CAP priority is 1. This aids in preventing the AEM from being selected when the CAP ID is wildcarded with an asterisk on an `audit`, `enter`, or `eject`.

## Non-Disruptive Maintenance

The AEM allows the CSE access and service a library TallBot through the use of a safety door (or "garage" door), without having to take the library offline. If the TallBot is defective, it parks itself in the AEM while the library remains online. If redundant TallBots are installed, the library can continue normal operations through the remaining functional TallBot.

The CSE uses a special key to unlock the Service Access lock on the safety door. This causes the AEM safety door to lower, sectioning off the AEM from the rest of the library. Once the safety door is fully down, the CSE opens the AEM safety door and safely performs maintenance on the defective TallBot or other AEM components, while the library remains online. The AEM itself is taken offline during the maintenance period, causing cartridge load/unload functions to be suspended.

When maintenance is complete, the CSE: closes the safety door; the safety door raises; the TallBot is re-initialized and performs an `audit` to the AEM CAP cartridge slots; and the AEM is brought back online.

## Aliasing SL3000 CAP IDs

The SL3000 has 12 CAPs, but some backup applications currently only support a maximum of 3 CAPs per LSM. CAP aliasing lets you alias CAPs 3-10 as CAPs 1 or 2.

Cap 0 is reserved for the Access Expansion Module (AEM) bulk load CAP. Since CAP 0 in an SL3000 has special attributes, it cannot be an alias CAP ID.

Map the Library CAP IDs to alias CAP IDs by using the following template.

```
$ACS_HOME/data/external/SL3000/SL3000_CAP_Aliases.SAMPLE
```

 **Note:**

When setting CAP IDs, for Library CAP IDs:

For Library CAP IDs:

- LSM must be zero (0)
- ID must be 3-10
- There can be no duplicate library CAP IDs

For Alias CAP IDs:

- CAP number must be 1 or 2
- There can be no duplicate alias CAP ID

```
# SL3000 CAP Aliases (SL3000_CAP_Aliases)
#
# This file maps real library SL3000 CAP IDs to CAP ID aliases.
# It is used when ACSAPI clients do not support SL3000
# CAP numbers greater than 2.
# (The SL3000 can have 12 CAPs, with a maximum CAP number of 11.)
# Alias a SL3000 CAP to an alias CAP number by providing the full
# SL3000 CAP ID and the aliased CAP number on the same line.
# For example:
# SL3000 CAP ID      Alias CAP Number
#           0,0,6           1
#
# The LSM must be zero.
# The CAP ID must be between 3 and 10
# No duplicate SL3000 CAP IDs.
# Rules for SL3000 CAP IDs:
# Rules for Alias CAP numbers:
# The CAP number must be 1 or 2.
# No duplicate alias CAP numbers.
# # Using the CAP Aliases file:
# (1) A # in the first column comments out a line.
# The above CAP alias example is commented out.
# (2) Only specify CAP numbers as an alias when there
# is no actual CAP at that location. For example,
# use CAP 2 as an alias when there is no CAP 2
# installed in the SL3000 library.
# (3) Only CAP numbers 1 and 2 can be used as aliases.
# NOTE: CAP number 0 (zero) is reserved for the
# Access Expansion Module (AEM) and thus cannot
# be specified as an alias.
# (4) After updating the CAP Aliases file, you must:
# a) Shutdown ACSLS.
# b) Reconfigure ACSLS (using acsss_config) to update
# the CAPs defined in the ACSLS database.
# c) Restart ACSLS.
# Save the CAP alias file as:
# $ACS_HOME/data/external/SL3000/SL3000_CAP_Aliases
# by copying and modifying the template:
# $ACS_HOME/data/external/SL3000/SL3000_CAP_Aliases.SAMPLE
#
# SL3000 CAP ID      Alias CAP Number
#           0,0,6           1
#           0,0,4           2
```

```

# Alias a SL3000 CAP to an alias CAP number by providing the full
# SL3000 CAP ID and the aliased CAP number on the same line.
# For example:
#   SL3000 CAP ID      Alias CAP Number
#           0,0,6              1
#
# Rules for SL3000 CAP IDs:
#   The LSM must be zero.
#   The CAP ID must be between 3 and 10
#   No duplicate SL3000 CAP IDs.
# Alias a SL3000 CAP to an alias CAP number by providing the full
# SL3000 CAP ID and the aliased CAP number on the same line.
# For example:
#   SL3000 CAP ID      Alias CAP Number
#           0,0,6              1
#
# Rules for SL3000 CAP IDs:
#   The LSM must be zero.
#   The CAP ID must be between 3 and 10
#   No duplicate SL3000 CAP IDs.

```

The example above shows a modified `SL3000_CAP_Aliases` file. The comment (#) character has been removed from the lines shown in bold to specify that alias CAP number assignments are enabled. In this example, the:

- physical CAP ID 0,0,4 is aliased to CAP number 1.
- physical CAP ID 0,0,6 is aliased CAP number 2.

## Moving Cartridges Before Removing Cells from a Partition

The SL3000 can partition down to the drive and cell level. For more information, see [Moving Cartridges Before Removing Cells from a Partition](#).

## SL3000 CAP Behavior

The SL3000 supports up to 12 Cartridge Access Ports (CAPs). There can be a CAP on each SL3000 panel.

For CAP behavior in a partitioned library, refer to [Partitioning a Library or Changing Partition IDs](#).

### Caution:

All ACSs must have at least one CAP. This can be **dedicated** or is shared. Because all of the CAPs in an SL3000 can be dedicated to other partitions, this is a crucial restriction when configuring SL3000 partitions.

## Finding Missing SL3000 Cartridges

If a cartridge is out of place or unaccounted for by ACSLS:

1. Perform a physical audit of the SL3000 using the SL Console.

The physical audit of the SL3000 is performed as a background task in between handling `mount` and other library operation requests.

 **Caution:**

If the SL3000 contents are out of sync with ACSLS due to manual operations such as loading cartridges directly, it is not advisable to attempt continued operations.

2. Run an ACSLS `audit` to update the ACSLS database to match the actual inventory of library cartridges.

## Varying the SL3000 Offline

Vary SL3000 components offline to ACSLS before they are powered off, if they are inoperative, and before you open an SL3000 access door. This notifies ACSLS that they are unavailable. Once they are available, vary them back online.

### Use ACSLS to vary SL3000 components offline, not SL Console

Vary SL3000 components (ACSSs, LSMs, and CAPs) offline to ACSLS, not the SL Console.

ACSLs allows outstanding requests to complete before taking components offline, unless it is a `vary offline force`. The SL Console has no knowledge of outstanding requests to ACSLS.

Varying components offline using SL Console may cause requests in progress to fail.

### When to vary SL3000 components offline to ACSLS

This section describes when to vary components offline to ACSLS.

#### Before Opening the Access Door

Before opening the SL3000 access door, issue the following command to vary the ACS offline:

```
vary acs acs_id offline
```

 **Note:**

If any CAPs in the SL3000 are in automatic mode, you must:

1. Set the CAPs to manual mode before opening the access door.
2. Set the CAPs back to automatic mode **after** you close the access door and the SL3000 comes back online.



## If a CAP is inoperative

If a CAP is inoperative, issue the following command to vary it offline:

```
vary cap cap_id offline
```

## Using the Dynamic Configuration (config) utility

The dynamic configuration (`config`) utility allows you to implement configuration changes to ACSLS libraries (and components) while ACSLS remains online and running. These configuration changes are recorded in the `acsss_config.log` file.

The following dynamic configuration utilities are supported:

- `config acs`
- `config drives`
- `config lsm`
- `config ports`

Using the `config` utility provides the following benefits:

- ACSLS can continue running, allowing you to perform `mount` requests to unaffected library components.
- Allows you to reconfigure specified library components while all other configuration information remains unchanged. For example, when specifying:
  - An ACS, the configurations of other ACSs are not affected.
  - An LSM, the configurations of other LSMs are not affected.
  - A drive panel (the drives on a panel) mounts and dismounts to all existing drives are not affected.

## Auditing New Cartridge Addresses

After an expansion has occurred, complete the following procedure:

1. Vary the ACS and ports offline.
2. Do one of the following:
  - Add or remove a module(s)
  - Change capacity
  - Re-partition the library
3. Vary the ACS and ports online.
4. Update the configuration in the ACSLS database using one of the following procedures.
  - Dynamic:
    - `config acs acs_id` or `config lsm lsm_id`
    - Audit the library.
  - Static:

- Bring down ACSLS: `acsss disable`
- Update the ACSLS configuration: `acsss_config`
- Bring up ACSLS: `acsss enable`
- Audit the library.

## ACSLs Dual TCP/IP

Dual TCP/IP connections between the ACSLS server and the library is an option that can be purchased for the SL3000.

### SL3000 Requirements

- Coordinate with both your system administrator and network administrator to understand your current network environment, and to identify all necessary IP addresses in advance.
- Coordinate with your system administrator to either configure your network interface or to validate that it is configured properly.

See [Overview](#) for more information on this feature, along with the *StreamLine Modular Library System Dual TCP/IP Feature* document.



#### Note:

The procedure for implementing Dual TCP/IP connections is the same for both SL8500 and SL3000 libraries.

# E

## ACSLs Support of the SL150

The SL150 library is a SCSI-attached, single-LSM library. This chapter discusses ACSLS support for the SL150 library.

### ACSLs Connectivity

The ACSLS server needs an FC Host-Bus Adapter (HBA) that is compatible with the bridged drive in the SL150 library.

When you run `$ACS_HOME/install/install_scsi_sol.sh`, the system is probed for the existence of new hardware: either a JNI card or an LSI Logic card. If either of these HBA cards is present on the system, but the corresponding driver is not in place, the routine displays a prompt that asks you for permission to locate the appropriate driver package and install it.



#### Note:

- Although the SL150 can be partitioned, ACSLS does not support partitioned SL150s.
- SAS bridged drives for control path are not supported. Only fibre-channel connections to bridged drives are supported.

### ACSLs and SL150 Library Differences

This section discusses ACSLS and SL150 library differences.

#### Library Configuration

- ACS, LSM - SL150 library is an ACS with a single LSM (1 ACS and 1 LSM).
- SL150 does not currently support the pass-thru port.
- SL150 has at least one base module, and can have up to nine expansion modules.
- ACSLS considers each SL150 module as a panel.
- SL150 libraries managed by ACSLS reports 6-character `volser` labels through ACSLS interface.

#### Identifying Library Locations

This section discusses library locations.

## Addressing schemes

- ACSLS addressing is ACS, LSM, panel, row, column.
- SL150 addressing is module, sides, row, column for storage cells.
- ACSLS addresses are zero-based. They start with panel 0, row 0, and column 0.
- SL150 libraries are one-based. They start with module 1, row 1, and column 1.
- In order to ensure consistency between the SL150 library and ACSLS internal addresses, ACSLS defines placeholders for panel 0, row 0, and column 0.

## Panels

- panel 0 = not installed
- panel 1 = base module
- panels 2-10 = expansion modules

## Row Numbers

- Within each module (panel), rows are numbered 1-3.
- Rows are numbered consecutively from the top down.
- Numbering is identical between the SL150 and ACSLS.

## Column numbers

### **SL150 Addressing:**

- module, side, row, column
- Columns 1-5, front to back on each side

### **ACSLS Addressing:**

- Numbered from 1 to 10.
- Columns 1-5 are on the left side from front to back.
- Columns 6-10 on are the right side from front to back.

## Drive addresses

### **SL150 addressing:**

- module, position
- module 1 = base
- modules 2-10 = expansions
- position = top to bottom

### **ACSLS addressing:**

- ACS, LSM, panel, drive
- Panel = module

- Drive number = 1-20, numbered from top to bottom

**Table E-1 Example of drive addressing**

| Module           | ACSLS Drive Identifier | SL150 Drive Address |
|------------------|------------------------|---------------------|
| Base module      | 0, 0, 1, 1             | 1, T                |
| -                | 0, 0, 1, 2             | 1, B                |
| Expansion module | 0, 0, 2, 3             | 2, T                |
| -                | 0, 0, 2, 4             | 2, B                |
| Expansion module | 0, 0, 3, 5             | 3, T                |
| -                | 0, 0, 3, 6             | 3, B                |
| Expansion module | 0, 0, 4, 7             | 4, T                |
| -                | and so on.             | and so on.          |

## ACSLS Limitations

- Maximum number of SL150 libraries = 31.
- Maximum number of LSMs = 127.
- Maximum number of modules (panels) in an SL150 library = 10.
- Expansion modules add one panel each.
- An SL150 with only the base module has just one panel.
- The SL150 firmware does not support unlabeled cartridges. This means that the `venter` command cannot be used.
- Partitioned SL150 libraries are not supported.
- SAS bridged drives for control path are not supported.

## Configuring SL150 Library Settings

The SL150 supports configuration settings that affect ACSLS.

- Partitioning  
Partitioned SL150 libraries are not supported by ACSLS.
- Bridged Drives  
ACSLS supports only FC control path. SAS bridged drives for control path are not supported.
- Configure the Mailslot  
Enables you to use the storage slots on one side of the base module as an extension of the standard mailslot.
  - Required setting for ACSLS: Standard Mailslot (4 slots)
- Drive Element Addressing Mode  
Enables you to control how SCSI element addresses are reported when empty drive slots are present in the library.

- Required setting for ACSLS: Address Only Installed Drives
- Reserved Cells  
The reserved cells are for library use only (diagnostic and cleaning cartridges):
  - Number (n) can be configured from 0 to 3 (maximum rows in base module).
  - Only affects base module - the first (n) cells in column 1 are reserved.
  - These cells are considered inaccessible to ACSLS
- Library Volume Label Format  
The SL150 library supports a variety of label format types (6-character, 8-character, etc.). For ACSLS operations supporting client operations, the default library setting is appropriate.
  - Required setting for ACSLS: Trim last two characters (default)

## SL150 CAP Behavior

The SL150 uses the term “mailslot” for the CAP. The mailslot in the base module has one four-slot magazine.

All mailslot operations are performed using either the library touch screen or the BUI.

- The SL150 must have one active mailslot for entering and ejecting cartridges.
- When ACSLS audits the mailslot, it examines all cells.

## CAP (Mailslot) Not Opening During an Eject

It is possible for problems to occur during an `eject` (or `enter`) operation if the SL150 CAP (mailslot) has been left locked by an instance of ACSLS that was previously managing the library. Scenarios in which this could occur include migration to a new ACSLS server, or an HA fail-over event.

When the mailslot has been locked by a host, the library touch screen display shows the following:

State: Locked by SCSI Prevent Media

The problem is less likely to be seen with a CAP in automatic mode, since ACSLS typically does not keep an auto-CAP locked. The problem is more likely to occur with a manual-mode CAP, or with any CAP (either mode) in a library that has been varied offline.

Mitigation:

To avoid encountering this problem, the following procedures are recommended.

- When migrating control of an SL150 library to a new ACSLS server (non-HA configurations):
  - Automatic mode CAP (recommended):  
Do not vary the library offline before shutting down ACSLS on the original server. ACSLS leaves the CAP unlocked at shutdown when the library is online.
  - Manual mode CAP:

Before shutting down ACSLS on the original server, set the CAP to operate in automatic mode. Do not vary the library offline.

- When managing the library in an ACSLS HA installation, operating the CAP in automatic mode is strongly advised.
  - Automatic mode CAP (recommended):

No action should be needed for an online library (the CAP is normally unlocked). If the library was offline at the time of a fail-over event, follow the steps below to unlock the CAP by restarting the bridged drive.
  - Manual mode CAP:

Follow the steps below to unlock the CAP by restarting the bridged drive.

Resolution:

A lock left by a previous host can be cleared by restarting the bridged drive using the SL150 BUI. Refer to the SL150 product documentation for information about restarting a bridged drive.

 **WARNING:**

This operation is disruptive to any current data path operation on the drive. You can schedule restarting the bridged drive for a time when it will not interrupt data path activity.

In the event that this problem has occurred, and the lock must be cleared:

1. Ensure that no data path (read/write) activity is occurring on the bridged drive.
2. Restart the bridged drive using the SL150 BUI.

## Auditing the Library

Perform an audit, if:

- This is a new library.
- One or more modules have been added, removed, or swapped.
- Cartridges have been manually added or removed through the mailslot.
- After library configuration settings have changed.

A self-audit of the library occurs when you:

- Power-on or re-initialize the library.
- Open and close the mailslot.

After the self-audit is complete, you must update your database using ACSLS.

## When this is a new library

A new library must be audited to synchronize the actual contents of the library with the database.

## After adding, removing, or swapping modules

After an SL150 module has been added, removed, swapped, or if you are auditing the library for the first time, perform the following steps:

1. Vary the SL150 library offline to ACSLS and power down the library.
2. Perform the library change, such as adding a module.
3. Power on or re-initialize the SL150 library.
4. Terminate ACSLS (if it is running) by using the following command:

```
acsss disable
```

5. Run `acsss_config`.
  6. Start up ACSLS by using the following command:
- ```
acsss enable
```
7. Perform an `audit` of the SL150 panel(s) affected.

## After Manually Adding or Removing Cartridges Through the Mailslot

If you have manually added or removed cartridges from the LSM through the library mailslot, you must synchronize the databases. Perform an `audit` as described under [query pool](#).

If you have entered or ejected cartridges from the LSM through the mailslot using the ACSLS `cmd_proc`, the database is automatically updated.

## After Changing Library Configuration Settings

If you have changed library configuration settings, you should reconfigure ACSLS using `acsss_config`, and then `audit` the library. You should also reconfigure and `audit` the library after changing the number of reserved cells

Follow these steps:

1. Vary the SL150 library offline to ACSLS, and power down the library.
2. Perform the library change.
3. Re-initialize the SL150 library.
4. Terminate ACSLS (if it is running) by using the following command:

```
acsss disable
```

5. Run `acsss_config`.
  6. Start up ACSLS by using the following command:
- ```
acsss enable
```
7. Perform an `audit` of the SL150 panel(s) affected.



# F

## ACSL S Support of the SL4000

Oracle's SL4000 library is the successor to the SL3000 library. It is physically similar to the SL3000 and retains many of the same features. The SL4000 offers the following benefits:

- Higher storage capacity (from 200 to 9,017 storage cells)
- Drive capacity from 1 to 120 tape drives (up to three DEMs)
- Secure, authenticated library communications using HTTPS protocol
- Support for dual AEMs to support bulk loading and redundant robotics
- Support for a variety of mixed media and tape drives
- Ability to partition an SL4000 at the storage cell and drive level

### **Caution:**

You must `audit` an SL4000 after it is configured or reconfigured to ACSLS. You must also `audit` an SL4000 after increasing or changing the activated capacity of the library.

The library reports the potential cell locations that are not accessible to ACSLS when ACSLS audits it. Inaccessible cell locations include: locations where CAPs, drives, and operator panels are installed; cell locations that the robots cannot access; cell locations that are not activated; and locations that are not in this partition.

## ACSL S Support

ACSL S provides support for the following SL4000 features:

- ACSLS supports the new LSM type SL4000 which is visible in queries and displays. The SL4000 library is managed and operated as a single-LSM ACS.
- ACSLS uses the same generic panel maps for both SL3000 and SL4000 libraries.
- ACSLS can manage up to 16 partitions of a single SL4000 library.
- The maximum number of CAPs in one LSM is increased to 15.
  - With AEMs, a maximum of 15 CAPs
  - With parking modules, a maximum of 14 CAPs
- Ability to dedicate a CAP to a partition within the SL4000.

Use the SL4000 GUI to dedicate CAPs to an SL4000 partition. The library reports both dedicated and shared CAPs to ACSLS. Only CAPs that are dedicated to a partition managed by ACSLS can be set to automatic mode.
- The SL4000 does not currently provide a CAP Aliasing feature.

## Before Configuring ACSLS for the SL4000

Before you configure ACSLS for the SL4000:

- Verify that all SL4000 components are operational.

ACSLs builds its library configuration from the information reported by the library. If SL4000 components are not operational, the library information may not be reported to ACSLS, and the ACSLS configuration of the SL4000 could be incomplete.

- Verify that required library configuration is complete.

ACSLs will require a known user name and password that have been defined on the SL4000 library. Any partitioning should also be performed, and if ACSLS will be managing a partition then it must not be a SCSI partition. The partition number will be required during library configuration.

To verify that all of the components of the SL4000 are operational, or to complete any required library set-up, log in to the SL4000 Web-based GUI. Refer to the *SL4000 User's Guide* for procedures.

## Configuring the SL4000 to ACSLS

Once the SL4000 components are operational, use one of the following ACSLS utilities to configure the SL4000 to ACSLS:

- `acsss_config` utility (see [Using acsss\\_config to Configure Your Library Hardware](#))
- Dynamic Configuration (`config`) utility (see [Using the Dynamic Configuration \(config\) utility](#))

ACSLs uses the SCI interface (HTTPS communications, including authentication) to communicate with, and operate the SL4000 library.

Refer to the `acsss_config` and `config` utilities for details on how to specify connection parameters for various library types, including the SL4000.

### Note:

- Two or more ACSLS servers cannot be connected to an SL4000 library. Though ACSLS does not enforce any limits, connecting two or more servers to an SL4000 will result in database performance and cartridge management issues.
- For proper functionality, ensure that the ACSLS server clock is in sync with the SL4000 internal clock, within a few minutes.
- ACSLS SCI connection to an SL4000 library requires an SL4000 user credential with a user role at the User level. The SL4000 Administrator role can also be used for this credential.

## Changing the SL4000 User Name and Password

If the SL4000 user name or password changes or expires, you must use the `acs_rewallet.sh` utility to update ACSLS authentication. See [acs\\_rewallet.sh](#) for more information.

## SL4000 Modules

There are five types of modules in an SL4000 library, with a total of 15 modules allowed in a single library.

### Note:

Refer to your SL4000 library publications for the latest updates to configuration and feature availability, as needed.

- Base module, one, *required*

This module consists of a single frame, which centralizes the infrastructure for all other modules in the library. One Base Module, and only one, is required for every library installation.

On the front of this module is:

  - A single, 26 cartridge-dual magazine-cartridge access port (CAP).
  - A service door for library access.
  - A front panel with three LEDs: Library Active, Service Required, and Wait.
  - An optional feature for a touch screen operator panel or window.
- Drive Expansion Module (DEM), maximum of three

This module allows further expansion of tape drives and provides additional data cartridge capacity. A maximum of three Drive Expansion Modules can be added to the required Base Module, on any side of the base module.
- Cartridge Expansion Module (CEM), maximum of twelve

This module provides additional cartridge cell capacity and growth. It does not include tape drives. A maximum of twelve Cartridge Expansion Modules can be added, though the maximum number is dependent on the number of other module types present.
- Parking Expansion Module (PEM), *leftmost* and the *rightmost side* modules

These are the left-most and the right-most side modules. They are the same as the cartridge expansion module, with six columns of arrays (three on the front wall and three on the rear wall) that are inaccessible to allow the library to park a defective robot. In the left PEM, it is the three leftmost columns (front and back). In the right PEM, it is the three rightmost columns.
- Access Expansion Module (AEM)

See [Using the AEM](#) for more information.

**Note:**

An SL4000 with dual robots has either AEMs or PEMs.

## SL4000 Panel Types

ACSLs uses the same panel types for both SL3000 and SL4000 libraries. Instead of having specific panel definitions that define each possible cell location, the SL4000 uses generic panel definitions.

The three generic panel types map to a variety of specific panel maps. They are:

- Single SL4000 cell panel type  
To determine which cells are present in this library and which cells are part of the activated capacity and are assigned to this partition, you must audit the library.
- Drive panel type  
When you configure the library, ACSLS determines which drives are present. Before you configure the library, make sure all drives are powered on and ready.
- Not installed  
This is a placeholder for the front and rear panels on modules that are not yet installed.

## Understanding SL4000 Addressing

SL4000 library modules consist of walls, columns, and rows that house cartridges, tape drives, cartridge access ports, and robotic units.

Like the SL3000, the SL4000 uses internal addresses that are somewhat different than the traditional PRC (panel, row, column) addresses used in ACSLS to map physical locations in a library.

- SL3000 and SL4000 addresses are one-based (not 0) and also use negative numbers.
- Library addresses use five parameters: library, rail, column, side, and row.

As with other libraries (SL8500, SL150), ACSLS provides a mapping of physical locations into PRC addresses (acs, lsm, panel, row, column) for administrative and client usage. See [SL4000 Addressing](#) for more information about the panel numbering used for an SL4000 library.

Refer to your SL4000 library publications for detailed information about physical configuration and addressing, and for any other procedures related to library operations.

## SL4000 Addressing

An SL4000 can be expanded without changing any of the ACSLS addresses for drives and storage cells. Since ACSLS addresses map the largest possible SL4000 library, you can add modules without changing the addresses of existing cells, drives, and CAPs.

If modules are not installed, their panels are simply mapped as "not installed". If panels are installed, the addresses on these panels become available. In this way, when you install a new module, addresses of the panels on existing modules do not change as long as those panels' positions do not change with respect to the Base Module.

The following figure shows SL4000 potential module and panel numbers:

**Figure F-1 SL4000 Addressing**

|                               |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |  |  |  |
|-------------------------------|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|--|--|--|
|                               |    |    |    |    |    |    |    | Base |    |    |    |    |    |    |    |  |  |  |
| <b>Panel #<br/>Rear Wall</b>  | 16 | 18 | 20 | 22 | 24 | 26 | 28 | 30   | 32 | 34 | 36 | 38 | 40 | 42 | 44 |  |  |  |
| <b>Module</b>                 | 8  | 9  | 10 | 11 | 12 | 13 | 14 | 15   | 16 | 17 | 18 | 19 | 20 | 21 | 22 |  |  |  |
| <b>Panel #<br/>Front Wall</b> | 17 | 19 | 21 | 23 | 25 | 27 | 29 | 31   | 33 | 35 | 37 | 39 | 41 | 43 | 45 |  |  |  |

ACS\_066

Make sure the site meets the following configuration requirements prior to beginning installation:

- The library must not exceed 15 modules.
- The maximum number of modules to the right of the Base is 7.
- The maximum number of modules to the left of the Base Module is 7.
- The Base Module must be centered in the middle of 15 module libraries.
- The library must not exceed 3 Drive Expansion Modules (DEM).

The smallest SL4000 configuration is a single Base Module (BM). In a single BM configuration, panels 30 and 31 are installed, and all other potential panel locations are not installed. When you add a Drive Expansion Module (DEM) on the left, panels 28 and 29 are also installed. If you add a Cartridge Expansion Module (CEM) to the right of the BM, panels 32 and 33 are now installed. The addresses of the cells and drives on the BM never change.

## CAP Numbering

The SL4000 supports up to 15 Cartridge Access Ports (CAPs). There can be a CAP on each SL4000 panel. CAPs are numbered, as follows:

- CAPs 1-14 - in the CEMs or DEMs to the left of the Base Module
- CAP 15 - in the Base Module (the only required CAP)
- CAPs 16-29 - in the CEMs or DEMs to the right of the Base Module

Just as panel numbers are added counting either down or up from the Base Module, CAP numbers are added counting either up or down from CAP 15 in the Base Module. When a CAP is installed in a module, the CAP number will match the module number, remembering that the Base Module (BM) is always module 15.

 **Caution:**

Each ACS must have at least one CAP. This can be either dedicated or shared. Because all of the CAPs in an SL4000 can be dedicated to other partitions, this is a critical restriction when configuring SL4000 partitions.

## Varying the SL4000 Offline

Vary SL4000 components offline to ACSLS before they are powered off, if they are inoperative, and before you open an SL4000 access door. This notifies ACSLS that they are unavailable. Once they are available, vary them back online.

### Use ACSLS to Vary SL4000 Components Offline, not SL4000 GUI

Be sure to vary SL4000 components (ACSSs, LSMs, and CAPs) offline to ACSLS, not the SL4000 GUI. ACSLS allows outstanding requests to complete before taking components offline, unless it is a vary offline force.

The SL4000 GUI has no knowledge of outstanding requests to ACSLS. Varying components offline using the SL4000 GUI may cause requests in progress to fail.

### When to Vary SL4000 Components Offline to ACSLS

This section describes when to vary components offline to ACSLS.

#### Before Opening the Access Door

Before opening the SL4000 access door, issue the following command to vary the ACS offline:

```
vary acs acs_id offline
```

 **Note:**

If any CAPs in the SL4000 are in automatic mode, you must:

1. Set the CAPs to manual mode before opening the access door.
2. Set the CAPs back to automatic mode **after** you close the access door and the SL4000 comes back online.

#### If a CAP is Inoperative

If a CAP is inoperative, issue the following command to vary it offline:

```
vary cap cap_id offline
```

## Using the Dynamic Configuration (config) utility

The dynamic configuration (`config`) utility allows you to implement configuration changes to ACSLS libraries (and components) while ACSLS remains online and running. These configuration changes are recorded in the `acsss_config.log` file.

The following dynamic configuration utilities are supported:

- `config acs`
- `config drives`
- `config lsm`
- `config ports`

Using the `config` utility provides the following benefits:

- ACSLS can continue running, allowing you to perform `mount` requests to unaffected library components.
- Allows you to reconfigure specified library components while all other configuration information remains unchanged. For example:
  - When you specify an ACS, the configurations of other ACSs are not affected.
  - When you specify an LSM, the configurations of other LSMs are not affected.
  - When you specify a drive panel (the drives on a panel), mounts and dismounts to all existing drives are not affected.

### Note:

Dynamic configuration can be used to configure a new ACS, or re-configure an existing ACS. You cannot use dynamic configuration to delete an SL4000 ACS.

## Moving Cartridges Before Removing Cells from a Partition

The SL4000 can partition down to the drive and cell level. If cells are reassigned from one partition to a different partition, cartridges in those cells will be orphaned, and they will no longer be accessible by the partition that they were in before.

To avoid this, before re-partitioning an SL4000, move cartridges to cells that will remain in your partition.

## Finding Missing SL4000 Cartridges

If a cartridge is out of place or unaccounted for by ACSLS:

1. Perform a physical audit of the SL4000 using the SL4000 GUI.

The physical audit of the SL4000 is performed as a background task in between handling `mount` and other library operation requests.

**▲ Caution:**

If the SL4000 contents are out of sync with ACSLS due to manual operations such as loading cartridges directly, it is not advisable to attempt continued operations.

2. Run an ACSLS `audit` to update the ACSLS database to match the actual inventory of library cartridges.

## Auditing New Cartridge Addresses

After an expansion has occurred, complete the following procedure:

1. Vary the ACS and ports offline.
2. Do one of the following:
  - Add or remove a module(s).
  - Change capacity.
  - Re-partition the library.
3. Vary the ACS and ports online.
4. Update the configuration in the ACSLS database using one of the following procedures:
  - Dynamic:
    - `config acs acs_id` or `config lsm lsm_id`
    - Audit the library.
  - Static:
    - Bring down ACSLS: `acsss disable`
    - Update the ACSLS configuration: `acsss_config`
    - Bring up ACSLS: `acsss enable`
    - Audit the library.

## Unknown Cell Information Reported When Auditing the SL4000 Library

In some scenarios, while the SL4000 is auditing, cell information may be reported to ACSLS as UNKNOWN. This can cause issues when ACSLS is mounting, dismounting or auditing an SL4000 library.

One known scenario is after a SL4000 library door has been opened and closed. The SL4000 will become operational, as well as the ACSLS LSM for that ACS connected to the SL4000 library. However, the library itself may still be auditing after becoming operational and will report cell information to ACSLS as UNKNOWN. This may cause mount and dismount requests to fail. If this occurs, wait for the SL4000 to stop auditing and then re-attempt the operation again.



## Bulk Operations with the AEM

This section describes bulk operations with the Access Expansion Module (AEM).

The AEM is essentially a very large CAP that has all of the characteristics of a CAP, such as online/offline states and the ability to be shared by partitions.

The AEM allows you to perform bulk loads or unloads of up to 234 cartridges at a time, to or from the library. It also allows for the non-disruptive maintenance of a library TallBot.

For more information about the AEM, see [Using the AEM](#) .

## Bulk Enters

Bulk enters on an AEM enable to you to enter up to 234 cartridges at a single time. You must start a normal enter specifying the AEM `cap_id`. This can either be a manual enter or you can set the AEM to automatic enter mode.

In a partitioned library, you can only set a CAP to automatic enter mode if the CAP is dedicated to your partition. Dedicating an AEM to only your partition makes it unavailable to other partitions. Consider other SL4000 users before doing this.

## Bulk Ejects

When ejecting more than 42 volumes to the AEM, the preferred tool for larger ejects is `lib_cmd` or `ejecting.sh`. Any other user interface, including `cmd_proc`, `xreject`, or an ACSAPI eject will limit the eject load to 42 cartridges at a time.

See [Using lib\\_cmd](#) and [ejecting.sh](#) .

### **Caution:**

When performing a bulk eject using the AEM, ensure all CAP magazines are present. The eject fails if there are missing CAP magazines at locations where ACSLS is attempting to eject cartridges.

The AEM should not to be used for entering or ejecting small numbers of cartridges. It should only be used for bulk enters and ejects.

Using the AEM for small enters and ejects is much slower since the service safety door must be lowered when the AEM is accessed to enter or remove volumes, and raised when access to the AEM is completed. For smaller numbers of cartridges, use rotational CAPs.

Since the AEM should not be used for small enters and ejects, the maximum CAP priority is 1. This aids in preventing the AEM from being selected when the CAP ID is wild-carded with an asterisk on an audit, enter, or eject.

## Aliasing SL4000 CAP IDs

The SL4000 does not currently provide a CAP Aliasing feature.

## SL4000 OCSI Retention Time

The OSCI retention time determines how long the SL4000 library retains events on the OSCI channel if ACSLS and the SL4000 become disconnected due to network outages or other network related issues.

The OSCI retention time is set to two hours and is configured by ACSLS when creating the OSCI destination channel on the library. This is an internally configured time and is not configured by the installer or other ACSLS UI.

If ACSLS and SL4000 connectivity is lost for longer than the two hour retention time, OSCI events may be lost. In most cases, no further action is required once connectivity is restored. However, if CAP enter or eject operations were occurring at the time of connectivity loss and the OSCI retention time has elapsed, it may be necessary to stop and restart ACSLS to re-sync ACSLS with the library.

## ACSLs Support for Library Upgrade of SL3000 to SL4000

A field upgrade option is available to convert an SL3000 library into an SL4000 library. Although only ACSLS 8.5 can support the SL4000, either ACSLS 8.4 or ACSLS 8.5 may be managing the SL3000.

If ACSLS 8.4 (or a patch to 8.4) is currently managing the SL3000, use either of the following options:

- You can perform the library upgrade during the upgrade from ACSLS 8.4.0 to 8.5.0.
- You can perform the library upgrade after first importing the SL3000 into ACSLS 8.5.0

With either option, you remove the SL3000 library from the current configuration (by running `acsss_config`), and then add the SL4000 library to the ACSLS 8.5.0 configuration. After the library upgrade is complete, use either `acsss_config` or `config acs new` to add the SL4000 library. Once you run an audit against the SL4000, the library is ready for operation.

- If you plan to bring an upgraded SL4000 directly into ACSLS 8.5, remove the SL3000 library from the ACSLS 8.4 configuration before exporting the database for the upgrade to the 8.5 release. The library upgrade can be performed at the same time as the ACSLS upgrade.
- If ACSLS 8.5 is already managing the SL3000 prior to the library upgrade, shut down ACSLS and run `acsss_config` to drop the SL3000. The library upgrade can be performed at that time. No database export or import is needed.

## Important Considerations

- Drives and media supported by the SL4000 are not the same as those supported in the SL3000. Any affected drives or media should be physically removed from the SL3000 as part of the library upgrade.
- Once the library is ready and operational, perform all required library setup to create a user and password, certificates, etc. You will need this authentication information to configure the library to ACSLS.

- You must always run an ACSLS audit against a new library before starting client operations.

## What About my Volume Information?

By default, ACSLS retains volume records for 5 days after a volume is ejected or is marked absent (no longer in the library). This allows us to preserve information such as use count, last access date, or ownership.

Be sure to retain absent/ejected volumes for some non-zero period (`dv_config -p ABSENT_VOLUME_RETENTION_PERIOD`). A value of zero indicates to delete those records immediately.

After the SL3000 library is removed, the volumes are considered absent. When audit is run against the SL4000, the volume records are re-activated with new home locations.

## What About my Logical Libraries?

There is no provision for exporting a logical configuration, and the physical drive identifiers associated with any logical library backed by the SL3000 will change (panel numbers will be different). Therefore, the logical configuration must be re-defined after the SL4000 has been added to the ACSLS 8.5 physical configuration.

You can use the `lib_cmd display` command to create a record of the existing logical configuration (libraries, drives, volumes and mappings) for reference.

## What About my Client Applications?

Because the drive identifiers will change between the SL3000 and SL4000, along with CAP identifiers, client applications may need to refresh their view of the library configuration and then re-inventory the library. This is especially true for ACSAPI clients, which use ACSLS drive identifiers directly. However, even FC (logical) clients must refresh their views, since the element addresses for their resources may have changed. There should be no impact to XAPI clients, as library configuration is re-discovered at startup.

## Using watch\_vol

You can use the `watch_vol` utility, along with the `vol_attr.dat` file, to automatically assign various attributes when a volume is entered, discovered, or re-activated. This may be helpful in automating some of the media management tasks, including assignment of volume owner, logical library, etc.

# G

## Logical Library Support

This appendix discusses logical library support.

### About Logical Libraries

A portion of the ACSLS physical library configuration can be presented to SCSI clients as a logical library that can behave as a target device on the SAN. A logical library, in reality, is a defined parcel of a physical library with specific user-assigned resources including tape drives and volumes. Logical libraries are exposed to client application software through a SCSI (fibre-attached) interface.

A fibre channel HBA port typically operates in initiator mode, and can issue SCSI requests to a remote target device such as a disk drive, tape drive, or media changer device. In order to use logical libraries, you must configure one or more fibre ports to operate in target mode instead.

You can select one or more FC ports for target mode operation, or set target mode ports back to initiator mode, when running the `install.sh` or the `getHba.sh` utilities. A reboot is required for any change to take effect.

A logical library can be created using any eligible physical library as the backing ACS. The backing ACS identifies the physical library from which the logical library is created. The backing ACS:

- Must be configured to the ACSLS library server, but it need not be online in order to create a logical library.
- Can be an entire physical ACS, or it can be a physical partition of an ACS if the library itself supports physical partitioning.

To create, manage, or delete logical libraries, you can use either the ACSLS Graphical User Interface (GUI) or the `lib_cmd` Command Line Interface (CLI).

### Benefits

Logical libraries enable you to do the following:

- Partition physical libraries into logical libraries.

Logical libraries can be managed and used by client applications as if they were separate libraries.

ACSLs provides a flexible partitioning mechanism that is not tied to specific physical storage locations. A logical library is defined in terms of volumes and drives, and ACSLS presents these to the client using logical locations. The logical library is not limited by any physical divisions within an ACS, such as panel or rail boundaries.

Logical libraries also protect clients from potentially disruptive changes. When ACSLS chooses a new physical home cell for a volume, keeping it close to a compatible drive, or when an existing logical library is expanded, the client application is not impacted.

- Report the entire 8-character barcode as the `volser`.

When creating your logical library, you can specify the volume label format for clients: 6-character, 8-character-prefix, 8-character-suffix, or all. The 8-character barcodes include media domain, and type after or before the `volser`.

 **Note:**

Volume labels longer than 6 characters are not accessible to ACSAPI clients.

- Manage client access to drives and volumes.

Drives and volumes that are assigned to a logical library are accessible only to the client that is using the library, and are not visible to other FC or ACSAPI clients. However, the ACSLS GUI and `cmd_proc` are able to see them since these interfaces operate in system administrator capacity.

- Provide multiple initiator support.

A logical library can be made accessible to multiple clients (specifically, to multiple initiator ports). The purpose is to support redundancy on the client side, not to allow simultaneous operation of a logical library by multiple clients. Only one client system should operate a logical library at a given time.

Allowing multiple initiators means that a single client system can access their assigned logical library over multiple FC HBAs or ports. If the client environment supports a fail-over capability, the logical library is immediately accessible to the new active client system.

Any Unit Attention or Check Condition raised for a logical library is presented on each connection that has been configured.

## Limitations

Logical libraries:

- Are only accessible to clients using the ACSLS SCSI Media Changer Client Interface. They are not available to clients that use the legacy ACSAPI.
- Cannot span more than one physical ACS (or physical partition).
- Cannot reserve specific storage cells in the physical ACS. Combining logical and physical partitioning would be a way to accomplish a similar result.
- Target mode FC ports may no longer be included in the output from certain Solaris commands, such as `fcinfo`.

## Creating a Logical Library

Before you use the ACSLS GUI or `lib_cmd` to create a logical library, a physical library must have been created using the `acsss_config` or `config_acs` utility. An audit should also be performed before you create the logical library.

When creating a logical library, you:

- Specify a physical ACS
- Specify the attributes of the logical library

- Assign one or more physical drives
- Assign one or more physical volumes
- Specify client access to the logical library

## Specifying a Physical ACS

When you specify the backing ACS for a logical library, you automatically limit the drives and volumes that can be assigned to the logical library in subsequent steps. Only those drives and volumes that are in that ACS, and are not already assigned to another logical library, are available.

A logical ACS number is assigned by ACSLS when the logical library is created. The ACS number range for a logical library is n001-n999 where n = (1 + the backing physical ACS ID).

### For example:

- a logical library backed by physical ACS 0 might be 1001
- a logical library backed by ACS 4 might be 5001

## Specifying the Attributes of the Logical Library

The following attributes can be specified for a logical library:

- Unique name for the logical library
- Capacity

This is the maximum number of volumes that can be accessible in the logical library at any time (the number of storage elements that they would be able to inventory). The minimum is 0, and the maximum is 64,536. There is no default.

The capacity can be used both to restrict the size of a logical library (setting a limit that is less than the actual physical capacity) or to over-allocate physical capacity (setting a limit higher than what is actually available).

- Number of import/export cells

The minimum is 2, and the maximum is 400. The default value is 2.

- Number of drive slots

This is the maximum number of drives that can be assigned. The minimum is 0, and the maximum 500. There is no default.

- Label format

Today, support for longer volume labels in physical libraries relies on library firmware and configuration.

However, logical libraries can report the entire barcode (the 6 volume characters with the media domain and type suffixed). This provides an 8-character volume: `xxxxxxL4`. The logical library is configurable so the media domain and type can either be appended to the volume or they can prefix the volume (such as, `L4xxxxxx`).

The label formats supported, are: 6-character, 8-character-suffix (the default), and 8-character-prefix.

## Assigning One or More Physical Drives

Any physical drive that is installed and configured in the backing ACS can be added to a logical library. There are no restrictions on drive type for a logical library.

Physical drives that are allocated to logical libraries:

- become inaccessible to ACSAPI clients.  
The physical libraries, along with any drives and volumes that are not allocated to logical libraries, remain accessible to ACSAPI clients.
- are allocated exclusively.  
They cannot be shared across logical libraries.

You cannot assign more physical drives to a logical library than the number of drive slots that were specified when creating the logical library. However, it is possible to assign fewer drives since a logical library can have empty drive slots.



### Note:

Locked drives cannot be assigned to a logical library. (A locked drive is already being used by some ACSAPI client.)

You can increase or decrease the number of drive slots at any time. If you want to lower the number of slots to a value below the number of currently-assigned physical drives, you first have to un-assign some drives.

## Assigning One or More Physical Volumes

When creating a logical library, you specify its capacity: the maximum number of volumes that can be accessible at any given time. Make volumes accessible by assigning them to a logical library.

Assignment is exclusive - a cartridge can only be assigned to one library. Most cartridges that are present in the backing ACS, and that have not already been assigned to any logical library, are eligible for assignment.

You can not assign volumes that are:

- cleaning cartridges (ACSLs and the library handle cleaning).
- locked (A lock indicates use by an ACSAPI client).
- owned (An owner indicates use by an ACSAPI client).

Once the number of accessible volumes in a logical library has reached the maximum capacity, no more cartridges can be added. Either increase the library capacity, or free some used space.

Space is released in a logical library when an assigned volume becomes inaccessible. This can happen when an assigned volume is:

- marked for `eject` by FC client.
- ejected from the library.

- marked absent by ACSLS.

Volumes that are assigned, but inaccessible, are automatically reactivated if they become accessible again (for example, an ejected volume can be re-entered into the backing ACS) and if the logical library has space available.

Finally, you can free space in a logical library by un-assigning volumes. A cartridge can be unassigned from a logical library if the cartridge is not in use (such as, it cannot be mounted in a logical drive, it cannot be selected for an `eject` operation, and so forth).

Notes:

- Un-assigning a volume from a logical library does not imply any type of deletion or eject operation. Ejecting a volume does not imply an un-assign operation.
- Remember that physical storage cells are neither pre-allocated, nor assigned to a logical library. A logical library contains specific volumes rather than specific cells. Cells only "belong" to a logical library when they contain an assigned volume.
- The total number of accessible volumes among logical libraries that share a backing ACS can never be more than the physical capacity of the backing ACS.

The number of volumes assigned to a logical library can be displayed from `lib_cmd` or from the Logical Library Detail Information page in the GUI. Assigned volumes are shown as two separate counts:

- Accessible Volumes - This represents volumes that are assigned, are in the library, and are generally available for client use.
- Inaccessible Volumes - This number represents volumes that are assigned, but are not available for client use due to any of the following reasons: 1) the volumes have been marked as absent or ejected; 2) the volumes have been marked for eject by the FC client; 3) the volumes have been re-entered but the logical library was full; 4) the volumes have been re-entered into the wrong physical library.

You can increase or decrease the number of storage slots at any time. If you want to lower the number of slots to a value below the number of currently-assigned physical volumes, you first have to un-assign some volumes.

## Specifying Client Connections

The ACSLS SCSI Media Changer Interface makes logical libraries available for client use over a Fibre Channel (FC) connection. To allow FC clients to access the logical libraries, you will need at least one FC port that is configured to operate in target mode.

If you configured a target port while running `install.sh` during the ACSLS installation process, continue to create the logical library. If you have not yet configured a target port, use `getHba.sh` to configure the target port. Then, shutdown ACSLS and reboot the ACSLS server. For more information, see [How to Configure an FC Port in Target Mode](#).

Once you have configured a target port, the GUI makes it easy to manage client connections when you are creating or updating a logical library. The GUI screens provide a list of the target ports that are available on the ACSLS system, and also provide a list of any clients that have inquired about those ports over a FC connection. ACSLS automatically detects those inquiries and remembers the WWN of each client.

To assign a logical library to a client, check the appropriate boxes in the GUI.

Once you have configured a target port, it is easy to manage client connections either from `lib_cmd` or the GUI when you are creating or updating a logical library. Both the GUI and the



interactive-mode `lib_cmd` provide a list of the target ports that are available on the ACSLS system. They also provide a list of any clients that have inquired about those ports over a FC connection. ACSLS automatically detects those inquiries and remembers the WWN of each client.

To assign a logical library to a client, use `create mapping` in `lib_cmd` or check the appropriate boxes on the Manage Connections page in the GUI.

A friendly 'alias' name can be used in lieu of the WWN to identify each client system. To assign an alias, use `edit initiator` in `lib_cmd`, or select the 'Edit Initiator Alias' operation on the Manage Connections page of the GUI

## Fast Load

The ACSLS Fast Load setting controls how completion of the `move` command is reported when the destination element is a tape drive.

With Fast Load enabled, a success status is returned once the operation has been validated and accepted by ACSLS, but before cartridge movement begins. If some error should occur during movement, ACSLS does not report that information. The client is responsible for identifying when the volume has been loaded and is usable, and for timing out the request in the case of any error.

With Fast Load disabled (the default setting), success status is not returned until the physical library has reported that the movement is complete. However, if an error should occur during movement, ACSLS reports that information to the client.

Physical libraries may provide their own Fast Load option which can affect the time required before ACSLS would return a success status (but only when ACSLS Fast Load is disabled). When ACSLS Fast Load is enabled, the library setting would have no effect on client notification.

## Deleting a Logical Library

A logical library can be removed from the ACSLS configuration when it is no longer needed. Removing a logical library also removes all logical drives, volume assignments, and client mappings. The number assigned to this logical library can now be used when a new logical library is created. ACSLS does allow gaps in logical library numbering, but will reuse available numbers.

### Note:

Remove any logical libraries that are associated with a physical ACS before removing the ACS from your library configuration. This ensures proper removal of associated fibre channel connections.

Before you remove a logical library, note the following points:

- The logical library should be offline.
- All logical drives should be offline.

This ensures that no client operations are in progress when deleting the library. If a logical drive or volume is found to be in use, the logical library is not deleted.

## Troubleshooting

This section describes some possible problem scenarios, and offers ideas on how to diagnose and resolve the problems.

### What if a client cannot see the logical library that I've assigned?

1. Make sure that ACSLS is running, by running `acsss status`. This utility shows if ACSLS is in general enabled.  
  
For FC problems, look at the `smcse` service. This service provides the SCSI Media Changer interface to logical libraries.
2. If ACSLS is running, check to make sure your client connections have been defined correctly.
  - You may have assigned a target-initiator pair that is not really connected over FC
  - You may have assigned the logical library to the wrong client (the wrong target-initiator pair)
  - The `getHba.sh` utility can be useful in identifying which initiators (clients) are detected and which ones are connected to specific target ports.

### What if a client is connected correctly, but still doesn't see the logical library?

You may need to force the client to re-discover target devices. The actual mechanism depends on the operating system of the client system. The client OS must be able to see the device before any client application is able to use it.

For example, on a Solaris client, the `cfgadm` and `devfsadm` commands can be used to update the device file system and configuration.

### What if a client has trouble configuring to a logical library?

A client backup application may not recognize the ACSLS logical libraries as a supported library type. Logical libraries return a unique inquiry string which may require an update from the software vendor. An application-specific mapping file may also be required in order to configure the library and its tape drives correctly.

Sometimes, client software expects that a tape library always appears as LUN 0 on a SCSI target. Normally, the first logical library on a target port for a client would, in fact, be presented as LUN 0. However, if multiple libraries are assigned to a client on a given target port, they cannot count on always seeing a library as LUN 0. This is a client issue.

### How to Configure an FC Port in Target Mode

During installation, you are asked to select an HBA to be used as a target port. This operation applies only if you intend to present one or more logical libraries to SCSI (FC) client applications.

After installation, use the `getHba.sh` utility to configure a Fibre Channel port in target mode. The `getHBA.sh` utility can be run any time a new HBA has been added to the system, or any

time the HBA ports are re-arranged. This utility identifies an appropriate HBA port to change from initiator to target mode in order to reveal a client access point to ACSLS logical libraries.

If you intend to configure one or more logical libraries for SCSI direct, attach clients, then select the HBA that should be converted from initiator mode to target mode. Ideally, the HBA that you select is connected to the fabric, and has visibility to the remote HBA intended as the SCSI client initiator.

Once the target-mode change has been set for the given HBA, a reboot is necessary in order to complete the HBA conversion.

To configure a Fibre Channel port in target mode, complete the following procedure:

1. Login as `root`
2. Go to the `$ACS_HOME/install` directory, or include it in your path.
3. Run `getHba.sh`

Example:

```
# ./getHba.sh
Please select the HBE port you intend for Target-mode operation:
Select from the following list:
1)HBA Port WWN 2100001b3213ble2   Not connected.
2)HBA Port WWN 2101001b3233ble2   Not connected.
3)None of these.
2
HBA Port WWN 2101001b3233ble2/pci@1,0/pci1023,7450@1/pci1077,141@3,1
Is this correct? (y or n):y
Are there additional ports you wish to reconfigure? (y or n):n
A reboot will be necessary for these changes to take effect.
```

# H

## Library Partitioning

The SL8500 and SL3000 support partitioning of their libraries.

One ACSLS server can manage multiple partitions in the same library.

- SL8500 partitioning
  - SL8500 Complex Partitioning

The SL8500 8.3 and later firmware supports partitioning across a library complex. Up to 16 partitions can be created in a library complex of SL8500s connected by pass-thru ports.

ACSLs 8.3 and later releases support up to 16 partitions. ACSLS 8.2 and earlier releases support up to 8 partitions.
  - Enhanced SL8500 Partitioning (available with the SL8500 7.0x firmware)
    - \* With Enhanced SL8500 partitioning, only a single stand-alone SL8500 can be partitioned.
    - \* Enhanced SL8500 partitioning can support up to eight physical partitions.
    - \* Enhanced SL8500 partitioning is at the drive and cell array level.
  - Legacy SL8500 Partitioning
    - \* With Legacy SL8500 partitioning, only a single stand-alone SL8500 can be partitioned.
    - \* Legacy SL8500 partitioning can support up to four physical partitions.
    - \* Legacy SL8500 partitioning is at the rail (LSM) level, and a rail includes all licensed tape drives and cartridges on the rail.
    - \* A legacy SL8500 partition can include from one to four rails. The rails in a legacy SL8500 partition must be contiguous. For example, a partition can include rails 2, 3 and 4, but a partition cannot be just rails 2 and 4.

For more information on partitioning, refer to the SL8500 documentation set.

- SL3000 partitioning
  - The SL3000 can support up to eight physical partitions.
  - The SL3000 can partition down to the drive and cell level.

For more information, refer to the *SL3000 User's Guide*.

## Aspects Common to Both SL8500 and SL3000 Partitioning

Partitioning the SL8500 or SL3000 library provides the following capabilities:

- ACSLS can manage open systems drives and cartridges, and HSC can manage mainframe drives and cartridges sharing the library.
- More than one instance of ACSLS can access the library.

- One instance of ACSLS can manage multiple partitions in the same library.
- Provides protection and isolation of data cartridges in separate partitions.
- A higher level of data organization.
- An increase in user efficiency.

 **Caution:**

- You must `audit` a partitioned library after it is configured or reconfigured to ACSLS. The library reports the potential cell locations that are not accessible to ACSLS when ACSLS audits it. Inaccessible cell locations include: locations where CAPs, drives, and operator panels are installed; cell locations that the robots cannot access; cell locations that are not activated; and cell locations that are not in this partition.
- If cells are reassigned from one partition to another, the cartridges in those cells will be orphaned, and will no longer be accessible by the partition that they were in before. The host managing the other partition could write over the data on the cartridges.

 **Note:**

Although the SL500 can be partitioned, ACSLS does not support partitioned SL500s.

## Guidelines for Partitioning

There are several steps that you must follow when you are partitioning a library. They are:

1. Plan your new configuration.
2. Schedule an outage to partition or re-partition a library.

When a library is re-partitioned, the library drops its communication to all hosts managing all partitions, even if their partitions are not affected by the change.

 **Note:**

ACSAPI clients cannot be assigned to a physical partition of a library, just as they cannot be assigned to an ACS.

3. Have your service representative license and enable partitioning on the SL8500 or SL3000.
4. Move cartridges from locations where they would not be accessible to locations that will still be in a partition managed by ACSLS.
5. Create your partitions using the SL Console. Refer to the SL Console help for instructions.

6. Activate partitioning using ACSLS as described in [Partitioning a Library or Changing Partition IDs](#).
7. You must configure ACSLS to a partitioned library using `acsss_config` by specifying that the library is partitioned and entering the partition number.

## Plan Your New Configuration

Plan your new partitioned configuration ahead of time.

- Organize the cartridges and drives for performance.
- If you will do a lot of enters into and ejects from a partition, an SL8500 partition should include at least one of the bottom three rails. Partitions with few enters and ejects can be on the top rail.

## Minimize Disruptions when Partitioning an SL8500 or SL3000

When an SL8500 or SL3000 is partitioned or the partitions in an SL8500 or SL3000 are modified, access to your data is affected. Tape drives and cartridges that were managed by one host server may be managed by a different host.

## Changes in Configuration

Changes in configuration occur when a:

- Library configuration has changed
- Library has been re-partitioned and the partition managed by ACSLS has changed (a message is displayed)
- Library capacity has changed and the cells reported as available to ACSLS have changed
- When LSM Configuration or capacity have changed

When the configuration may have changed, you are notified by ACSLS Event Log and Event Notification messages.

## Keeping Outages to a Minimum

The following procedures provided below ensure that outages are minimized, with continued access to your data. These procedures explain how to:

- Partition an un-partitioned SL8500 or SL3000 into two or more partitions.
- Modify the partitions in an SL8500 or SL3000 that is already partitioned.
- Merge a partitioned SL8500 or SL3000 so it is a single, un-partitioned ACS.

## Challenges

The biggest challenges arise when: LSMs (SL8500 rails) are removed from a legacy SL8500 partition; or cells and drives are removed from an SL3000 or enhanced SL8500 partition.

Refer to the procedures for [Moving Cartridges Before Removing Cells from a Partition](#) for details.

## Partitioning a Library or Changing Partition IDs

These procedures describe how to partition an un-partitioned library or to change partition IDs on an existing partitioned library.

1. Move the cartridges before removing an SL8500 LSM or SL3000 or SL8500 cells from an ACS.  
If ACSLS manages one of the partitions in the new library configuration, follow the procedures for [Creating the Volume List File](#) or [Moving Cartridges Before Removing Cells from a Partition](#) to move the cartridges off of LSMs that will be assigned to another partition.
2. Quiesce the library before re-partitioning it, and vary the ACS(s) and port(s) offline.
3. Partition the library using the SL Console.
4. The following occurs after you press <Apply>. The library:
  - Rejects any new requests from ACSLS.
  - Drops all host connections.
  - Applies partitioning changes.
  - Accepts new host connections.
5. Reconfigure ACSLS to activate the partition changes.  
Because the library was partitioned for the first time, or it was re-partitioned and the partition ID changed, ACSLS cannot reconnect to the library. In order to reconnect, follow the procedure below:
  - a. Bring down ACSLS: `acsss disable`.
  - b. Run `acsss_config` to configure the ACS and select Option 8.  
Reply y (yes) when asked if the ACS is a partitioned library. After replying y (yes), enter the partition number. Refer to [Setting CSI Tuning Variables](#) for more information.
6. Start ACSLS with the new configuration: `acsss enable`
7. Vary the port(s) online and ACS(s) to diagnostic mode.
8. Audit the ACS(s).
9. Vary the ACS(s) online.

## Re-partitioning a Library

These procedures describe how to re-partition an existing partitioned library.

1. Move the cartridges before removing an LSM or cells from an ACS.  
If ACSLS manages one of the partitions in the new library configuration, follow the procedures for [Creating the Volume List File](#) or [Moving Cartridges Before Removing Cells from a Partition](#) to move the cartridges off of LSMs that will be assigned to another partition.
2. Quiesce the library before re-partitioning it, and vary the ACS(s) and port(s) offline.

This causes ACSLS to reject new requests from clients while the ACS(s) remain available for dynamic reconfiguration (after the re-partition).

3. Re-partition the library using the SL Console.

The following then occurs after you press **Apply**. The library:

- Rejects any new requests from ACSLS.
- Drops all host connections.
- Applies partitioning changes.
- Accepts new host connections.

4. Reconfigure ACSLS to activate the partition changes.

 **Note:**

- There may be no changes to a specific partition. The changes may only affect other partitions. In this case, no ACSLS configuration changes are needed.
- Because the library was not partitioned the first time or it was re-partitioned and the partition ID did not change, ACSLS can reconnect to the library.

Select one of the following:

- Run Dynamic Config (`config acs acs_id`) while ACSLS is running. This updates the ACSLS configuration to match the new library configuration. Follow this procedure:
    - Vary the port(s) online.
    - Update the ACS(s) configuration with `config acs acs_id`.
  - Or, bring down ACSLS and run `acsss_config` (option 8) to update the partition configuration in ACSLS. Follow this procedure:
    - Bring down ACSLS: `acsss disable`
    - Update the configuration using `acsss_config`.
    - Restart ACSLS: `acsss enable`.
    - Vary the port(s) online.
5. Vary the ACS(s) to diagnostic mode.
  6. Audit the ACS(s).
  7. Vary the ACS(s) online.

## Changing a Partitioned ACS to an Un-partitioned ACS

1. Optionally, `eject` cartridges from the partition(s) managed by ACSLS. If ACSLS will not be managing the non-partitioned library, you may want to eject cartridges from the partition(s) managed by ACSLS. This enables you to enter them into other ACS(s) that ACSLS manages.

You can use the `ejecting.sh` utility for ejecting multiple cartridges.



2. Quiesce the library before re-partitioning it, and vary the ACS(s) and port(s) offline.
3. Re-partition the library using the SL Console.  
The following then occurs after you press **Apply**. The library:
  - Rejects any new requests from ACSLS.
  - Drops all host connections.
  - Applies partitioning changes.
  - Accepts new host connections.
4. Reconfigure ACSLS to change the ACS to an un-partitioned library or to remove this ACS (partition) from its configuration.  
Because the library was changed from partitioned to un-partitioned, ACSLS cannot reconnect to the library. In order to reconnect, follow the procedure below:
  - a. Bring down ACSLS: `acsss disable`
  - b. Run `acsss_config` to configure the ACS.  
Reply n (no) when asked if the ACS is a partitioned library. See [Setting CSI Tuning Variables](#) for more information.
5. Start ACSLS with the new configuration: `acsss enable`.
6. Vary the port(s) and ACSs online.

## Viewing an ACS Partition ID

To view the partition ID(s) for one or more ACSs, use one of the following commands:

- `query lmu all` for all ACSs.
- `query lmu acs_id` for a single ACS.

Output Example:

```
ACSSA> q lmu 0
2008-02-27 06:08:02

ACS:   0      Mode: Single LMU      Active Status: Not Comm
Not Partitioned      Standby Status: -

      ACS State      Desired State
      offline       online

Port   Port State  Desired State  Role CL  Port Name0, 0      offline
online      -      13      172.27.2.6
```

## CAP Behavior

This section discusses CAP behavior.

## Partitioned Library

### **WARNING:**

All ACSs must have one CAP that is **dedicated** or is shared. Because all of the CAPs in an SL3000 can be dedicated to other partitions, this is a crucial restriction when configuring SL3000 partitions.

## Dedicating a CAP in a Partitioned Library

With an SL3000 library, a CAP can be dedicated to a partition. When CAPs are dedicated to a partition, ACSLS can set these CAPs to automatic mode.

## Sharing CAPs in a Partitioned Library

In an SL8500 library with legacy partitioning, CAPs are always shared among all partitions.

In a partitioned SL3000, or SL8500 with enhanced partitioning, library CAPs can be shared between partitions. The process is as follows:

1. When a host uses a CAP to enter or eject cartridges, it reserves the CAP for its exclusive use.
2. The host must complete the enter or eject and leave the CAP empty and closed.  
If you must terminate an enter on either of the following:
  - ACSLS - the canceled `enter` command must be issued
  - HSC - the `drain` command must be issued
3. The CAP is now available for use by any partition in the library.

## When Another Host Needs the CAP

If a host does not complete the enter or eject, the CAP may still contain cartridges for the affected partition. The CAP is still in a reserved status. This causes problems when other partitions need to use the CAP. To resolve this, the library must identify the host holding the reserve and terminate the enter or eject from that host.

In rare cases, you may not be able to access the host holding the reservation, either because the host crashed and cannot be restarted, or you lack physical access or the security authority to issue the commands needed. In these cases, the CAP reservation can be overridden using the SL Console.

### **Caution:**

Overriding a CAP reservation using the SL Console should be done only as a last resort.

## Assigning CAP Priorities for Shared CAPs

ACSLs can manage multiple partitions in the same library, and these partitions can share the same CAP(s). Audit ejects all cartridges with a duplicate and unreadable `vol_id`. When you audit multiple partitions (ACSSs) at the same time, the audits may try to use the same shared CAP to eject the cartridges. The first audit reserves the CAP and the other audits are unable to reserve the CAP. The later audits are then unable to eject the cartridges.

To prevent this from happening, set the CAP priorities so different shared CAPs are the highest priority CAP in the different partitions. For example, if partitions 2 and 3 in an SL3000 are sharing CAPs 5 and 6, set CAP 5 as the highest priority CAP in partition 2 and CAP 6 as the highest priority CAP in partition 3.

## Reserving the CAP

Consider the following when reserving the CAP:

- When a CAP is in use for an `enter` or `eject` operation, all 39 slots (SL8500) and 26 slots (SL3000) are reserved for that operation. The CAP cannot be subdivided.
- For addressing purposes, the CAP needs a location (ACS,LSM,CAP#). The LSM number is associated with the second rail in each library. For example: ACS#,1,0 for CAP A and ACS#,1,1 for CAP B
- Reserving the CAP

The CAP is a common component among partitions. Each host will:

- reserve the CAP to use it for an `enter` or `eject`.
- release the CAP when the `enter` or `eject` is terminated. Another host can then use it.

## Terminating a CAP Reservation

1. When ACSLS finds that a CAP is reserved by another partition, it reports the partition ID and host ID holding the reservation.  
Contact the host owning the reserved CAP and request for them to complete the `enter` or `eject`.  
Once this is done, the empty CAP is then unreserved and available.
2. If the CAP reservation cannot be released by the host holding it, the `SL Console` (SLC) command overrides the CAP reservation.
  - a. The SLC command displays a warning to the host informing it that the reservation from the specific partition owner will be removed.
  - b. The SL Console queries the CAP to find the partition owner/requester/reserver.
  - c. The SLC displays the partition name (HLI1, HLI2, HLI3, HLI4, or Default) in the warning.
3. The library sends a CAP Overridden Message to the host that held the reservation.  
This host should terminate the `enter` or `eject` task that was in progress.

4. The library now owns the CAP.  
The CAP must be empty and closed before it is available to be used by another host. If the:
  - CAP has cartridges in it, you must empty the CAP before a partition can use it using the SL Console.
  - CAP is open, you must empty and close it. The library can then `audit` it and ensure that it is empty. Until the CAP is closed, it cannot be made available to any partitions.
  - CAP is empty, it will be locked and available to be reserved by any partition.
5. The CAP is now unreserved and is available for use by any partition.

## Moving Cartridges to a Specific Cell

An SL3000 can partition down to the drive and cell level, and an SL8500 with enhanced partitioning can partition to the drive and cell array level. If cells are reassigned from one partition to another, the cartridges in these cells are orphaned, and are no longer accessible by the partition that they were in before. The host managing the other partition could then write over the data on these cartridges.

To prevent cartridges from being orphaned when partition boundaries change, and before you re-partition the library, move the cartridges to cells that will remain in the partition.

Since SL3000 is a single LSM, the existing ACSLS `move` command does not work. The cartridges would be moved somewhere else in the library, and may be moved to another cell that will also be removed from the partition. To handle this, the move command provides the ability to move cartridges to a specific cell.

The syntax for moving cartridges to a cell is like moving them to an LSM. However, instead of specifying an LSM ID, you specify a cell ID, as follows:

*a=acs, l=ism, p=panel, r=row, and c=column*

```
move AAAAAA a,l,p,r,c (for a cell move)
```

```
move AAAAAA a,l (for a standard LSM move)
```

Example for a cell move:

```
move EDU010 0,1,0,5,1
```



### Note:

You can use the `display` command to display a list of available (empty) cells in a specified panel: `display cell a,l,p,*,* -status empty -f status`

# Troubleshooting

This appendix summarizes tools, tips, and techniques for troubleshooting problems in ACSLS. The range of troubleshooting resources include logs, key observation points, and diagnostic probes.

## ACSLs Event Log

The ACSLS Event Log is the first stop for useful information if problems with your library operation occur. This log contains information about library events, status changes and errors. All sub-components within ACSLS will report events to the `acsss_event.log` by sending messages to a process called the Event Logger. The standard Event Log, which is automatically created when ACSLS is installed, is contained in the file `$ACS_HOME/log/acsss_event.log` and where `$ACS_HOME` is usually `/export/home/ACSSS/`.

Logged events include:

- **Significant Events**  
Significant Events are normal events that can help you manage the library. For example, events are logged when an audit is initiated or terminated, a device changes state, or a CAP is opened or closed.
- **Library Errors**  
Library Errors are events where both fatal and nonfatal hardware and software errors are logged. Examples include: LSM failures; problems with cartridges; database errors; process failures; and library communications failures.

Each message in the Event Log includes a time stamp, the name of the component reporting the message, and a description of the event. For a complete explanation of each message, consult the *ACSLs Messages* manual.

A window on the ACSLS console displays a running tail of the Event Log. You can generate a similar display from any shell window.

1. As user `acsss`, run the following command:

```
acs_tail $ACS_HOME/log/acsss_event.log
```

2. To view the entire Event Log, use a text editor, such as `vi`, that enables you to navigate through the log, search for specific errors, or follow specific sequences of events.

## Managing the Event log

ACSLs continues sending messages to the `acsss_event.log`.

- When this file reaches a threshold size (500KB by default), the file is renamed to `event0.log` and saved in the log directory. The `acsss_event.log` then continues as a new file.
- When `acsss_event.log` again reaches the threshold size, the `event0.log` is renamed to `event1.log` and the `acsss_event.log` is renamed `event0.log`.

- This process continues for as many log files that are configured for retention.

By default, nine Event Log files are retained in the log directory. With each subsequent threshold, the oldest file is removed and all remaining files are sequentially renamed.

You can configure the maximum size of the `acsss_event.log` and the number of log files to retain using `acsss_config`, Option 2. See [Setting CSI Tuning Variables](#).

## Using *greplog* to Search Event Logs

The diagnostic tool, `greplog`, enables you to perform keyword searches through any and all Event Log files. Used very much like the UNIX `grep` utility, `greplog` will return the complete log message associated with a given keyword expression. This enables you see the messages' date and time stamp, message number, and the function text related to every message containing that expression.

### Format

```
greplog [-iv] pattern file_1 file_2 ... feline
```

### Options

- `-i` instructs `greplog` to ignore the case of the search pattern expression.
- `-v` instructs `greplog` to filter out all messages containing the expression and to display all of the entries in the log file. The exception are those entries which match the pattern expression.

pattern: the pattern is the search criteria to be used.

```
file_1 file_2 ... file_n
```

The `greplog` tool accepts multiple file parameters and wildcard expressions in the file list.

### Examples

- To display all occurrences within an event sequence, use the sequence number.  

```
greplog 1392 acsss_event.log
```
- To search the Event Log for all messages about volume CART89:  

```
greplog CART89 acsss_event.log
```
- To search all archived copies of the Event Log for messages about tape mounts:  

```
greplog -i mount event*.log
```

## Additional Logs

The `acsss_event.log` contains all of the messages pertaining to any aspect of the ACSLS running processes. However, there are additional files in the log directory that contain status information about external utilities, such as backup and restore and installation utilities.

- `acsss.pid` - Stores the process id of the currently running `acsss_daemon`.
- `acsss_config.log` - Contains a summary of each library configuration.

- `acsss_config_event.log` - Contains event messages that were posted by the `acsss_config` routine.
- `bdb_event.log` - Contains event messages that were posted by the database backup utility, `bdb.acsss`.
- `cron_event.log` - Contains messages that were posted by `cron` utilities. To see the cron schedule, run the command `crontab -l`.
- `acsls_start.log` - Contains startup or shutdown messages involving the `acsls` service.
- `di_trace.log` - Contains trace information related to the database interface.
- `ejectingLogs` - A directory containing summary information from `ejecting.sh` operations for the past ten days.
- `install.log` - Contains event messages posted while running the installation script, `install.sh`.
- `ipc_trace.log` - Contains trace information pertaining to ACSLS inter-process communications.
- `rdb_event.log` - Contains event messages that were posted by the database restore utility, `rdb.acsss`.
- `timed_bkup.sh.log` - Contains event messages related to the automatic database backup utility.

Additional trace logs may be found in the log directory depending upon the specific tracing that you have enabled on your system. The logs include the following:

- `acsss_stats.log` - Volume statistics tracing is enabled by `acsss_config`.
- `acsss_trace.log` - Client-server tracing is enabled at the request of Software Support personnel.
- `acslh.log` - Host-LMU tracing is enabled at the request of Software Support personnel.
- `scsilh.log`, `mchangerX.log`, `scsipkt.log` - All of these contain traces of SCSI communications to a SCSI-attached library, and are enabled at the request of Software Support.

## Trace Log Management

Trace logs that are enabled at the request of Software Support can grow quite rapidly. These logs need to be monitored and managed in order to mitigate problems of a full disk.

The utility `monitor.sh` is provided to perform automatic log management and archiving services. The syntax, is:

```
monitor.sh <name of log>
```

When this utility is enabled to monitor a specific log, it will allow the log to grow to a size of 1MB (default) and then will compress the log using `gzip`, placing the compressed logfile with a time stamped name in the `ACSSS/log/log_archives` subdirectory. This operation will continue if tracing stays enabled.

## Java Component Logs

A number of logs are maintained by the Java components in ACSLS including ACSLS GUI and Logical Library software components. These logs are found in the `$ACS_HOME/log/sslm` directory.

WebLogic installation procedures are logged in the `weblogic.log`. WebLogic and ACSLS GUI operations are logged in the `AcslsDomain.log` and the `AdminServer.log`.

An audit trail of user activity in the Web-based GUI is found in the `guiAccess.log`.

Transactions between Java components and legacy ACSLS components are logged in the `surrogate_trace.0.log`.

IPC packets between Java client components and the ACSLS server are traced in the `acslm_ipc_trace.0.log`.

Errors encountered by the ACSLS GUI are logged in the `gui_trace.0.log`.

Low-level communication between the SMCE and the SCSI (fibre) client are logged in the `smce_trace.0.log`.

These logs are found in the `$ACS_HOME/log/sslm` directory.

## Key Observation Points

There are numerous utilities that enable you to verify the status of various aspects of ACSLS.

- `psacs` - shows a summary of all of the ACSLS running processes. It is the best indication whether ACSLS is running or not. A typical output should display no fewer than twelve different processes, all children of a common parent process.
- `acsss status` - checks if the `acsdb` database service is running
- To display your ACSLS release and maintenance level:
  - On Solaris:

```
pkginfo -l STKacsls
```
  - On Linux:

```
rpm -q ACSLS
```
  - On Solaris or Linux:

```
in_get_version
```

## Diagnosing ACSLS Startup Problems

- Look at the `acsss_event.log`.
- Look at the `acsls_start.log`.
- Look at the end of the `acsss_event.log` for messages explaining the problem.
- Refer to the ACSLS Messages Guide for an explanation of the messages and what you can do to resolve them.



- Display the status of ACSLS services with `acsss l-status`.  
Use `acsss l-status` to show a status summary of the ACSLS services. For each service, the `logfile` entry points to log data which may contain detailed messages explaining the condition that prevented ACSLS from starting up.
- ACSLS time outs during startup
- On Solaris, to display the calculated ACSLS start-up timeout period based on your configuration use `acsss timeout`.

## Testing Library Connections

ACSLs provides utilities for verifying a good physical connection to the library. The tool you select is best determined by the context of your activity.

### testports

This utility tests the connection to each library that has been configured to StorageTek ACSLS. It is also the easiest to use and is the most comprehensive. The test is un-obtrusive and does not impact normal library operations. Since `testports` uses the StorageTek ACSLS database to determine the library port name and library type, the library must already have been configured to StorageTek ACSLS in order for `testports` to function.

- For TCP/IP libraries, `testports` verifies the connection and whether the library is online and in use by StorageTek ACSLS.
- For SCSI and serial-attached libraries, the 'acs' and 'port' must be offline in order for `testports` to open the test connection.

To run this utility, the command syntax, is:

```
testports
```

The library compat level or microcode level displays.

### testlmutcp

This utility submits a TCP/IP packet to a network-attached library.

To test the library connection, include the library hostname or ip address in the command line:

```
testlmutcp <ip_address>
```

or:

```
testlmutcp <hostname>
```

To test the connection while the library is online to ACSLS, specify an unused socket number between 50002 and 50016. For example:

```
testlmutcp <ip_address>:50002
```

A successful response will include the compatibility level of the attached library.

## testlmu

This utility can be used to test connectivity between ACSLS and legacy StorageTek serial-attached libraries. To run this utility, submit the devlink path to the serial port device node:

```
testlmu /dev/term/0
```

The library must be offline to ACSLS in order for `testlmu` to open the serial connection.

## pinglmu.sh

This utility enables you to verify communication between ACSLS and a serial-attached library while the library is online to ACSLS. A successful response includes the library compatibility level.

## probescsi.sh

This utility exercises the connection between the ACSLS server and a SCSI or fibre-attached library. To run this utility, specify the devlink path to the mchanger device. The syntax, is:

```
probescsi.sh /dev/mchangerX
```

where *X* is the specific mchanger instance of the library being tested.

The library must be offline to ACSLS in order for `probescsi` to open the SCSI connection. A successful response includes the microcode level of the attached library.

## probeFibre.sh

This utility discovers all fibre-attached libraries that are reachable from the ACSLS server. The syntax, is:

```
probeFibre.sh
```

A successful response displays the model number of each fibre-attached library along with its target, LUN IDs, and the World Wide Port Name (WWPN).

Using the `-v` option, you can also display the model number of the host bus adapter.

```
probeFibre.sh -v
```

## showDevs.sh

This utility reveals details about every mchanger device for which a mchanger link has been created.

- `showDevs.sh`

Displays library model, revision level and the capacity of each attached mchanger library.

- `showDevs.sh -w`

This option also includes the WWPN of each library.

- `showDevs.sh -s`

This option also includes the serial number of each library.

## Troubleshooting an Offline SL4000 ACS

When network disruptions occur, the SL4000 library may disable the OSCI channel for ACSLS, placing an ACS offline. Use the following procedure to re-enable the ACS:

1. Verify that the SL4000 library is online and in an operational (or degraded) state.
2. Verify that the SL4000 OSCI connection to ACSLS is enabled. In the SL4000 BUI, select **Notifications** and then **SCI screen**. Here you can enable the OSCI connection. Refer to the *SL4000 User's Guide*.
3. Use the `testports` utility to verify the network connection. See [testports](#).
4. Verify that `port`, `lsm`, and `acs` are online. Enter the following commands in order, replacing `acs_num` with the `acs` number:

```
vary port acs_num,0 online
vary lsm acs_num,0 online
vary acs acs_num online
```

## Testing a Client Connection

Client applications communicate with ACSLS over TCP/IP using the RPC (remote procedure call) protocol. If a client system cannot communicate with ACSLS, you can use `rpcinfo` to test whether ACSLS is reachable from the client machine.

1. From the ACSLS server, verify that ACSLS is running.

```
psacs
```

2. From the ACSLS server, verify that the RPC daemon is running.

```
ps -ef | grep rpc
```

3. From the ACSLS server, verify that program number 300031 is registered for TCP and IDP.

```
rpcinfo | grep 300031
```

This program number confirms that ACSLS is running and that ACSLS has registered with RPC.

4. From the client machine, or any UNIX machine on the network, use `rpcinfo` to exchange a packet with program number 300031 on the ACSLS server.

Specify the IP address of the ACSLS server along with the program number.

```
rpcinfo -t <ip address> 300031
```

If the communication exchange was successful, the `rpcinfo` utility will display the following message:

```
program 300031 version 1 ready and waiting
program 300031 version 2 ready and waiting
```

This confirms that ACSLS is available for client connections across the network.

## CAP in Fibre Library Attached Through a Bridged Drive is Locked

A CAP in a Fibre-attached library that is connected through a bridged drive may become locked when a different ACSLS instance takes over management of the library. For details about and solutions to this problem, see [CAP \(Mailslot\) Not Opening During an Eject](#) in [ACSLS Support of the SL150](#).

## Gathering Diagnostic Information for Oracle Support

As part of the service call, Oracle Support may ask you to send the entire set of diagnostic logs and other diagnostic information for analysis. All of this data can be collected with a single command:

```
get_diags
```

When this utility has collected all of the information, it prompts you to either email the data or to make it available for manual transfer.

If you elect to e-mail the data directly from the ACSLS machine, make sure that email communication is possible between your ACSLS machine and the Internet. Your enterprise may have a firewall to prevent e-mail going directly from the target machine. In this case, you can e-mail the information to yourself within the enterprise and then forward the diagnostic data to Oracle Support.

Alternatively, you can elect to transfer the information manually. The `get_diags` utility advises you where to find the waiting tar packages for transfer. Typically, the staging area for diagnostic data is `/export/backup/diag/acsss`.

## ACSLS and Security-Enhanced Linux (SE Linux)

SE Linux is enabled by default in Oracle Linux. Beyond standard Unix level access control, SE Linux enforces access to system resources according to user role and the immediate context domain. When SE Linux enforcement is enabled, the ability of ACSLS to access its own PostgreSQL database would be inhibited without a special policy that establishes the role and context for such access. You may need to disable SE Linux, or to run in permissive mode, if problems occur.

## Un-installing SE Linux Policy Modules for ACSLS

Three SE Linux policy modules are loaded into the kernel when you install ACSLS: `allowPostgr`, `acsdb`, and `acsdb1`. These modules provide the definitions and enforcement exceptions that are necessary in order for ACSLS to access its own database and other system resources while SE Linux enforcement is active. With these modules installed, you should be able to run normal ACSLS operations, including database operations such as `bdb.acsss`, `rdb.acsss`, `db_export.sh` and `db_import.sh` without the need to disable SE Linux enforcement.

For purposes of quicker software upgrades, the SE Linux policy modules that were loaded by ACSLS are not removed automatically when you uninstall the ACSLS package. To remove them manually, get a listing of the ACSLS modules:

```
# semodule -l | grep acsdb
# semodule -l | grep allowPostgr
```

For each module, remove the module in the following manner:

```
# semodule -r <module name>
```

## Managing SE Linux Enforcement

After installing ACSLS, if you encounter access related issues where the system responds with 'permission denied' while traditional file permission settings appear to be valid, the source of the access denial may be SE Linux.

To verify whether SE Linux enforcement is enabled, run the command: `sestatus`

```
# sestatus
SELinux status:   enabled
Current mode:     enforcing
```

You can disable SE Linux enforcement temporarily using the command: `setenforce:`

```
# setenforce Permissive
```

With SE Linux enforcement in the permissive mode, you can now check whether access to the failed resource can be restored. If the necessary resource is available to the authorized user in the permissive mode but not in the enforcing mode, this suggests the need for an updated SE Linux policy

To disable SE Linux security permanently (across boots):

1. Edit the file: `/etc/selinux/config`
2. Change: `SELINUX=enforcing` to `SELINUX=permissive`

To re-enable SE Linux enforcement, `root` needs to have the `sysadm_r` role.

```
# newrole -r sysadm_r
# setenforce enforcing
```

After you have verified that SE Linux was the cause of the apparent restriction, you can view the actual rules that disallowed access to the needed resource by looking at the SE Linux audit log.

```
# vi /var/log/audit/audit.log
```

The `audit.log` provides a summary for each access attempt that succeeded or failed SE enforcement. You should look for the events that failed. For ACSLS, look specifically at events related to the users `acsss` and `acsdb`.

You can view the SE Linux context attributes associated with any given file or directory:

```
# ls -Z <file name>
```

You can view the context attributes of a given process or those of your current shell using the command: `secon`. It is possible to change the context attributes of a file or directory using the command `chcon`. Consult the man pages for these operations.

It is possible to create a policy module in response to the failed operations found in the `audit.log`.

```
# cd /var/log/audit
# audit2allow -a -M <ModuleName>
```

This evaluates the failures that were logged by SE Linux and it creates a policy module file `<ModuleName>.pp`. This file can now be loaded into the Linux kernel in order to allow the operations that had been blocked.

```
# semodule -i <ModuleName>.pp
```

Since `audit2allow` creates a policy that enables all of the restrictions identified in the `audit.log`, it is wise to make sure that the `audit.log` contains only those operations that you specifically want to allow. You can save the original `audit.log` and create a new one.

```
# mv audit.log audit1.log
# touch audit.log
```

Proceed with the operations that you want to capture before you create a policy module for them.

For more information regarding SE Linux, consult the man page:

```
# man selinux
```

## Verifying the GUI is Operational

The `checkGui.sh` utility checks common factors to assess whether the GUI is operational. In the event that the GUI is not working, this utility may lead users to the likely cause of the problem.

This utility checks the following:

- Is weblogic enabled on the system?
- Are there phantom or stale processes that may inhibit weblogic operation?
- Is the SlimGUI application successfully deployed?
- Can weblogic and the GUI respond to an http request sent to localhost?
- Can weblogic respond to an http request sent to the internet address of the host?
- Is the firewall service enabled on the server? If so, is there a policy to accept incoming requests to weblogic ports 7001 and 7002?

On Linux systems, you may find that the firewall called `iptables` is enabled by default. You can disable `iptables` completely, or you can add a policy to accept incoming traffic to ports 7001 and 7002.

1. To enable these ports (as `root`), edit the file `/etc/sysconfig/iptables`. Add the following two lines:

```
-A INPUT -p tcp --dport 7001 -j ACCEPT
-A INPUT -p tcp --dport 7002 -j ACCEPT
```

Make sure that you do not insert these rules after another rule that will match incoming packets before they are examined. For example, do not append them to the end of an `iptables` chain after a rule to `REJECT all`.

If you are using the `iptables` command to add these rules:

- List (`iptables -L`) or Print (`iptables -S`) the table.

- Add the rules.  
Just Appending (`iptables -A`) the rules to the end of a chain may not produce your desired outcome because prior rules may prevent the new rules from matching any input.

Perhaps Insert (`iptables -I`) the rule by `rulenum`.

- List (`iptables -L`) or Print (`iptables -S`) the table after the change and make sure that existing rules do not prevent the new rules for ports 7001 and 7002 from ever being examined.

This ensures that the new rules can match an incoming packet.

The `checkGui.sh` utility checks for the existence of rules to ACCEPT input on ports 7001 and 7002. It does not verify that these rules are in the right iptables chain or that the new rules will actually be processed. In other words `checkGui.sh` does not verify that there are no prior rules which would prevent the examination of the new rules.

2. Restart iptables:

```
service iptables restart
```

The comparable service on Solaris is `ipfilter`, which is not normally enabled by default.

## GUI Troubleshooting Tips

The following table discusses some GUI troubleshooting tips.

**Table I-1 GUI Troubleshooting Tips**

| Problem                                                                                                                | Solution                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| I have placed <code>https://&lt;hostname&gt;</code> in the browser but the response page declares "Unable to connect". | The correct URL is: <code>https://hostname.domain:7002/SlimGUI/faces/Slim.jsp</code>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| The ACSLS GUI page is incomplete. Some frames are incomplete or entire sections missing.                               | Click the refresh button on your browser.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| Java WebLogic rejects what I know to be a valid user i.d. and password. I can't log in                                 | Consult with your local ACSLS administrator. If you are the administrator, use the <code>userAdmin.sh</code> utility to list users, add a user, or change a user password.<br><br>If the users still have trouble logging in, check your system for sufficient memory, and then restart the ACSLS GUI with option-5 of <code>userAdmin.sh</code> . Alternatively, you can restart WebLogic using <code>svcadm disable weblogic</code> and <code>svcadm enable weblogic</code> .                                                                                                                                                                                                           |
| A java error stack trace is displayed in one or more windows of the GUI.                                               | Press the refresh button on your browser.<br><br>If the problem persists, use <code>acsss status</code> to verify that ACSLS services are running.<br><br>If services are not running, bring them up with <code>acsss enable</code> .<br><br>If ACSLS services are running, restart the GUI using <code>userAdmin.sh</code> . Alternatively, you can restart WebLogic with <code>svcadm disable weblogic</code> and <code>svcadm enable weblogic</code> .<br><br>If you do not have <code>root</code> access to the system, you can shutdown all services with <code>acsss shutdown</code> , then, restart them up with <code>acsss enable</code> . The GUI is restarted by this process. |

**Table I-1 (Cont.) GUI Troubleshooting Tips**

| Problem                                                                                               | Solution                                                                                                                                                          |
|-------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| The selection for "Logical Libraries" is missing in the index tree frame.                             | You must first create a logical library. Select: Configuration and Administration ->Logical Library Configuration ->Create Logical Library.                       |
| No volumes are listed on the Volumes page under Tape Library Operations or Tape Libraries & Drives.   | This is an indication that an initial audit has not been taken for the library. Select: Tape Library Operations ->Audit.                                          |
| No volumes are listed on the Volumes page under Logical Libraries.                                    | This is a sign that volumes have not yet been assigned to the logical library. Select: Logical Library Configuration ->Assign Volumes.                            |
| GUI response time is slow                                                                             | Increase the Alert Update Interval under the Preferences button in the GUI masthead.                                                                              |
| I need to add GUI users, change the passwords of GUI users, or set <code>acsls_admin</code> password. | See <a href="#">userAdmin.sh</a> .<br>This utility lets you add users, change user's passwords, and tells you how to reset the <code>acsls_admin</code> password. |
| Browser requires a security certificate.                                                              | See <a href="#">Configuring a Self-Assigned Digital Certificate for HTTPS</a>                                                                                     |

## Linux 7.3 and 7.6 Performance Issues

Under Linux 7.3 and 7.6, significant ACSLS performance slowdown conditions may occur. These conditions are traced to missing host entries and the use of the SSSD service.

Slow resolution of UIDs, groups, automounts, permissions, and related details can have wide-ranging impacts. Many applications, including ACSLS, use the `su` command, which requires resolution of system user details prior to proceeding.

### Missing Host Entry

Make sure that you do not have a missing host entry in your `/etc/hosts` file. Include both the fully qualified host name and shortened version. For example:

```
[root@mycomputer ~]# cat /etc/hosts
10.0.0.0      mycomputer.us.domain-name.com    mycomputer
```

### SSSD

Use of the Linux System Security Services Daemon (SSSD) service can impact ACSLS performance, causing numerous types of slowdown conditions. These conditions revolve around refreshing of a cache and re-walking trees of user names, groups, permissions, and so on.

If you encounter a slowdown condition, use the following guidelines to determine whether the condition is related to your use of SSSD:

#### Diagnosis 1

1. Verify that the machine is responding slowly to any `su` command.



2. Check the `/etc/nsswitch.conf` file for occurrences of "sssd". If you do not find occurrences of "sssd" then proceed with [Diagnosis 2](#).
3. For diagnostic purposes, remove any occurrences of "sssd". No restart required.
4. Test the `su` command again several times, to varying users. If response time is improved, then the SSSD service is the root cause of your slowdown condition.

## Diagnosis 2

If there are no occurrences of "sssd" in the `/etc/nsswitch.conf` file, SSSD can still cause slow performance.

1. As `root`, enter the following command to display the status of the SSSD service:  

```
systemctl status sssd.service
```
2. If the service is active, enter the following command to turn it off:  

```
systemctl stop sssd.service
```
3. Enter the following command to disable the SSSD service:  

```
systemctl disable sssd.service
```
4. Test the `su` command again several times, to varying users. If response time is improved, then the SSSD service is the root cause of your slowdown condition.

## SL4000 OCSI Retention Time

The OSCI retention time determines how long the SL4000 library retains events on the OSCI channel if ACSLS and the SL4000 become disconnected due to network outages or other network related issues.

The OSCI retention time is set to two hours and is configured by ACSLS when creating the OSCI destination channel on the library. This is an internally configured time and is not configured by the installer or other ACSLS UI.

If ACSLS and SL4000 connectivity is lost for longer than the two hour retention time, OSCI events may be lost. In most cases, no further action is required once connectivity is restored. However, if CAP enter or eject operations were occurring at the time of connectivity loss and the OSCI retention time has elapsed, it may be necessary to stop and restart ACSLS to re-sync ACSLS with the library.

# J

## Issues Installing ACSLS Client Applications

This appendix summarizes known issues when installing client applications that communicate with ACSLS and their successful resolution.

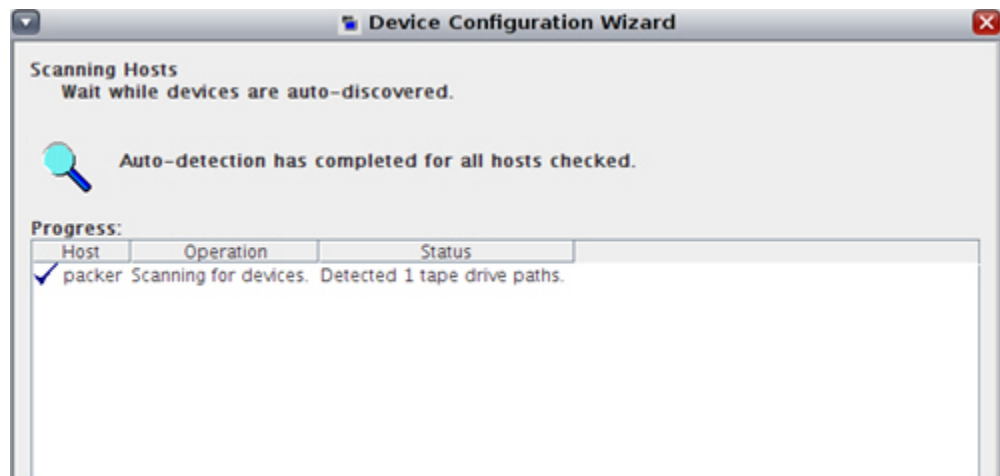
### Installing an ACSAPI Client on Solaris 11

Solaris 11 does not include all of the packages needed for client applications to communicate with the ACSLS server using RPC.

Problem:

In this example, the client was unable to detect the ACSLS library.

**Figure J-1 Device Configuration Wizard - Drive Path Detection**



Resolution:

1. Modify the `/etc/hosts` file and reboot the ACSAPI client server.  
Default on Solaris 11: (has just a loopback IP address)

```
:::1 <client_hostname> localhost
127.0.0.1 <client_hostname> localhost loghost
```

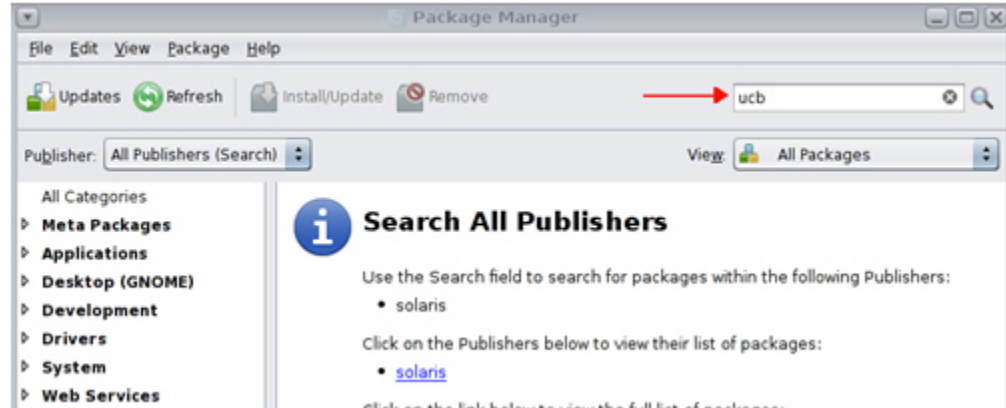
Updated `/etc/hosts` file on Solaris 11: (includes an actual IP address for the ACSAPI client)

```
:::1 localhost
127.0.0.1 localhost
<client_IP_addr> <client_hostname> loghost
```

2. Install `compatibility/ucb` package required for ACSLS communication using Package Manager.

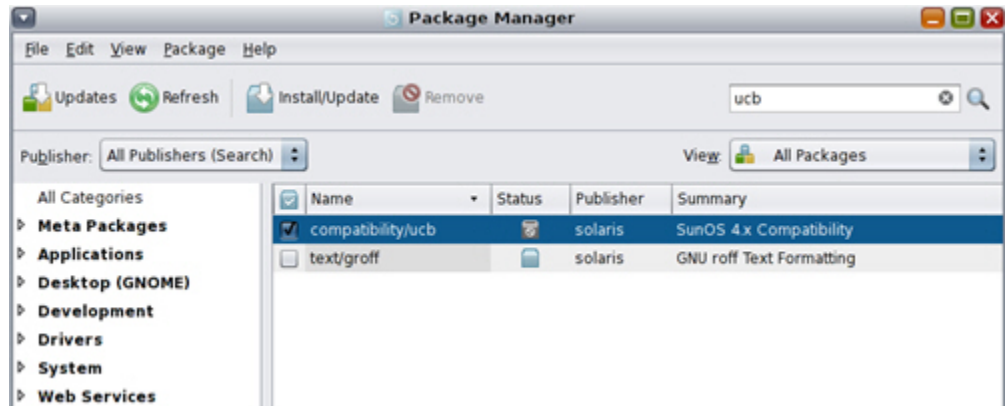
- a. Start Package Manager.
- b. Search for 'ucb'.

**Figure J-2 Package Manager Search**



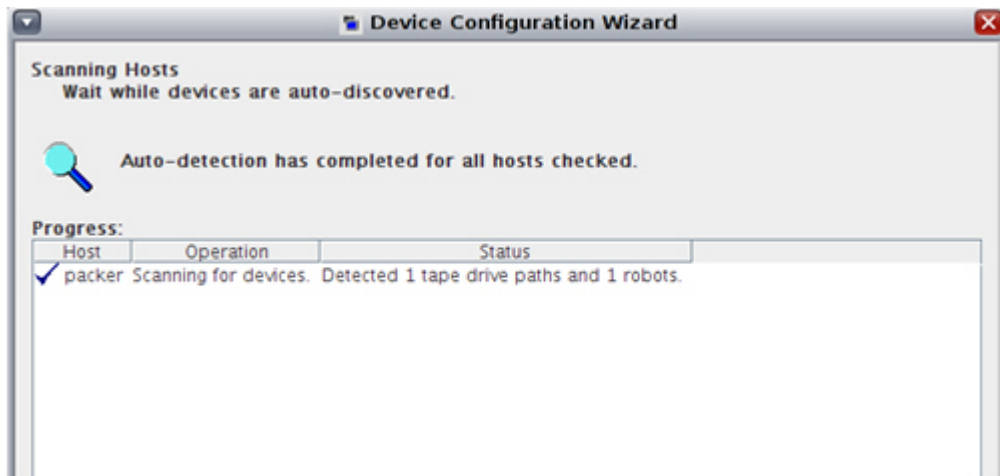
- c. Check the compatibility/ucb package and click Install/Update.

**Figure J-3 Package Manager Search Results**



- d. Click on the General tab to make sure that the latest version is installed.
  - e. Run Install/Update again, if a newer version is available.
3. Reconfigure the client.

Figure J-4 Device Configuration Wizard - Drive Path and Robot Detection



# K

## Library Performance

This chapter discusses how to maximize your library performance during pass-thru activity between Library Storage Modules (LSMs). ACSLS works to minimize pass-thru activity in several ways. You can enhance performance by using these ACSLS facilities to minimize pass-thru activity. They are:

- Sending enough concurrent mount and dismount requests
- Connecting to Multiple SL8500s
- Minimizing pass-thru activity between LSMs
- Floating cartridges during a dismount
- Entering cartridges into a library
- Ejecting cartridges from a library
- Maintaining empty storage cells in a library
- Minimizing pass-thru activity with ACSAPI requests and ACSLS commands
- Using these ACSAPI requests and ACSLS commands

### Sending Enough Concurrent Mount and Dismount Requests

Some client applications are designed for simple SCSI Media Changer libraries, which only process a single request at a time. However, ACSLS and SL8500 and SL3000 libraries, are designed to process many requests at the same time. Each library can have at least 40-50 concurrent requests in its queue, and the ACSLS queue is essentially unlimited.

A client can send ACSLS hundreds of concurrent mount and dismount requests. To maximize library performance, the client(s) should send ACSLS enough requests to keep all of the robots, and so forth in the library, busy.

If a client can only send ACSLS a limited number of concurrent requests, you can attach multiple clients to ACSLS. For example, if a particular client application only sends ACSLS a single mount or dismount request at a time, by connecting 12-16 client applications per SL8500 to ACSLS, you can send ACSLS enough concurrent mount and dismount requests to keep all of the robots in an SL8500 busy.

Here are the details to understand how many concurrent requests may be needed to keep of the library's robots busy:

- The library does not return the mount response to ACSLS until the drive is loaded and ready. In this way, ACSLS knows whether the mount was successful.
- In a smaller SL8500, with about 3000 storage cells, it takes the robot about 10-15 seconds to move a cartridge to a drive.
- However, it takes an LTO drive about 19 seconds to load the cartridge and come ready.
- To keep even one robot on a rail busy, you need at least three overlapping mount requests on that rail.

This lets the robot(s) process other mounts while the first drive receiving a cartridge is loading and coming ready. With two robots per rail, you may want to have at least four concurrent mounts or dismounts per rail.

- An SL8500 has four rails, so you need 12-16 concurrent mount and dismount requests per SL8500 to keep all of the robots in an SL8500 busy.

All of these concurrent requests could come from a single ACSAPI client, or they could come from 12-16 different clients, each of which only issues a single request at a time, and waits for the response before issuing the next request.

## Connecting to Multiple SL8500s

If your ACSLS server is managing a string of SL8500s connected through pass-thru ports, connect to each SL8500 in the string. This lets ACSLS optimize library performance by routing requests to the library that answers the status or initiates the cartridge movement.

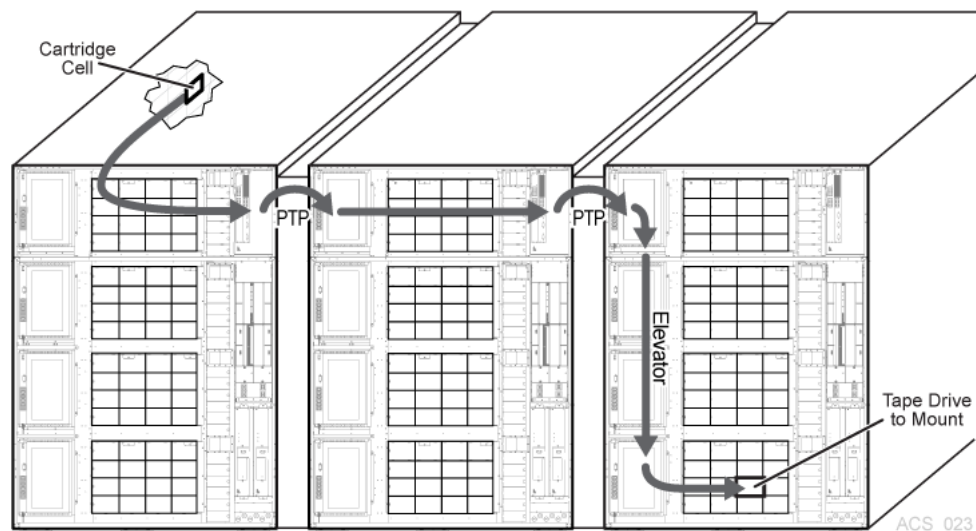
Refer to [Multi TCP/IP Support](#) for details on connecting to multiple SL8500s in a string.

## Minimizing Pass-thru Activity Between LSMs

To improve tape library performance, it is important to minimize pass-thru activity from one library to another. When cartridges and drives are in connected libraries, it is always desirable if tape mounts occur when both the drive and the cartridge are in the same LSM (rail within a library). Mounts that require fewer pass-thrus result in better performance.

The following figure shows the pass-thrus that can be required when a cartridge and drive are in different LSMs.

**Figure K-1 Minimizing Pass-thru Activity Between LSMs**



For more information, see:

- [Minimizing Elevator and PTP Activity](#)
- [Configuring Tape Drives to Support Your Workloads](#)
- [Managing Cartridge Locations](#)

## Floating Cartridges During a Dismount

When cartridges are dismantled, ACSLS tries to avoid pass-thru activity among LSMs by assigning a new home cell whenever the cartridges' old home cell is in a different LSM than the drive. This is ACSLSs' default behavior. To disable this feature, use the "Extended Store" feature for an LSM. Cartridges being dismantled are "floated" to a new home cell to avoid movement between LSMs.

## Maintaining Empty Storage Cells in an LSM

ACSLs cannot "float" cartridges to an LSM on a dismant if all of the storage cells are full. Similarly, when cartridges are entered into a full LSM, the cartridges must be passed-thru to an LSM with empty cells.

To optimize performance, identify cartridges that do not need to be kept in the library for StreamLine access, and `eject` those cartridges from the library. You may also want to move cartridges from full LSMs to other LSMs with enough empty cells so that all LSMs have empty storage cells. See [Moving Least-Recently Accessed Cartridges from Active LSMs](#) .



### Note:

Managing the cartridges in a library should be done when the library is not busy with mount requests.

## Mount and Dismount Timeouts

The timeouts for mounts and dismants need to be coordinated between ACSLS and client applications. The clients' timeouts need to be long enough that mounts and dismants that require pass-thru between SL8500 rails and libraries can finish. They also need to be long enough so multiple concurrent requests can finish, especially if the requests may wait for a pass-thru port.

ACSLs also automatically queues requests during a temporary library or tape drive outage and retries them when the library or drive is available. This prevents requests from failing during a Redundant Electronics switch, when a library or drive is re-booted, or when the library's access door is opened.

The default time to queue mounts and dismants (`MOUNT_RETRY_TIME_LIMIT`) is 20 minutes. Customers can set this to any time between 5 and 80 minutes. ACSLS also checks every `MOUNT_RETRY_DELAY` to see if the library is available to process the queued request. Both of these are dynamic variables, which means that they can be changed while ACSLS is running, and the change takes effect immediately.

Adjust the ACSAPI clients' timeouts to allow requests in a large library complex to complete. Also coordinate the `MOUNT_RETRY_TIME_LIMIT` with the ACSAPI clients' mount and dismant timeouts.

## Entering Cartridges into a Library

When cartridges are entered into a library, each cartridge is assigned a home cell in the closest LSM to the CAP. If the LSM with the CAP has empty cells, home cells are assigned in this LSM. If the LSM with the CAP is full, home cells are assigned in the closest LSM with empty cells.

To optimize performance, `enter` cartridges into an LSM with tape drives that are compatible with the cartridge(s). For example, `enter` 9940 cartridges into an LSM with T9940B tape drives, not into an LSM with only T9840B tape drives. Otherwise, the 9940 cartridges must be passed-thru to an LSM with T9940B tape drives.

## Ejecting Cartridges from a Library

When cartridges are ejected from a library, the cartridges must be moved to the specified CAP(s).

To optimize performance, `eject` cartridges to a CAP in the LSM where the cartridges are located. If the cartridges are located in several LSMs, choose a CAP that is close to most of the cartridges.

## Minimizing Pass-Thru Activity Using ACSAPI Requests and ACSLS Commands

These ACSAPI requests and ACSLS `cmd_proc` commands can help you avoid unnecessary pass-thru activity between LSMs.

The following applies to both ACSAPI requests from a client to ACSLS, and to ACSLS commands entered using `cmd_proc`. The term "request" is used for both ACSAPI requests and `cmd_proc` commands.

## Mounting a Specific Cartridge

This applies to either mounting a data cartridge or a specific scratch (empty) cartridge that you selected. Use the following commands:

- `query mount`
- `mount`

### `query mount`

- For a specified cartridge (*vol\_id*), return a list of drives that are compatible with the cartridge's media type, ordered by proximity to the cartridge. (The compatible drives in the closest LSM are listed first, and so forth.)
- With ACSLS 7.3 and later releases, drives that are the same pass-thru distance from the cartridge are in least-recently-used order.

**Example:** The compatible drive within the cartridge's LSM with the longest time since a cartridge was dismounted, is first; the drive with the next longest time, is second, and so on.



**Command:** `query mount vol_id`

## mount

Mount your specific cartridge, selecting an available (not in-use) drive from as close to the top of the list as possible.

## Mounting a Scratch Cartridge Selected by ACSLS

The following section discusses mounting a scratch cartridge selected by ACSLS.

### query mount scratch (also known as query mount \*)

For a specified scratch pool, a list of all drives compatible with the media type(s) of the cartridges in a specified scratch pool is returned. A specific media type can be specified to restrict the drives to only those compatible with the specified media type.

The list of drives returned is ordered so that the drives closest to the densest scratch pools are listed first.

**Command:**

- ACSAPI
  - cmd\_proc
  - query mount scratch
- ```
query mount * pool_id [media media_type]
```

Either a specific media type or ALL\_MEDIA\_TYPE (media \* in a cmd\_proc command) can be specified. ANY\_MEDIA\_TYPE is **not** supported.

#### Note:

When ALL\_MEDIA\_TYPE is specified, cartridges with media compatible with the drive are selected to determine the densest scratch pool.

### mount scratch (also known as query mount \*)

For a specified drive, select a scratch (empty) cartridge and mount it. Optionally, select the cartridge from a specified scratch pool and/or with a specified media type. ACSLS selects a compatible scratch cartridge in the closest LSM to the drive. In order to rotate use among all cartridges, the compatible cartridge with the least recent access date is selected within the selected LSM.

**Command:**

- ACSAPI  
cmd\_proc  
mount scratch  
mount \* drive\_id [pool\_id] [media media\_type]
- For an ACSAPI request, *drive\_id*, *pool\_id* and *media\_type* must be specified. (Either a specific media type, ALL\_MEDIA\_TYPE, or ANY\_MEDIA\_TYPE can be specified for *media\_type*.)

- With a `cmd_proc mount *` command, if `pool_id` is not specified, it defaults to the common pool (pool 0).  
A specific media type can be identified. If `media *` is specified, `ANY_MEDIA_TYPE` is chosen. If `media` is not specified, `ALL_MEDIA_TYPE` is chosen.

The special `media_type` values of `ANY_MEDIA_TYPE` and `ALL_MEDIA_TYPE` are processed, as follows:

- When `ALL_MEDIA_TYPE` is specified, a cartridge with a media type compatible with the drive is selected. (This is based on the `media_compatibility` file.)
- When `ANY_MEDIA_TYPE` is specified, the `scratch_preferences` file identifies the preferred list of media to mount on a drive.

## Using these ACSAPI Requests & ACSLS Commands

This section describes how to use the ACSAPI requests and commands for improving tape library performance.

In the following discussion, the term “request” is used for both ACSAPI requests and `cmd_proc` commands.

When mounting a specific cartridge (where the `vol_id` is known):

- Precede the `mount` request with a `query mount` request.
- Pick the first “available” drive, and specify this drive in the `mount` request.

When mounting a scratch cartridge on a specific drive, there are two options:

- To select the closest scratch cartridge to a specific drive:  
Use a `mount scratch` request specifying a drive and, optionally, a scratch pool.

For ACSAPI requests, specify one of the following:

- `ALL_MEDIA_TYPE` (compatible media is selected)
- `ANY_MEDIA_TYPE` (uses the scratch preference list).

For `cmd_proc` commands, select one of the following:

- Do not specify a media type (compatible media is selected)
- Specify `media *` (uses the `scratch preference list`).
- To select a specific drive from a list of drives, and then select the closest scratch cartridge:

Pick the drive, then pick the `scratch tape`: This ensures the drive is close to the scratch cartridge.

- Enter a `query mount scratch` request to identify the closest available drive to the most scratch media in the desired scratch pool.
- Enter a `mount scratch` request to mount a scratch cartridge on the selected drive. Optionally, specify the scratch pool.

For ACSAPI requests, either specify:

- `ALL_MEDIA_TYPE` (and compatible media will be selected), or
- `ANY_MEDIA_TYPE` (to use the scratch preference list).

For `cmd_proc` commands, either:

- Do not specify a media type (and compatible media will be selected), or
- Specify `media *` (to use the scratch preference list).

# L

## Firewall Security Option

The firewall-secure option enables the running of ACSLS behind a firewall while the client software makes requests across that firewall.

Firewall security is also offered to ACSLS clients, allowing them to operate behind their own respective firewalls. This is made available by Oracle to its Independent Software Vendor (ISV) partners. Contact the ISV for your client software component to find out the latest status for each specific client.

### Running ACSLS Behind a Firewall

This Firewall-Secure solution provides the following benefits:

- Enables ACSLS to run behind a firewall (that is, ACSLS on the secure side of firewall, client on opposite side).
- Enables ACSLS client(s) to run behind their own firewall(s) (that is, client(s) on secure side, ACSLS on the opposite side of firewall).

#### Note:

The ISV must have implemented the supplied changes within their client-side software component.

- Preserves compatibility with current ACSLS client implementations, allowing those clients to continue to run with ACSLS in the firewall solution.
- Preserves current ACSAPI/Client functionality and performance. This includes all functionality that is available in a non-firewall environment.

A complete solution would include combining the first two capabilities above. This enables ACSLS and the ACSLS client(s) to each run behind their own respective firewalls (that is, two firewalls between ACSLS and the client(s)), and still have the same communications performance as within a non-firewall environment.

### Addressing Security Areas

ACSLS has addressed the following security concerns:

#### RPC

The use of RPC within ACSLS is a concern for some sites in trying to run within a firewall environment. Preserving compatibility with the current installed client base precludes the ability to remove RPC completely from the ACSLS.

The ACSLS firewall-secure feature has addressed the concerns inherent in RPC, which are:

- The need to allow outside (untrusted) parties to initiate connections to the trusted host across an unrestricted range of ports (1024-65535).
- The exposure of the mapping of available services on a platform through the portmap (or `rpcbind`) daemon running on well-known port 111.

## Security

In a firewall solution, the fundamental security comes from restricting access from the non-secure side into the trusted (secure) side. In all cases, some limited and controlled access must be allowed in order to perform communications and allow data exchange. The goal is to allow that data exchange within a well-defined and restricted set of entry points, allowing you to control those access points and their corresponding communications. This goal is met by this solution.

### Note:

If you have an IPv4-based edge firewall, it should be configured to drop all outbound IPv4 protocol 41 packets and UDP port 3544 packets to prevent Internet hosts from using any IPv6-over-IPv4 tunnelled traffic to reach internal hosts.

## Communications Components

ACSL/Client communications rely on two network interface components to handle network communications between client platforms and the ACSLS platform. Software acting as a client or proxy-server for ACSLS implements one of these two components in order to be compatible with ACSLS platforms and existing clients. The component residing on the client platform is known as the SSI; the component residing on the ACSLS platform is known as the CSI. While it would be desirable to implement all changes within one side (such as the ACSLS platform), in order to maintain client compatibility and to provide all of the firewall-secure features, it is necessary that corresponding changes be made to each side to get the benefits. The positive, is that each side can independently implement the features and achieve the firewall-secure benefits on its own side (such as, changes to the ACSLS allow the ACSLS platform to run behind a secured firewall).

## Benefits of the Firewall-Secure Option

This section describes the benefits of the Firewall-Secure option.

### ACSL/Server-Side

With changes to just the server-side component, as provided within this Firewall-Secure solution, the benefits include:

- Restricting incoming connections for ACSLS communications to a single TCP port for all registered program numbers (there are two registered program numbers for the ACSLS CSI, both of which will be serviced by one single port).
- Allowing users to specify the identity of that port, and configure their firewall in a corresponding fashion.

- Allowing users to turn-off ACSLS communications to UDP ports.
- Allowing users to disable any communication by the ACSLS server to the **client-side** portmapper(s)\* (UDP/TCP port 111). The portmapper must still remain running on client platforms to preserve compatibility with client side code. However, it will not be used for network communications initiated by the server, and therefore, the clients' firewall(s) can be configured to disallow access to it.
- Outgoing connections from the ACSLS server side to the client(s) are unrestricted, for the server-side ports used to preserve current performance. This follows the widely accepted practice by the security community.

## ACSL S Server Port Restriction

This firewall solution restricts the number of incoming ports through which an outside party can initiate network communication. The ports are limited to either one or three: the single customer-specified port for ACSLS incoming requests, plus, possibly, the two portmapper ports (TCP and UDP port 111).



### Note:

To disallow client access to the **ACSL S server** portmapper, thus, disallow access to UDP and TCP ports 111, changes must be made to the **client** software component. See the client-side discussion below.

The server-side of the solution, above, is implemented completely within ACSLS.

## Client-Side (CSC)

The changes made to the CSC place identical restrictions on the client-side platform to those described above. This gives the CSC an identical capability to reside behind its own secure firewall. This solution provides the following benefits:

- Restricts incoming connections for communications (response) to the CSC to a single TCP port for each registered program number. There is one registered program number for the ACSLS SSI.
- End-users can specify the identity of the TCP port and configure their firewall similarly.
- Turns-off client-side communications to UDP ports.
- Disables any communication by the client to the **ACSL S server** portmapper (UDP/TCP port 111). The portmapper must still remain running on the ACSLS platform to preserve compatibility with ACSLS code. However, client network communications are not initiated through the portmapper. Therefore, the ACSLS server firewall can be configured to disallow access to it.
- Outgoing connections from the client side to the ACSLS server are unrestricted for the client-side ports used to preserve current performance.

## Client Port Restriction

This solution restricts the number of incoming ports through which an outside party can initiate network communication. The ports are limited to either one or three: the single

customer-specified port for client incoming responses, and possibly the two portmapper ports (TCP and UDP port 111).

 **Note:**

To disallow ACSLS server access to the **client's** portmapper (and thus disallow access to UDP and TCP ports 111), the changes must be made to the **ACSLs server** software component (see ACSLS server-side discussion above).

This solution has a two-step implementation:

- Oracle StorageTek has made the needed code changes to the CSC Developer's Toolkit 2.3 (or later) source code.
- Clients of ACSLS wanting to provide this security for their client platform must integrate these changes into their client-side SSI code, rebuild the product, and again certify their Client System Component (CSC) with ACSLS.

## Advantages

The client-side and server-side parts of the solution are independent. Therefore, if only one of the two sides is behind a firewall, with respect to the other side, the software changes need to be implemented only on that side. In addition, changing only one side maintains compatibility with all current client and server implementations which already exist and compatibility with other software components which use the CSI/SSI interface.

 **Note:**

This includes compatibility with current Oracle StorageTek products.

This solution does not impact current performance for client/server communications.

## Turning on Firewall Secure Features and Setting Variables

To run the ACSLS server behind a firewall, and, optionally, the ACSLS Client behind a firewall, set variables on both of the ACSLS server and the Client system when they are behind a firewall. These variables enable you to restrict incoming communication to a single port, and optionally, disable the portmapper.

### ACSLs Variables

`CSI_TCP_RPCSERVICE` - Enable CSI support for RPC using the TCP protocol.

- **Function:** Enables the CSI to operate as a TCP RPC Server. If any clients want to communicate with ACSLS over TCP, set this option to `TRUE`.
- **Valid Options:** `TRUE` or `FALSE` (`TRUE` is the default.)
  - `TRUE` enables TCP access for clients to the CSI.

- FALSE disables TCP access for clients to the CSI.
- **Other Details:** The ACSLS product must be restarted for this option to take effect.

CSI\_UDP\_RPCSERVICE - Enable CSI support for RPC using the UDP protocol.

- **Function:** Selecting this option enables the CSI to operate as a UDP RPC Server. If any clients want to communicate with ACSLS over UDP, set this option to TRUE.
- **Valid Options:** TRUE or FALSE (FALSE is recommended.)
  - TRUE enables UDP access for clients to the CSI.
  - FALSE disables UDP access for clients to the CSI.
- **Other Details:** The ACSLS product must be restarted for this option to take effect. The firewall-secure CSI is only supported for TCP communications. Set CSI\_UDP\_RPCSERVICE to FALSE, unless you have legacy client applications inside the firewall with the ACSLS server.

CSI\_USE\_PORTMAPPER – Enable the portmapper.

- **Function:** Selecting this option causes the CSI to interrogate the portmapper when it cannot send a response to a client. If you do not want to allow access to the portmapper on the client, set this option to ALWAYS.
- **Valid Options:** ALWAYS, NEVER, or IF\_DUAL\_LAN\_NOT\_ENABLED
  - ALWAYS means that the portmapper should always be interrogated when the CSI cannot send a response to a client.
  - NEVER means that the portmapper should never be interrogated when the CSI cannot send a response to a client. This option should be selected if clients do not support a port mapper.
  - IF\_DUAL\_LAN\_NOT\_ENABLED specifies that the portmapper should be interrogated only if dual LAN support has not been enabled. If dual LAN support has been enabled, then it is assumed that clients do not support a port mapper.  
IF\_DUAL\_LAN\_NOT\_ENABLED is the default for backward compatibility.
- **Other Details:** The ACSLS product must be restarted for this option to take effect.

CSI\_FIREWALL\_SECURE - Enable the CSI to be used behind a firewall (with a user-defined inbound port).

- **Function:** This option enables the ACSLS server to operate behind a secured firewall. Specify the inbound port used by the ACSLS and limit access to a single port. Configure the firewall to reject incoming ACSLS traffic on all but that port. This ensures that only that port is exposed for use by those outside clients wanting to initiate communications with ACSLS.

To restrict port access, complete the following steps to set up the secure firewall for a specified port:

- Set this option to TRUE.
- Specify the port to be used by the CSI on which incoming ACSLS requests are allowed. (Specified by CSI\_INET\_PORT.)
- For some legacy client applications that do not support fixed port RPC, opening UDP/TCP port 111 in the firewall may be required to support portmapper query requests from the client.
- The firewall-secure CSI is only supported for TCP communications.



Set `CSI_UDP_RPCSERVICE` to `FALSE` unless you have legacy client applications inside the firewall with the ACSLS server.

- Configure the firewall behind which the ACSLS server resides to allow external clients to initiate and receive communications on the previously specified port. Do not forget to set up fixed port on the client application with the same port to minimize open firewall ports.
- Restart ACSLS.
- **Valid Options:** `TRUE` or `FALSE` (Default is `TRUE`)
  - `TRUE` – Restrict access to the ACSLS server to only use a single port for incoming requests from clients.
  - `FALSE` – Do not restrict ports used for client requests to the ACSLS server.
- **Other Details:** The ACSLS product must be restarted for this option to take effect.

`CSI_INET_PORT` - Port number used by the CSI to receive incoming ACSLS requests.

- **Function:** This option specifies the port used by the CSI for incoming TCP requests from clients.
- **Valid Options:** A number between 1024 and 65535, but not 50003. (Default is 30031)
- **Other Details:** The ACSLS product must be restarted for this option to take effect. This variable is only used when firewall-secure CSI is enabled with `CSI_FIREWALL_SECURE` and is set to `TRUE`.

## Displaying and Setting ACSLS Variables

Use the ACSLS `acsss_config` utility or the `dv_config` utility to display, and set the ACSLS static and dynamic variables:

- `dv_config -d`  
Displays all of the ACSLS static and dynamic variables and their settings.
- `dv_config -p <variable_name> -u`  
Prompts you to change a variable, and, if it is a dynamic variable, update ACSLS global shared memory. Enter ? at the prompt to see a full description of the variable. After the full description is displayed, you will be prompted again.

## ACSAPI Client System Variables

The ACSLS client system must be built with the ACSLS CSC Toolkit 2.3 (or later) or a later release for Firewall-Secure Operation to be enabled for the client system.

There are four environment variables to enable for firewall secure operation on the ACSLS Client. You must set these variables to specific values. Each of these variables must be set and exported to the SSI's environment before the SSI process is started. They are then interpreted and used by the SSI, as indicated below.

If your CSC uses a script to start up the SSI, it is recommended that you set and export these variables from within that script. Additionally, Client developers may have provided a method for the end-customer to configure them appropriately, based on the CSC and the environment in which it runs.

`CSI_UDP_RPCSERVICE` –Determines whether UDP is used for network communications.

- **Function:** Enables/disables use of UDP as the underlying network transport layer for SSI network communications.
- **Valid Options:** `TRUE` or `FALSE`
- **Other Details:** This environmental variable must be set to `FALSE` for the firewall-secure CSC. The firewall-secure ACSLS applications packets are all sent using the TCP network transport.

`CSI_TCP_RPCSERVICE` – Determines whether TCP is used for network communications.

- **Function:** Enables/disables use of TCP as the underlying network transport layer for SSI network communications.
- **Valid Options:** `TRUE` or `FALSE`
- **Other Details:** This environmental variable must be set to `TRUE` for the firewall-secure CSC. The firewall-secure ACSLS applications packets are all sent using the TCP network transport.

## New Variables in CSC Toolkit 2.3

`SSI_INET_PORT` – Fixed port number for incoming responses.

- **Function:** Specifies the port the SSI will use for incoming ACSLS responses.
- **Valid Options:** 0 or 1024–65535, except 50001 and 50004.
  - 0 indicates that the previous behavior of allowing the port to be dynamically allocated should remain in effect.
  - 1024 –65535 indicates that number should be used as the TCP port on which the SSI will accept ACSLS responses.
  - DO NOT specify 50001 or 50004, as they are used by the `mini_el` and SSI.
- **Other Details:** Setting this environmental variable to a nonzero value makes the SSI use this port for incoming ACSLS responses. This means that the firewall needs to allow incoming requests on that port in order for the ACSLS responses to be received by the SSI. This is the only port on which the ACSLS will initiate connections with the CSCs' SSI.

### Note:

This value must match the value configured in the firewall which protects the CSC platform, allowing incoming requests for connections on this port.

`CSI_HOSTPORT` –Eliminates queries to the portmapper on the ACSLS server. Instead, send requests to this port on the ACSLS server.

- **Function:** Specifies the port to which the SSI will send its ACSLS requests on the ACSLS server. The ACSLS CSI must be using this port (that is, with firewall-secure fixed port set to this same value) for accepting inbound ACSLS requests from CSCs.
- **Valid Options:** 1024 –65535, except 50003, and 0 (this value must match the value set on the ACSLS server for the port used by the CSI for inbound packets)

- 0 indicates that the previous behavior of querying the portmapper on the ACSLS server will continue to be used.
- 1024 –65535 indicates the value used by the CSI for incoming requests.
- DO NOT specify 50003, as it is used by acslm.
- **Other Details:** Setting this environmental variable eliminates queries from the SSI to the ACSLS servers' portmapper. The value of this variable specifies the port number on the ACSLS server to which the SSI should send its outgoing ACSLS requests. This permits a firewall-protected ACSLS server to disallow access to the portmapper at its firewall. The portmapper query previously provided the port number to which the SSI should direct its ACSLS requests.

 **Note:**

This value must match the value of the port used by the CSI for accepting and servicing incoming requests. The firewall-secure feature must be applied to ACSLS for this port to remain reliably fixed at a specifiable value. If there is a mismatch, there will be no communication between the CSC and ACSLS.

## Displaying and Setting Environmental Variables on the Client

On the client, the commands used to set environment variables depend on your shell and OS.

- On UNIX and Linux, display an environment variable by using the following command:

```
echo ${variable-name}
```

- With the ksh and bash shells, you can set an environment variable by using the following command:

```
<environment_variable> = <value>
```

```
export <environment_variable>
```

## Firewall Secure Solution Scenarios

The following diagrams show possible scenarios of the operation, port usage, and relationship of the ACSLS components when used across a firewall. They are intended to be understood in with the text just presented (above). The "SSI" in the following diagrams is the network interface component of ACSLS that runs on the client-side of the communications. The CSI is the network interface component of ACSLS that runs on the ACSLS platform.

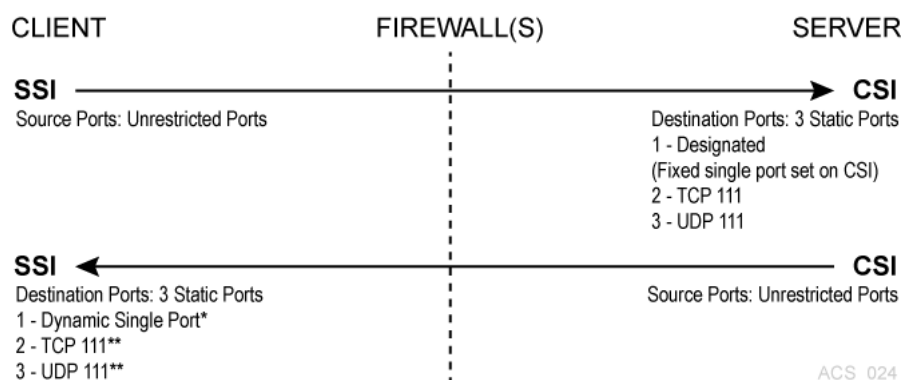
 **Note:**

The ACSLS CSC Developer's Toolkit 2.3 (or later) and the new environment variables are required to support these scenarios.

## Firewall Security on ACSLS Server-Side Only

In this example, firewall security is implemented on the ACSLS server-side (CSI) only. The CSC Toolkit 2.3 (or later) and the new environment variables are **not** required to support this scenario.

**Figure L-1 Firewall security on ACSLS Server-Side only**



In this example, dynamic means that the port is selected by the SSI at startup from the range 1024-65535. The port is not designated by the user, nor is it typically the same port across new executions of the SSI (that is, from one instance of an SSI running process to the next).

The portmapper 111 port(s) on the SSI-side is *only rarely* queried by the CSI. It is only accessed by the CSI in the case where the return port number provided by the SSI in its request packet does not function (that is, results in a network interface failure) for sending the response packets back to the SSI. In this case, as a retry mechanism, the CSI queries the SSI-side portmapper for the port to use, which is registered with the portmapper under the SSI's program number.

To secure ACSLS behind a firewall, you need the following settings:

- ACSLS: ACSLS must be restarted after any changes.
  - CSI\_TCP\_RPCSERVICE = TRUE
  - CSI\_UDP\_RPCSERVICE = FALSE
  - (However, this must be TRUE if you have any clients who use UDP to communicate with ACSLS)
  - CSI\_USE\_PORTMAPPER = ALWAYS (This could be IF\_DUAL\_LAN\_NOT\_ENABLED)
  - CSI\_FIREWALL\_SECURE = TRUE
  - CSI\_INET\_PORT = <1024-65535, not 50003> default 30031
- Client SSI Settings - Environment variables that allow the client to run behind a firewall.
  - CSI\_TCP\_RPCSERVICE = TRUE
  - CSI\_UDP\_RPCSERVICE = TRUE (could be FALSE)
  - SSI\_INET\_PORT = 0

This is a new environment variable with ACSLS CSC Developer's Toolkit 2.3 and not required for this scenario.

- `CSI_HOSTPORT = 0` or `<1024-65535, not 50003>` default 30031

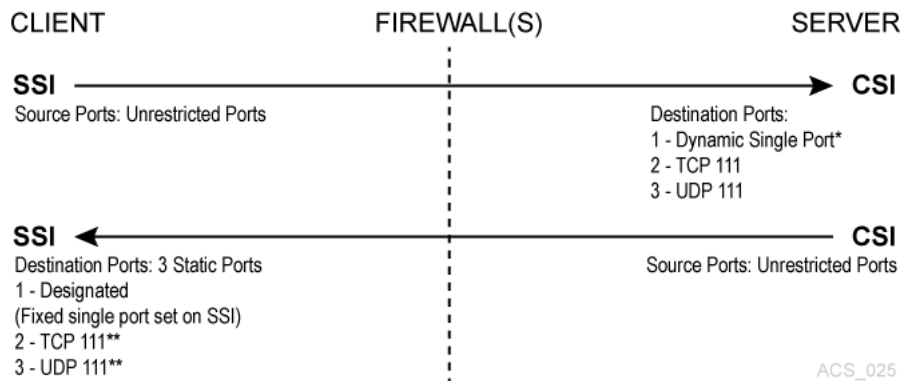
It is not needed if you are using a portmapper on the ACSLS server. If defined and not zero, this must match `CSI_INET_PORT` on the ACSLS Server. This is a new environment variable with ACSLS CSC Developer's Toolkit 2.3 (or later) and not required for this scenario.

Configure your firewall to allow the client to send requests to the ACSLS server using the port specified by `CSI_INET_PORT` (on the ACSLS server) and `CSI_HOSTPORT` (on the client). Allow the client to access the portmapper (port 111) on the ACSLS server and ACSLS to access the portmapper (111) on the client.

## Firewall Security on Client-Side Only

In this example, firewall security is implemented on the client-side (SSI) only. The CSC Toolkit 2.3 (or later) and the new environment variables are required to support this scenario.

**Figure L-2 Firewall Security on Client-Side Only**



In this example, dynamic means that the port is selected by the CSI at startup from the range 1024-65535, and the port is not designated by the user, nor is it typically the same port across new executions of the CSI (from one instance of an CSI running process to the next).

The portmapper 111 port(s) on the SSI-side is only rarely queried by the CSI. It is only accessed by the CSI, in the case, where the return port number provided by the SSI in its request packet does not function (that is, it results in a network interface failure) for sending the response packets back to the SSI. In this case, as a retry mechanism, the CSI queries the SSI-side portmapper for the port to use, which is registered with the portmapper under the SSIs' program number.

To secure the client system behind a firewall, you need the following settings:

- ACSLS: ACSLS must be restarted after any changes.
  - `CSI_TCP_RPCSERVICE = TRUE`
  - `CSI_UDP_RPCSERVICE = FALSE`

If you have any clients using UDP to communicate with ACSLS, this must be TRUE.

- CSI\_USE\_PORTMAPPER = ALWAYS (This could be IF\_DUAL\_LAN\_NOT\_ENABLED)
- CSI\_FIREWALL\_SECURE = FALSE
- CSI\_INET\_PORT = 0

- Client SSI Settings - Environment variables that allow the client to run behind a firewall.

- CSI\_TCP\_RPCSERVICE = TRUE
- CSI\_UDP\_RPCSERVICE = FALSE (could be TRUE)
- SSI\_INET\_PORT = <1024 –65535, except 50001 and 50004>

This is a new environment variable with ACSLS CSC Developer's Toolkit 2.3 (or later) and not required for this scenario.

- CSI\_HOSTPORT = 0 or <1024-65535, not 50003> default 30031

Not needed if you are using a portmapper on the ACSLS server. If defined and not zero, this must match CSI\_INET\_PORT on the ACSLS Server. This is a new environment variable with ACSLS CSC Developer's Toolkit 2.3 (or later) and not required for this scenario.

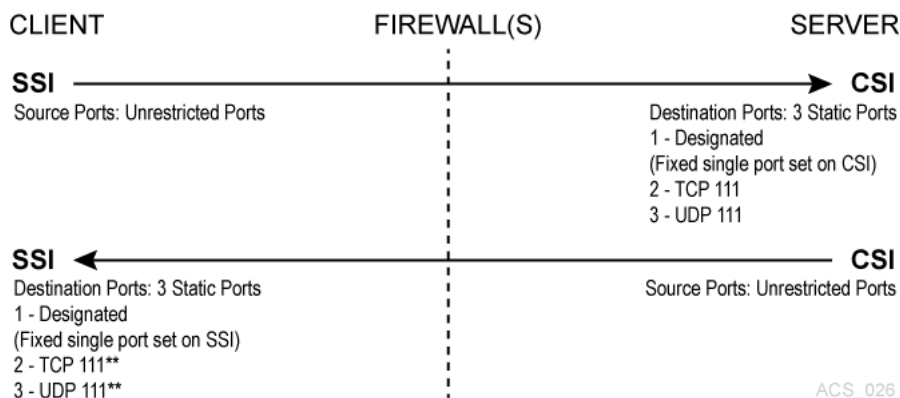
You must configure your firewall(s) to allow:

- The client to send requests to the ACSLS server using the port specified by CSI\_INET\_PORT (on the ACSLS server) and CSI\_HOSTPORT (on the client).
- The client to access the portmapper (port 111) on the ACSLS server.
- The ACSLS server to send responses to the client using the port specified by SSI\_INET\_PORT on the client.
- The ACSLS server to query the portmapper on the client using port 111.

## Firewall Security on Both ACSLS Server and Client-Side with Portmapper

In this example, both the client (SSI) and the ACSLS server (CSI) are implementing Firewall-Secure APIs. The client and server are still relying on the portmapper for port identification. The CSC Toolkit 2.3 (or later) and the new environment variables are required to support this scenario.

**Figure L-3 Firewall Security on both ACSLS Server and Client-Side with Portmapper**



The portmapper 111 that port(s) on the SSI-side is *only rarely* queried by the CSI. It is only accessed by the CSI, in the case, where the return port number provided by the SSI in its request packet does not function (that is, it results in a network interface failure) for sending the response packets back to the SSI. In this case, as a retry mechanism, the CSI queries the SSI-side portmapper for the port to use, which is registered with the portmapper under the SSI's program number.

For firewalls protecting both the ACSLS server and the Client, you need the following settings:

- ACSLS: ACSLS must be restarted after any changes.
  - CSI\_TCP\_RPCSERVICE = TRUE
  - CSI\_UDP\_RPCSERVICE = FALSE

However, this must be TRUE if you have any clients who use UDP to communicate with ACSLS.

  - CSI\_USE\_PORTMAPPER = ALWAYS (This could be IF\_DUAL\_LAN\_NOT\_ENABLED)
  - CSI\_FIREWALL\_SECURE = TRUE
  - CSI\_INET\_PORT = <1024-65535, not 50003> default 30031
- Client SSI Settings - Environment variables that allow the client to run behind a firewall.
  - CSI\_TCP\_RPCSERVICE = TRUE
  - CSI\_UDP\_RPCSERVICE = FALSE
  - SSI\_INET\_PORT = <1024 –65535, except 50001 and 50004>
  - CSI\_HOSTPORT = <1024-65535, not 50003> default 30031

This must match CSI\_INET\_PORT on the ACSLS Server.

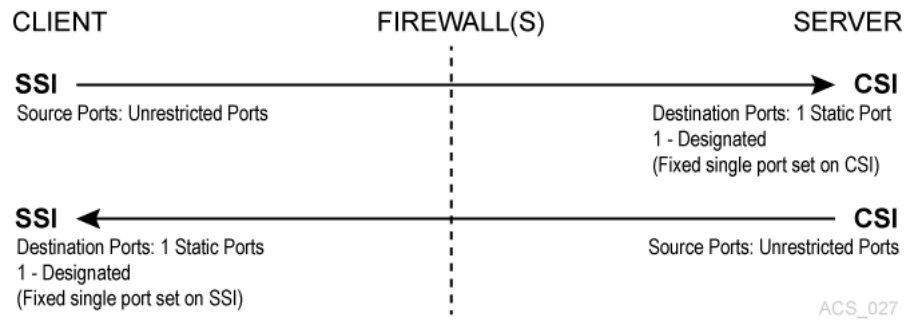
You must configure your firewall(s) to allow:

- The client to send requests to the ACSLS server using the port specified by CSI\_INET\_PORT (on the ACSLS server) and CSI\_HOSTPORT (on the client).
- The client to access the portmapper (port 111) on the ACSLS server.
- The ACSLS server to send responses to the client using the port specified by SSI\_INET\_PORT on the client.
- The ACSLS server to query the portmapper on the client using port 111.

## Firewall Security on Both ACSLS Server and Client-Side without Portmapper

In this example, both Client (SSI) and ACSLS Server (CSI) have implemented the firewall-secure features. The client and server have enabled the “No Portmapper” feature. The CSC Toolkit 2.3 (or later) and the new environment variables are required to support this scenario.

**Figure L-4 Firewall Security on both ACSLS Server and Client-Side without Portmapper**



For the most secure configuration, you need the following settings:

- ACSLS: ACSLS must be restarted after any changes.
  - CSI\_TCP\_RPCSERVICE = TRUE
  - CSI\_UDP\_RPCSERVICE = FALSE
  - CSI\_USE\_PORTMAPPER = NEVER
  - CSI\_FIREWALL\_SECURE = TRUE
  - CSI\_INET\_PORT = <1024-65535, not 50003> default 30031
- Client SSI Settings - Environment variables that allow the client to run behind a firewall.
  - CSI\_TCP\_RPCSERVICE = TRUE
  - CSI\_UDP\_RPCSERVICE = FALSE
  - SSI\_INET\_PORT = <1024 –65535, except 50001 and 50004>
  - CSI\_HOSTPORT = <1024-65535, not 50003> default 30031

This must match CSI\_INET\_PORT on the ACSLS Server.

You must configure your firewall(s) to allow:

- The client to send requests to the ACSLS server using the port specified by CSI\_INET\_PORT (on the ACSLS server) and CSI\_HOSTPORT (on the client).
- The ACSLS server to send responses to the client using the port specified by SSI\_INET\_PORT on the client.

## Turning On the Firewall Security on ACSLS Servers

To turn on the firewall-secure option, you must set several variables using the `acsss_config` utility.

1. Log in as `acsss`.
2. Stop the ACSLS server.



 **Note:**

You must bring down the ACSLS server for the new firewall-secure variables to take effect.

```
acsss disable
```

3. To run the configuration script, enter the following command:

```
acsss_config
```

The ACSLS feature configuration screen appears.

4. Select option 1 - Set CSI tuning variables  
Accept the default for all variables, except for the following.

- a. Set the value to `TRUE` at the following prompt:

```
Changes to alter use of the TCP protocol will not take effect until the
product is restarted.
CSI support for RPC using the TCP protocol is enabled [TRUE].
```

**Variable:** `CSI_TCP_RPCSERVICE`

Turning on TCP insures that the TCP protocol is available for use by clients of ACSLS for network communications. The firewall-secure feature of ACSLS supports TCP only, so clients should perform network communications using this protocol.

- b. Set the value to `FALSE` at the following prompt:

```
Changes to alter the use of the UDP protocol will not take effect until
the product is restarted.
CSI support for RPC using the UDP protocol is enabled [TRUE].
```

**Variable:** `CSI_UDP_RPCSERVICE`

 **Caution:**

Ensure that no ACSLS clients are depending on this UDP protocol. The firewall-secure ACSLS runs on TCP only. Turning off UDP insures that no clients will access the server using this protocol. This enables you to disallow all general UDP access to the ACSLS platform at the firewall, allowing only those accesses which are specifically required in your environment.

Allow clients access to the UDP and TCP port 111 for portmapper access, unless those clients implement the firewall-secure feature, and specifically turn-off their queries to the ACSLS portmapper.

- c. Set the value to `NEVER` at the following prompt:

```
Changes to alter use of the port mapper will not take effect until the
product is restarted.
Enable port mapper: (ALWAYS / NEVER /IF_DUAL_LAN_NOT_ENABLED)
[IF_DUAL_LAN_NOT_ENABLED].
```

**Variable:** `CSI_USE_PORTMAPPER`

**NEVER** enables clients of ACSLS to disallow external access to the portmapper on those client platforms.

**Important:** This does **not** allow you to turn-off external access to the portmapper on the ACSLS platform; to do that, the client(s) of ACSLS must have adopted the firewall-secure changes in the client software component(s), and this feature must be turned on in the client software component.

This feature ensures that the ACSLS server will not make any queries of the portmapper on the client platform. This enables any firewall **which is protecting the client** to disallow access to the portmapper.

- d. Set the value to `TRUE` at the following prompt:

```
Enable CSI to be used behind a firewall (user-defined inbound port) (TRUE/  
FALSE) [FALSE]:
```

**Variable:** `CSI_FIREWALL_SECURE`

`TRUE` enables you to specify the single port that ACSLS will use for accepting inbound client communications (TCP connections). This variable simply enables this feature. The specific port will be specified in the next variable.

- e. Set the value to an available fixed port on the ACSLS server, at the following prompt:

```
Port number used by the CSI to receive incoming ACSLS requests.
```

**Variable:** `CSI_INET_PORT`

This is the port which will be used by the ACSLS CSI component for accepting incoming network connections. Specify a port in the range of 1024-65535, excluding port 50003.

**IMPORTANT:** Configure your firewall to allow incoming connections on this port. This ensures that only that port is exposed for use by those outside clients wanting to initiate communications with ACSLS. You may disallow connections on all other incoming ports except this one, and UDP/TCP port 111 (unless clients have implemented the feature to eliminate their queries to the ACSLS portmapper; in that case, port 111 may also be disallowed at the firewall). The recommended default value for this port is 30031. It is unlikely (but not impossible) that this port will be used by other processes on most systems. See [Troubleshooting](#) for steps to take if there is a port conflict.

5. Select `E` to exit `acsss_config`.  
Your changes are saved.
6. Restart ACSLS by entering the following command:

```
acsss enable
```

## Turning Off Firewall Security on ACSLS Servers

Some variables used above for turning on the firewall-secure feature are also related to turning off that feature. To turn off the firewall-secure behavior, it is only necessary to perform the steps below, but a specific site may want to make modifications to other variables as well.

1. Log in as `acsss`.
2. Stop the ACSLS server.

 **Note:**

Bring down the ACSLS server for the new firewall-secure variables to take effect.

```
acsss disable
```

3. To run the configuration script, enter the following command:

```
acsss_config
```

4. Select option 1 - Set CSI tuning variables  
Change the following values that were set when you configured the firewall-secure feature. Change the following variables

- a. Set the value to ALWAYS at the following prompt:

Changes to alter use of the port mapper will not take effect until the product is restarted.

Enable port mapper: (ALWAYS / NEVER /IF\_DUAL\_LAN\_NOT\_ENABLED)  
[IF\_DUAL\_LAN\_NOT\_ENABLED].

**Variable:** CSI\_USE\_PORTMAPPER

- b. Set the value to FALSE at the following prompt:

Enable CSI to be used behind a firewall (user-defined inbound port)  
(TRUE/FALSE) [FALSE]:

**Variable:** CSI\_FIREWALL\_SECURE

5. Select E to exit acsss\_config.  
Your changes are saved.

6. Restart ACSLS by entering the following command:

```
acsss enable
```

## Firewall-Secure Configuration

The following requires that you are knowledgeable regarding configuring the network firewall behind which ACSLS resides. ALL firewalls are “third-party” software, and will have varying details regarding setting them up correctly for protecting your network environment. The following is not meant to be a recommendation of firewall security policy, but rather, a set of helpful instructions for what the firewall must / can do regarding the ACSLS product, only. See your System Administrator for other security details.

Here is a list of details for setting up your firewall for the ACSLS platform:

- Put in place an overall rule to disallow UDP incoming and outgoing connections.
- Put in place an overall rule to disallow TCP incoming connections (TCP outgoing connections must remain open).
- Put in place a specific rule to allow incoming TCP connections on the port which you specified for the ACSLS usage. **IMPORTANT:** This port **must** match the one you configured under `acsss_config`, or you will receive no client communications at the ACSLS server.

If all of your clients have implemented the firewall-secure feature and make no queries to the ACSLS platform's portmapper, you are done. If the clients still make use of that portmapper on the ACSLS platform, you must add the following:

- Put in place a specific rule to allow incoming and outgoing connections on the well-known portmapper TCP and UDP port 111.

**Example:**

The following is an example of the rules which were put in place for an iptables-based firewall in order to put all of the above rules in place.



**Note:**

These are in addition to other rules configured for the specific firewall.

```
echo " - FWD: Allow all connections OUT and only existing/related IN"
$IPTABLES -A FORWARD -i $EXTIF -o $INTIF -m state --state \
ESTABLISHED,RELATED -j ACCEPT
# These rules allow client access to the portmapper
$IPTABLES -A FORWARD -p tcp -i $EXTIF --dport 111 -j ACCEPT
$IPTABLES -A FORWARD -p udp -i $EXTIF --dport 111 -j ACCEPT
# These rules allow client access to the ACSLS CSI for network communication
# Note: This assumes that the CSI firewall-secure port was specified as 30031
$IPTABLES -A FORWARD -p tcp -i $EXTIF --dport 30031 -j ACCEPT
# Catch all rule, all other forwarding is denied and logged.
$IPTABLES -A FORWARD -j drop-and-log-it
```

## Troubleshooting Firewall-Secure Communications

Troubleshooting a network communications interface which includes the ACSLS platform and clients, and now includes intervening firewall(s), may involve multiple steps. By introducing the firewall(s) into the path between ACSLS and its clients, there are more potential causes for network communications failures. Additionally, there are more components that must be configured in a way that corresponds with the settings in other components, and if these settings do not match, the network communications will be impacted. Here is a list of things to check and try if you have done all the configuration work on ACSLS, its client(s), the firewall(s), and network communications are not working.

**1. Checking the ACSLS platform:**

- Is the ACSLS up and running? If not, check the `acsss_event.log` for possible reasons, or for pointers to a possible culprit.
- Is the CSI being brought up successfully by ACSLS? If not, there should be informative messages in the `acsss_event.log` which point towards the cause. Bad values for some configuration parameters or a port conflict are likely possible causes.
- Is there a port conflict being reported in the `acsss_event.log` which causes the CSI to fail? If so, you should use the "netstat" or similar system utility to tell you which ports are in use on the system, and configure the ACSLS to use an available port. Remember to reconfigure the firewall to match.
- Is the CSI registering for the port you expect? Use the command `'rpcinfo -p'` to look at the portmap table. The CSI is registered under program number 300031. Check to

make sure that the port registered under that program number is the one you expect (the default port is 30031, with one less zero than the program number).

2. If ACSLS and the CSI are up and running and are correctly registered, the next step would be to check access to the ACSLS platform across the firewall:
  - Is the ACSLS reachable through basic RPC? Use the "`rpcinfo -t <hostname> <program-number> <version-number>`" command to send a simple RPC request to the CSI. (Use "`man rpcinfo`" on your system to get more information on the `rpcinfo` command and its use.) Do this from a machine on the inside of the firewall with ACSLS (such as, from the ACSLS platform itself), and from outside the firewall. If it works from inside but not from outside, then the firewall is intercepting your ACSLS requests. Double-check the configuration of the firewall and the ACSLS port. Check to be sure that the portmapper is accessible through the firewall (this test cannot be used from outside the firewall if access to the portmapper is disallowed).
  - Do the ports configured for ACSLS and for the firewall, match? Double-check these parameters. This is a likely cause of failure in network communications. Aside from the configured values, perform the '`rpcinfo -p`' command mentioned above, to insure that the CSI is indeed registering with the expected portnumber. If it is not, look in the `acsss_event.log` for information about the cause.
  - Is the ACSLS receiving requests, but unable to send back responses? If you check the `acsss_event.log` and find that the CSI reports many dropped network packets or failures to communicate with network clients, then the client requests are getting in, but the responses are not getting out. Again, this is an indication that they are being blocked by a firewall.

3. If your problems are still not resolved.

The above addresses several levels of things to look for. If these yield no specific answer, it is time to do some lower-level checking to find out where communications are being broken-down. The best way is through the use of a network packet sniffer facility, such as 'snoop' under Solaris. Use "`man snoop`" on your Solaris-based system to get more information on the `snoop` command and its use.

Similar packet tracing facilities are available on other network-connected systems.

- To use this, you have to do your packet sniffing from locations that show you where the packets are getting to, and where they are being lost. This may be from both inside and outside of the firewall.
- Additionally, looking at the packet data will be informative. If either side is allowing use of the portmapper, it is likely that you will see some PORTMAP packets.
- Also, you should see RPC packets passing between the ACSLS and its clients.
- Finally, looking at the transport-level TCP connection will inform you of the specific ports being used on each side for the connection. This is often critical information to find out where the communications are being stopped.

More detail on performing these operations is beyond the scope of this manual, but your System Administrator should be able to provide some help in this area.

## Frequently Asked Questions

- Why do I need the Firewall-Secure solution for ACSLS?  
The Firewall-Secure solution enables you to effectively run the ACSLS behind a firewall, and enables you to restrict ports on that firewall so that security is significantly enhanced.
- What releases of ACSLS will support the Firewall-Secure feature?  
Only ACSLS 7.0.0 and above supports this feature.
- What is the maximum number of ports I will have to leave open if I use this firewall-secure feature?  
The maximum number of ports on which you might have to allow incoming network connections is three: one for the ACSLS network interface, and two for the portmapper (UDP and TCP 111). Outgoing ports are unrestricted, in accord with accepted industry security practices.
- What is the minimum number of ports I can leave open?  
The minimum number is one. This is possible if your clients (ISV software) have also implemented the Firewall-Secure features in their client, and make no queries to the portmapper which resides on the ACSLS platform. When that is the case, the only port that need be open for incoming connections is the one user-specified TCP port used by the ACSLS network interface.
- Why doesn't the feature use a range of ports?  
There is no architectural advantage to using a range of ports, and there are some security disadvantages. The non-firewall-secure ACSLS uses a range of ports which consists of the full range of dynamic ports available on any given platform. This is correctly perceived as a potential compromise to the security of a site. Restricting this as much as possible, without adversely affecting ACSLS performance, is the goal in order to eliminate that compromise. Since the ACSLS network interface uses only one incoming port at any given time, there is no reason to extend the range beyond one port, provided that port is dedicated to ACSLS use for the ACSLS platform.
- What if the port I choose conflicts with another usage of that port on my system?  
This is one of the reasons that the port is made user-specifiable. The specific ports available will vary from one customer site to another. The user is not allowed to use one of the well-known reserved ports from 0-1023. The default port of 30031 falls within the range of registered ports, which makes it less likely (though not impossible) that another application which uses dynamic ports use it. Although it is in the range of registered ports, there is no application registered to use it, which makes it a reasonable default selection.
- Does this feature allow me to protect my ACSLS server with a firewall?  
Yes, with this feature in place, your ACSLS server can be put on the trusted side of a firewall, with clients accessing it from the opposite (untrusted) side or from the same side.
- Does this feature allow me to protect my ACSLS clients (ISV components) with a firewall?  
Potentially, yes, but not by itself. In order to realize this scenario, your client software components (clients of the ACSLS) must have adopted the Firewall-Secure feature, which has been made available using the StorageTek ACSLS Client System Component Developer Toolkit. Contact your client software provider for a current update on their status.

- If I want to be able to protect my clients with a firewall, what should I do?  
You should contact your client software provider. They can tell you whether they have adopted any Firewall-Secure changes in their CSC (client software component).
- What about the portmapper? Can I completely disallow access to the portmapper?  
If your clients have adopted the Firewall-Secure changes, they may allow you to shut off the client's queries to the ACSLS platform's portmapper. In that case, you may disallow access to the portmapper on the firewall which protects the ACSLS platform. In any other case, the clients will depend on the ACSLS server side portmapper to help them make a connection with the ACSLS network interface, and it must be available for their use.
- Why must the client implement some changes in order for my ACSLS server firewall to shut down access to the ACSLS platform portmapper?  
Because it is the client that is making these queries of the ACSLS platform. If the client continues to make these queries, the ACSLS platform must continue to provide the portmappers' services in order for those queries to succeed.
- I think the portmapper is bad. Why didn't you remove it completely?  
The portmapper provides an important service to legacy clients. Removing it completely would invalidate the interface on which those clients depend. In short, no legacy clients would work without recoding, retesting, and again certifying with the new non-portmapper interface. In this firewall-secure solution, we have provided the capability to remove the queries to the portmapper from both the ACSLS to the client, and from the client to the ACSLS, but we cannot force client software to conform to this. Thus, the portmapper must remain available at least as an optional service until a site's clients have adopted the Firewall-Secure features and no longer make use of the portmapper service.
- Some of my clients have adopted the Firewall-Secure features and some have not. How can I take advantage of this?  
Those clients which have adopted these features may be protected behind their own respective firewalls. In addition, access to the portmapper's well-known ports may be restricted at the firewall, and then configured to allow access to the portmapper only by those clients who require it. The details and ability varies based on the specific firewall in use at the site.
- I think RPC is bad. Why didn't you remove it completely?  
The ACSLS network interface has been RPC-based since the first release of ACSLS. It has proven to be an effective, stable, and reliable mechanism, offering various advantages at the network communications layer. However, it can also be more difficult to secure a platform which uses RPC due to its common dynamic allocation of ports and use of the portmapper. In this Firewall-Secure solution, both of these areas are addressed, which enables the customer to effectively configure a firewall in a restricted fashion, yielding the security benefits for which they have the firewall in place.  
Additionally, complete removal of RPC from the ACSLS network interface would invalidate all current (legacy) ACSLS clients, making it impossible for any of them to communicate with ACSLS without recoding, retesting, and again certifying their CSCs (client software components).
- How will the Firewall-Secure feature affect network communications performance and timing between my ACSLS clients and the ACSLS server?

There is no effect on performance due to the new Firewall-Secure features. The usage of a firewall may have performance implications, but this will be based on the operational characteristics of each specific customers' firewall implementation. With a firewall which has negligible impact on performance, the ACSLS and its clients will continue to perform as they did before installing the Firewall-Secure feature. Also, the ACSLS network interface tolerances can be configured, so that delays imposed by the firewall could be handled gracefully.

- How does the Firewall-Secure feature affect the rest of my ACSLS operations?

There is no effect or impact on other parts of the ACSLS operations due to the installation of the Firewall-Secure solution.

- How does the Firewall-Secure feature affect the ACSLS functionality that my clients use (through the ACSAPI)?

The full set of functionality that is provided through the ACSAPI (and which our ACSLS clients use today to interface with ACSLS) will operate the same under the Firewall-Secure feature as it does without the feature installed. In particular, this Firewall-Secure feature supports access control, and also all of the newer features that have been added to the ACSLS product. The full functionality of the ACSAPI will continue to be supported by this feature.

- Does the Firewall-Secure feature work with the ACSLS HA (High Availability) solution?

The Firewall-Secure feature does not adversely affect HA operation. However, the HA solution is not designed to be run across a firewall (that is, with each HA server on opposite sides of a firewall). The HA solution requires remote access to the portmapper, so the firewall could not disallow that access if an attempt were made to run each server on opposing sides of a firewall. There are other details of running across a firewall that could adversely affect an HA setup; it is highly recommended that this not be done.

If the HA servers are set up on the same secured side of the firewall, that set of HA servers could be set up with the Firewall-Secure feature, and clients on the opposite side of the firewall would be able to interact across the firewall with those servers with the same performance and behavior as they would against a non-firewall-secure HA solution.

- Does this Firewall-Secure feature work with other StorageTek software products?

Inter-operability with other StorageTek products and partner products (that is, client software components which communicate with ACSLS) has been completely preserved. Those products can continue to operate without modification, communicating with the ACSLS server, with the ACSLS server running behind a secured firewall, or in the same environment with those products (as it does today).

- Do other StorageTek software products have the same Firewall-Secure features?

Other StorageTek products do not gain the Firewall-Secure benefit simply by being used in the same environment with the Firewall-Secure ACSLS. Each product can work with a firewall secured ACSLS (see previous question), but putting each of those products behind its own respective firewall is a question that the specific product itself must address. Some StorageTek products already have built-in policies which allow some restriction at a firewall used to protect the platforms where those products run. Additionally, any product which acts as a client to ACSLS has the option of adopting the Firewall-Secure changes which were made to ACSLS, and which are provided as part of the ACSLS CSC Developer's Toolkit 2.3 (or later).



# M

## CSCI

This chapter describes the Client Server Communications Interface (CSCI) component.

This chapter covers the following topics:

- Overview of CSCI
- Error messages
- Environmental variables

Installation of CSCI should follow installation of the operating system and installation and configuration of ACSLS.



### Note:

For ACSLS to function properly, the `START_CSCI_PROCESS` configuration variable must be set to `TRUE`. You do this in Option 1 of `acsst_config`. See [Setting Variables that Control ACSLS Behavior](#).

## Overview of CSCI

The Client Server Communications Interface (CSCI) is a stand-alone component responsible for the passing of request and response packets between a client system and a server system. CSCI is an optional component that lets MVS or RMLS/CSC clients communicate with ACSLS. If you do not use CSCI, you use CSI by default to communicate with ACSLS.

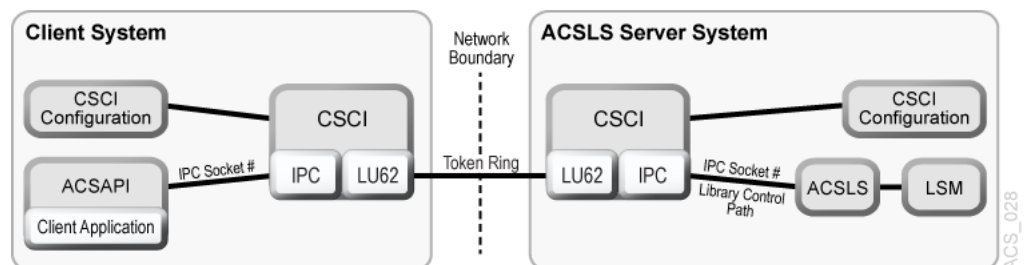
## CSCI Connectivity

CSCI provides IPC and TCP/IP connectivity to link the ACSLS server with MVS clients or RMLS/CSC clients

## Architecture and Subsystems

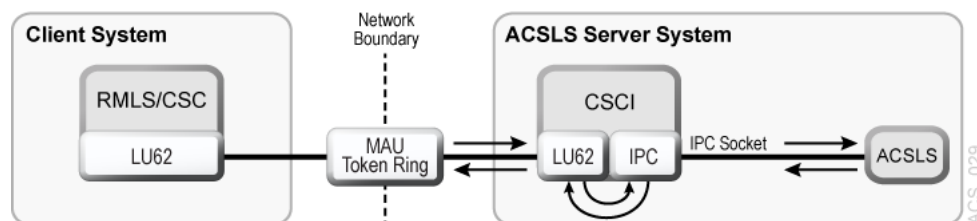
The following figure shows an overall CSCI architecture and subsystems.

**Figure M-1 CSCI Architecture and Subsystems**



The following figure shows how the CSCI system connects with the network interface. On the server-side, the data flows from the network hub through the TCP/IP interface, to the CSCI interface. From here, the data flows through an IPC interface to the ACSLS software in both directions, but in a half duplex mode, for example: one direction at a time.

**Figure M-2 CSCI System with Token Ring Interface**



## Error Messages

CSCI installation and configuration error messages relate to the following conditions:

- Improper ownership
- Improper access permission
- Undefined environment variables
- A variety of miscellaneous conditions
- Improper CSCI and TCP/IP configuration

## Common Error Messages

```
Communication failure, verb = cmroutine, rc = xx, errno = yyy IPC failure
on Socket CSCIZzzzz
```

Explanation: The read failed on the named connection.

Variable:

- `xx` is IBM CPI return code
- `yyy` is the UNIX errno value in `/usr/include/sys/errno.h`
- `CSCIZzzzz` is the socket number that was in use
- `cmroutine` is the IBM CPI routine name that failed

```
Read failed for connection: connection_name
```

Explanation: The read failed on the named connection.

Variable: `connection_name` is the name of the connection on which the read failed.

```
Write failed for connection: connection_name
```

Explanation: The write failed on the named connection.

Variable: `connection_name` is the name of the connection on which the write failed.

## Environmental Variables

CSCI is invoked by `csciTcpServer.sh`, a Transaction Program Name (TPN) in the `/$ACS_HOME/bin` directory where ACSLS executables reside. This shell script defines CSCI environment variables.

The following table describes CSCI environment variables for the server subsystem and a shell script similar to the one provided in the distribution of ACSLS/CSCI software. This shell script is started by TCP/IP on the server.

**Table M-1 CSCI Server Subsystem Environment Variables**

Name	Description
<code>START_CSCI_PROCESS</code>	Automatically starts CSCI at ACSLS startup (TRUE or FALSE)
<code>CSCI_TRACE</code>	This variable indicates whether CSCI tracing is ON or OFF.
<code>CSCI_SLEEP_TIME</code>	This variable indicates the PER sleep time at the end of each round-robin loop through the connection table (polling timeout). Default is 100 msec.
<code>CSCI_INITIAL_CONNECTION</code>	This variable indicates the CSCIs' initial connect name to open first. This is the connection service name requiring i/o on initial startup of the CSCI. Clients usually open IPC first; Servers open TCP/IP first.
<code>CSCI_SERVER_NAME</code>	This variable indicates the name of the CSCI Server. It must be set the same for all CSCIs' Clients defined to a single CSCI server. The server name must match its CSCI Clients <code>CSCI_SERVER_NAME</code> . The matching of the Server and Client <code>CSCI_SERVER_NAME</code> provides an end-to-end logical link between machines. Finally, this name is used later for the "connectionname" in other environment variables. The value assigned to the <code>CSCI_SERVER_NAME</code> must be replicated in the connectionname part of the subsequent variables.
<code>CSCI_connectionname_NET_TYPE</code>	This variable indicates what Network type this CSCI is configured for from the CIF subsystem. The NETTYPE defines the transport layer used by this CSCI. Valid assignments are LU62 or IPC.
<code>CSCI_connectionname_INPUT_SOCKET</code>	This variable indicates the input socket number to use as input to this CSCI. This number would match the Server or Client application output socket number.
<code>CSCI_connectionname_CON_TYPE</code>	This variable indicates this CSCI type. This <code>CON_TYPE</code> defines the CSCI to be either a SERVER or a CLIENT.
<code>CSCI_connectionname_TRANSLATE_TYPE</code>	This variable indicates this CSCI XDR type to be employed during execution. This can be assigned to either XDR or NONE.
<code>CSCI_connectionname_DESTINATION_SOCKET</code>	This variable indicates destination or output socket number to be used by the CSCI Server. This socket number would be the matching input socket number to the CSCI Server Application.

# N

## Media Management

This chapter discusses how to manage the same type of media recorded in different densities and how to separate cartridges written in the old density from cartridges written in the new density.

### Overview

In this overview, you gain an understanding of:

- Restrictions and the possible problems you may encounter
- Examples of tape drives that record on the same media in different densities
- Recommended solutions

### Restrictions

Often, new tape drives are introduced that use existing tape media, but record at a higher density. These new drives can often read tapes recorded in the old density, but they cannot write in the old density. The older tape drives can neither read nor write in the higher density.

Because of these restrictions, you may encounter the following problems:

- When a tape written in the new density is mounted on an older drive, the older drive cannot read the tape.
- When storage management applications try to fill-up partially used tapes by appending additional files at a later time, it will fail if the tape was written using a different density than one to which a new tape drive can read but cannot write.

If you have a mix of both older and newer tape drives in your libraries, you must manage your tape cartridges that are of the same media type.

### Examples

The following examples show tape drives that record on the same media, in different densities:

- T10000 Media used by T10000A and T10000B Tape Drives  
The T10000B tape drive uses the same media as the T10000A, but writes data at double the T10000A's density. The T10000B can read T10000A media and can reclaim (write from beginning of tape) it for writing T10000B density data, but it cannot append data to a previously written T10000A.  
The T10000A drive can reclaim a T10000B cartridge for writing T10000A density data, but can neither read from, nor append data to a T10000B cartridge.
- 9840 Media used by T9840C and T9840D Tape Drives  
Manage 9840 media when a combination of T9840A, T9840B, T9840C, and/or T9840D drives are present, because:

- T9840C  
The T9840C uses the same media as the T9840A and T9840B, but records at double the density.
- T9840D  
The T9840D writes at almost double the T9840C drives density.  
Manage 9840 media when a combination of T9840A, T9840B, T9840C, and/or T9840D drives are present.

## Solution

ACSLS has tools you can use to manage common media on which two or more types of drives can write, but are in incompatible recording densities. The client application must use these tools to manage the read/append data incompatibilities.

Manage drives that record on common media in different densities within an ACS by either:

- Replacing all of the older drives in an ACS with the new drives at the same time.  
This is the simplest and safest strategy. Using this strategy, you avoid the problems caused by managing a combination of drives using different densities. If you are unable accomplish this, you can gradually replace the older drives with the new drives, as discussed in the second bullet.

 **Note:**

Do not append files to any tapes written in the old recording density after replacing the old drives with the new drives. With Veritas NetBackup, this is done by *suspending* the tapes.

- Gradually replacing the older drives with the new drives.  
This requires you to manage the common media recorded in different densities. You can do this by:
  - Using the recording format information returned when the cartridge is dismounted in SL8500 and SL3000 libraries to manage media. This is discussed in the next section.
  - Creating separate ACSLS pools for each format.
  - Using the facilities of a backup application (for example, Veritas NetBackup, Legato NetWorker, IBM Tivoli, or CA BrightStor) to manage media pools.

## Managing Common Media with Recording Densities Reported by ACSLS

When a cartridge is dismounted from a modern tape drive in a modern library, the recording format of the cartridge is reported to ACSLS. ACSLS saves the recording format in its database. You can use the `display volume` command to display this information.

Recording format is reported by:

- Libraries:
  - SL3000
  - SL8500 (with 4.10 or later firmware)
- Tape Drives:
  - All T10000 tape drives (with 1.38 or later firmware)
  - T9840C and T9840D

The example below describes how to manage T10000 media recorded in both T10000C and T10000D formats.



#### Note:

The discussion below uses the following syntax conventions for commands and utilities:

- Commands and Utilities entered as is are in **bold**.
- Variables (you must enter the exact value) are in *italics*.

## Procedure

After the T10000D drives are installed, use the recording format information to mount cartridges on compatible tape drives. Again, cartridges written at T10000C density can be read by either T10000C drives or T10000D drives, but only T10000C drives can append data written in a T10000C format. Only T10000D drives can read or append to cartridges written in a T10000D format.

No special processing is necessary when mounting scratch cartridges. When you write the cartridge from beginning-of-tape, the previous recording format does not matter.

To select a drive to read or append to a cartridge with data already recorded on it, follow this procedure. The following commands identify drives that can read and write a cartridges recording format.

### Mounting Non-scratch Cartridges

1. Display the recording format of the cartridge.

```
display volume vol_id -f recording_format_family recording_format_model
```

The recording format for this volume is displayed.

2. Use the following `query` command to identify a compatible drive:

```
query mount vol_id
```

Drives compatible with a cartridge are displayed.

- Select the correct drive type before mounting the cartridge.

This `query` command returns both T10000C and T10000D drives (because both drive types are compatible with T2 media).

- Use the drive type to separate T10000Cs from T10000Ds.

Selecting the first compatible drive in the list minimizes pass-thru and improves library performance.

3. Mount the cartridge on a drive compatible with the recording format with the following command:

```
mount vol_id drive_id
```

#### Migrating to the New Recording Format

- When all the data on the cartridge has expired, you can re-write the cartridge from beginning-of-tape, in a new recording format.  
For more information on expired cartridges, refer to [Populating the LSM](#) .
- After all T10000C drives are converted to, or replaced by T10000D drives, all T2 media can be used by T10000D drives if data is not appended to cartridges written in T10000C format.

## Managing Common Media Recorded in Different Densities with ACSLS Pools

The following procedure applies to all cases where different tape drives record on a common media, in different densities. Use the following table to apply the procedure to your specific case:

Old Format Drive(s)	New Format Drive(s)
T10000C	T10000D
T9840C	T9840D
LTO 7	LTO 8

### Procedure

1. Identify all LTO data (non-scratch) cartridges in the ACS, and assign them to a LTO 7 pool. Do this before installing LTO 8 drives in an ACS that has LTO 7 drives.

Now you can assign LTO scratch cartridges to either the LTO 7 or LTO 8 pool.

- a. Define LTO 7 and LTO 8 media pools using the following command:

```
define pool pool_id
```

- b. Report all LTO 7 media.

You can do this by using either the `display volume` command (option 1) or a custom `volrpt` (option 2), as shown below:

Results write a file.

- Option 1

Use the `display volume` command to write the results to a file.

```
display volume * -media LTO-6.4T > filename
```

where:

All LTO 7 tape cartridges (media LTO-6.4T) are reported.

*filename* is the filename to which the output is written. The **ACS ID** in which the cartridge is located is also listed.

Read the output, selecting cartridges from specific ACS(s), if desired.

If you want cartridges from one ACS, use the `-home` operand to only select cartridges from this ACS.

```
display volume * -home acs_id.*,*,*,* -media LTO-6.4T> filename
```

- **Option 2**

Use the custom `volrpt` for all volumes in the ACS. The results write a file:

```
volrpt -d -f custom_volrpt_file -a acs_id > filename
```

where:

*custom\_volrpt\_file* is the filename specifying the fields reported in a custom `volrpt`. The following fields are reported:

VOLUME_ID	6	2
MEDIA_TYPE	7	2
VOLUME_TYPE	4	2

*acs\_id* is the ID of the ACS that you are managing.

*filename* is the filename to which the output is written.

Read the output, selecting only the volumes with a media type of LTO-6.4T.

- c. **Assign the selected volumes to the appropriate pools.**

Assign all non-scratch (`VOLUME_TYPE = D`) cartridges to the LTO 7 pool using the following command:

```
set scratch off pool_id vol_id
```

Assign the scratch cartridges (`VOLUME_TYPE = S`) to either the LTO 7 or LTO 8 pools using the following command:

```
set scratch pool_id vol_id
```

- 2. **After the LTO 8 drives are installed, use the pools to mount cartridges on compatible tape drives.**

Again, cartridges written at LTO 7 recording density can be read by either LTO 7 drives or LTO 8 drives.

The following commands can be used to identify, select and mount cartridges from the pool appropriate for an LTO 7 or LTO 8 drive. Use the pools to ensure you are mounting cartridges on the correct drive.

 **Note:**

If you want to mount a scratch cartridge on the correct drive type, follow procedure A. If you want to mount a non-scratch cartridge with data already recorded on it, follow procedure B.

**Procedure A - Mounting Scratch Cartridges**

- a. **Query drive(s) to identify drive types.**



```
query drive drive_id | all
```

- b.** Identify the correct pool (media type) for the cartridges to be mounted on the desired drive.
- c.** Mount scratch cartridges to a drive from the specified pool.

```
mount * drive_id pool_id
```

#### Procedure B - Mounting Non-scratch Cartridges

- a.** Display the status of drives compatible with a cartridge.

```
display volume vol_id -f pool
```

The pool for this volume is displayed.

- b.** Identify a compatible drive using the query command.

```
query mount vol_id
```

Drives compatible with a cartridge are displayed.

Select the correct drive type before mounting the cartridge. This query command returns both LTO 7 and LTO 8 drives (because both drive types are compatible with LTO-6.4T media).

Use drive type to separate T9940As from T9940Bs.

- c.** Mount the cartridge on the selected drive.

```
mount vol_id drive_id
```

- 3.** When all of the data on the cartridge has expired, you can migrate the scratch cartridges to the pool for the new recording density. Because LTO 7 and LTO 8 drives can re-initialize cartridges in their density, scratch cartridges can be reassigned to a different pool.

```
set scratch pool_id vol_id
```

- 4.** After all LTO 7 drives are converted to, or are replaced by LTO 8 drives, all LTO-6.4T media can be used by LTO 8 drives.



# XAPI Client Interface

This chapter introduces the XAPI client interface to ACSLS.

## The XAPI Client Interface to an ACSLS Server

The XML API (XAPI) is an API that enables StorageTek clients and servers to communicate using a common protocol over TCP/IP. ACSLS 8.4 and later releases can be configured with XAPI support.

Clients who were required to use an MVS based server (HSC for real tape and/or VTCS for virtual tape) can now use ACSLS to manage StorageTek libraries.

Refer to the *StorageTek Enterprise Library Software XAPI Client Interface to ACSLS Server* document in the ELS documentation set for details about XAPI and XCMD configuration, administration, and operator commands and the supported XAPI user requests and ACSLS XAPI control variables.

## The ACSLS XAPI Service

Refer to the *StorageTek ACSLS 8.5 Installation Guide* for procedures for installing and uninstalling the ACSLS XAPI service.

When installed, the XAPI component is started and stopped using ACSLS.

- To start ACSLS with the XAPI service, enter the following at the Unix command prompt:

```
acsss enable
```

- To stop ACSLS and the XAPI service, enter the following at the Unix command prompt:

```
acsss disable
```

- To stop ACSLS, the XAPI service, the ACSLS database, and all ACSLS components, enter the following at the Unix command prompt:

```
acsss shutdown
```

## XAPI Variables

Use ACSLS the `acsss_config` or `dv_config` utility to display, and set the ACSLS XAPI static variables.

You must restart ACSLS for the changes to take effect.

- `dv_config`
  - `dv_config -d` to display all ACSLS variables.
  - `dv_config -p <variable_name>` to update the XAPI variable.

See [Setting Variables that Control ACSLS Behavior](#).

- `acsss_config`

Select option 9 to display and change XAPI variables. This option only displays if you have installed the XAPI server.

ACSLIS Feature Configuration

Enter the number followed by Return for your choice from the following menu to configure product behavior in that area.

Press ? followed by the Return key for help.

```
1: Set CSI tuning variables
2: Set event logging variables
3: Set general product behavior variables
4: Set access control variables
5: Set automatic backup parameters
6: Rebuild Access Control information
7: Event Notification settings
8: Define or Change Library Configuration
9: Set XAPI server variables
E: Exit
```

Menu choice:

The XAPI variables are:

- `XAPI_PORT`

Prompt: *Changes to the user-defined inbound port to the XAPI server will not take effect until the XAPI server is restarted. Port number used by the XAPI server to receive incoming XAPI requests. [50020]:*

This option specifies the port used by the XAPI server for incoming TCP requests from clients. Enter a number between 1024 and 65535 to define the port used by the XAPI server. Do **not** specify port 50003.

The XAPI server must be restarted for this variable to take effect.

- `XAPI_WORK_PATH`

Prompt: *Changes to the XAPI work directory will not take effect until the xapi server is restarted. Place the XAPI log and trace information in which directory. [\$ACS\_HOME/log/xapi]:*

Select a directory where XAPI server work files will be placed. As installed, XAPI server logs information to the `$ACS_HOME/log/xapi` directory. In normal usage, the value of this variable won't be changed. An alternate path may be specified if there are disk space problems in the file system which contains `$ACS_HOME`. The path given must be an absolute path (one which starts with / or `$ACS_HOME`).

The XAPI server must be restarted for this variable to take effect.

- `XAPI_LOG_SIZE`

Prompt: *Changes to the XAPI log size will not take effect until the xapi server is restarted. Maximum XAPI log size in Mbytes. [20]:*

This option specifies the threshold size for the XAPI Log, expressed in Mbytes (here defined as "1048576 bytes."). Enter a non-negative number. 20 is the default value of this option.

The XAPI server must be restarted for this variable to take effect.

- XAPI\_LOG\_FILE\_NUM

Prompt: *Number of XAPI Log archive files to retain. [10]:*

This option specifies the number of archived XAPI Log files to retain. When the current vlog.file size exceeds the threshold size, the log file is renamed with a 0 to n suffix. 0 is the newest file and n is the oldest. The archived files are saved in the XAPI\_WORK\_PATH directory. Once the specified number of archived logs is reached, the oldest file will be removed from the archive directory each time a new one is added to that directory. You can retain no fewer than one archived file and no more than 99. Enter a number between 1 and 99 to specify the number of archived log files to retain.

The XAPI server must be restarted for this variable to take effect.

- XAPI\_TRACE\_SIZE

Prompt: *Changes to the XAPI trace size will not take effect until the xapi server is restarted. Maximum XAPI trace size in Mbytes. [50]:*

This option specifies the threshold size for the XAPI Trace, expressed in Mbytes (here defined as "1048576 bytes."). Enter a non-negative number. 50 is the default value of this option

The XAPI server must be restarted for this variable to take effect.

- XAPI\_TRACE\_FILE\_NUM

Prompt: *Number of XAPI Trace archive files to retain. [10]:*

This option specifies the number of archived XAPI TRACE files to retain. When the current vtrace.file size exceeds the threshold size, the trace file is renamed with a 0 to n suffix. 0 is the newest file and n is the oldest. The archived files are saved in the XAPI\_WORK\_PATH directory. Once the specified number of archived logs is reached, the oldest file will be removed from the archive directory each time a new one is added to that directory. You can retain no fewer than one archived file and no more than 99. Enter a number between 1 and 99 to specify the number of archived log files to retain.

The XAPI server must be restarted for this variable to take effect.

- XAPI\_STARTUP\_FILE

Prompt: *Changes to the XAPI startup file name will not take effect until the xapi server is restarted. Name of the XAPI startup file with control parms. [xapi\_startup\_file]:*

This option specifies the name of the XPI startup file. This file resides in the XAPI\_WORK\_PATH directory and includes XAPI startup parameters.

The XAPI server must be restarted for this variable to take effect.

- XAPI\_TAPEPLEX\_NAME

Prompt: *Changes to the XAPI Tapeplex name will not take effect until the xapi server is restarted. Name of the XAPI Tapeplex. [ ]:*

This option specifies the name of the XAPI Tapeplex. The XAPI server must be restarted for this variable to take effect.

Please enter text whose length does not exceed 8 characters.

# P

## Accessibility Features in ACSLS

Oracle strives to provide high quality information technology that is accessible to people with disabilities. Specific accessibility features in ACSLS are summarized here.

### User Interfaces

This section discusses the ACSLS user interfaces.

#### The ACSLS (text-only) Command-line Interface (CLI)



##### Note:

`cmd_proc` can run in curses mode or line mode. For accessibility, you must run `cmd_proc` in line mode by specifying the `-l` attribute:

```
cmd_proc -l
```

See [Using `cmd\_proc`](#) for more information about `cmd_proc` modes.

The primary CLI in ACSLS is known as `cmd_proc`. An ACSLS operator can manipulate all physical library operations from the `cmd_proc`, including the following:

- Tape movement activities such as mount, dismount, enter, eject and move operations.
- Query operations for tape drive and tape volume location and status, lsm status, lmu status, and CAP status.
- Control operations to vary all physical library resources online and offline and the ability to switch between multiple library interfaces (switch lmu).
- Tape administrative activities such as tape ownership, pool administration, lock functions for job control, and management of cleaning cartridges.

Administrative functions that are not accessible from the CLI include the following:

- Logical library administration: functions to create a logical library, to assign and unassign tape drives, and assign and unassign tape volumes to logical libraries.
- Logical library operation: functions to mount and dismount logical volumes to configured drives.

A second CLI, `lib_cmd`, is used in lieu of the ACSLS GUI to manage logical libraries. Using `lib_cmd`, you can:

- Create or delete a logical library.
- Edit attributes of a logical library.
- Assign and unassign volumes or drives to a logical library.

- Define mappings between a client application and a logical library.
- Display the status of a logical library.
- Display the status of volumes or drives in a physical or a logical library.
- Vary a logical library or an assigned drive online or offline.

## Primary Administrative Interface: Unix Shell Functions

Primary functions to install and configure ACSLS, to backup and restore the ACSLS database, and to operate auxiliary utilities are accessible through the standard UNIX shell (sh, csh, ksh, and bash) under Solaris.

## The ACSLS Graphical User Interface (GUI)

The ACSLS GUI is divided into three frames:

- The top frame always displays the ACSLS GUI masthead with five titled navigation buttons, and summary information including user id, server id, and dashboard-level status.
- The left-most frame always displays the expandable Navigation Tree.
- The content displayed in the right-hand frame changes dynamically as the user asserts content selections from the Navigation Tree or the preferences button from the masthead.

The Web-based GUI enables the user to perform all logical library operations and most physical library operations. GUI library operations include the following:

- Administrative functions to create a logical library and to assign or unassign volumes to a logical library, and to set volume ownership or cleaning attributes.
- Physical and logical tape movement activities such as mount, dismount, enter, and eject.
- Query operations for physical tape drive and tape volume location and status, lsm status, lmu status, and CAP status.
- Query operations for physical library resources that have been assigned to a logical library.
- Control operations to vary physical and logical library resources online and offline.

Administrative functions that are not accessible from the GUI include the following:

- The ability to switch between multiple library interfaces (switch lmu).
- Specific tape administrative activities including pool administration and lock functions for job control.

These tasks are handled through the `cmd_proc` interface.

## Accessibility Provisions in the ACSLS GUI

- Every page displayed in the content frame is identified with a title.
- A text equivalent tool tip is provided for every graphical element in the GUI. A screen reader will make the alternative text available (using the mouse to hover over the respective graphical element will also show that tooltip).

- All data tables displayed in the content frame are identified with column and row headers.
- The expandable Navigation Tree can be configured to be fully expanded by default.
- The Masthead includes color coded dashboard elements to indicate faulty status conditions. Color-blind users can look beside each status icon where any number higher than zero signifies the same message, while displaying how many library resources share the errant status.

## Special Steps Required to Configure Accessibility Modes in ACSLS

This section discusses the special steps required to configure accessibility modes in ACSLS.

### GUI Tree Menu

The ACSLS GUI provides an expanded or collapsed tree menu. By collapsing the tree menu, the complexity of the menu is hidden behind a few initial selections which become expanded as the menu portions are selected. For sight-impaired users who may rely on keyboard navigation or an audio summary of the menu, a collapsed tree menu may keep large portions of the GUI hidden from view. To enable an expanded tree menu, follow this procedure:

1. Locate the Preferences button in the ACSLS GUI Masthead. Clicking the Preferences button will cause the Preferences page to be displayed in the GUI.
2. In the Per-user Preferences portion of the Preferences frame, locate the 'Default Tree Menu' drop-down menu.
3. Click the drop-down arrow and select "expanded".
4. Once the expanded choice has been selected, click the "set" button.

# Glossary

## **absent cartridge**

A cartridge that is in the database, but that couldn't be found when all recorded locations for the cartridge were catalogued. If a nonzero retention period is set, the volume status is changed to `STATUS_VOLUME_ABSENT`.

## **ACS**

See Automated Cartridge System.

## **ACSEL**

See ACS Event Logger.

## **ACS Event Logger (ACSEL)**

The software component that receives messages from other ACSLS components and writes them to an Event Log.

## **ACS ID**

A unique identifier for an ACS.

## **ACSLH**

See ACS Library Handler.

## **ACS library**

A library is composed of one or more ACSs, attached tape drives, and cartridges residing in the ACSs.

## **ACS Library Handler (ACSLH)**

The part of the ACSLM that communicates directly with the LMU.



**ACSLM**

See ACS Library Manager.

**ACS Library Manager (ACSLM)**

The software component that validates and routes library requests and responses.

**ACSLs**

See ACS Library Software.

**ACSLs database**

ACSLs database containing information about the location and status of the data or cleaning cartridges. The information includes cell location, scratch status, and so on).

**ACSLs platform**

The server hardware and software that provide the proper environment for ACSLS.

**ACS Library Software (ACSLs)**

Manages contents of multiple libraries and controls library hardware to mount and dismount cartridges on ACS tape drives.

**ACSLs database**

A database used by ACSLS to track the library configuration and the locations and IDs of all data or cleaning cartridges in the library.

**ACSSA**

See ACS System Administrator.

**ACS System Administrator (ACSSA)**

The interface between the Command Processor and the rest of the system.

**ADI**

Application Data Interchange.

**audit**

A physical inventory of the contents of all or part of a library.

**Automated Cartridge System (ACS)**

The library subsystem consisting of one or more libraries connected through pass-thru-ports.

**automated library**

See library.

**backing library**

Identifies the physical library from which the logical library is created.

**beginning of tape (BOT)**

The location on a tape where written data begins.

**BOT**

See Beginning of Tape.

**CAP**

See Cartridge Access Port.

**CAP ID**

A unique identifier for the location of a CAP. A CAP ID consists of the ACS ID, the LSM number, and the CAP number.

**cartridge**

A plastic housing containing a length of data recording tape. The tape is threaded automatically when loaded in a transport. A plastic leader block is attached to the tape for automatic threading. The spine of the cartridge can contain an OCR/Bar Code label listing the volume ID.

**Cartridge Access Port (CAP)**

A bidirectional port built into the door panel of an LSM, which provides for the manual entry or automatic ejection of data or cleaning cartridges.

**cartridge drive (CD)**

A device containing two or four cartridge drives and their associated power and pneumatic supplies.

**cartridge tape I/O driver**

Operating system software which issues commands (for example, read, write, and rewind) to cartridge subsystems.

**cartridge transport**

An electromechanical device that moves tape from a cartridge over a head that writes and reads data from the tape. A transport is distinct from the power and pneumatic sources that supply the electricity and air it needs to function. See cartridge drive.

**CCI**

See client computing system.

**CD**

See cartridge drive.

**cell**

A receptacle in the LSM in which a cartridge is stored.

**channel**

A device that connects the host and main storage with the input and output control units.

**client applications**

Software applications that manage tape cartridge contents. They access tape cartridges by interacting with ACSLS. Any number of client applications can be resident on a client system.

**client computing system**

A computer and an executable image of the operating system.

**client software**

This software manages tape cartridge contents, generates requests for cartridges, and drives data to and from cartridges. The client software is not part of ACSLS.

**Client System Component**

Software which provides an interface between the client computing system's operating system and ACSLS.

**Client System Interface (CSI)**

The software component that translates and routes messages between the ACS Library Manager and the Client System Component.

**command access control**

Limits access to commands.

**command area**

The bottom area of the `cmd_proc` interface where you enter requests and receive responses.

**command processor (cmd\_proc)**

The screen interface of the ACSSA. `cmd_proc` lets you enter system commands.

**complex partitioning**

Partitioning across multiple SL8500 libraries connected via pass-thru ports in a library complex. This also includes support for up to 16 partitions.

**control path adapter**

A hardware device which converts a Client Computing System's control protocol to the control protocol of the StorageTek Library Control System.

**control unit (CU)**

A microprocessor-based unit logically situated between a channel and up to fifteen cartridge transports. The CU translates channel commands into transport commands and sends transport status to the channel.

**CSE**

Customer Services Engineer.

**CSC**

Client System Component.

**CSI**

See Client System Interface.

**CSI variables**

Used to define various options to fine-tune communications between a CSC and the CSI. You change these variables in the `acsss_config` program.

**CU**

See control unit.

**cycle error messages**

Messages that indicate a library or ACSLS failure.

**database**

A collection of interrelated data records. See also ACSLS Database.

**data path**

The network path that enables client applications read/write access to tape cartridges.

**data path adapter**

A hardware device which translates a Client Computing System's data protocol to the data protocol of the StorageTek Control Unit.

**display area**

The top area of the `cmd_proc` interface that collects messages regarding the status of the library.

**Dual TCP/IP**

provides two separate host connections between the host software (ACSL or HSC) and a library

**dynamic configuration**

enables you to implement configuration changes to ACSLS libraries (and components) while ACSLS remains online and running.

**ejected cartridge**

A cartridge that has been ejected from the library. If a nonzero retention period is set, the cartridge status is changed to `STATUS_VOLUME_EJECTED`.

**end of tape (EOT)**

The location on a tape where written data ends.

**EOT**

See end of tape.

**EPO**

Emergency Power Off.

**EPROM**

See erasable programmable read only memory.

**erasable programmable read-only memory (EPROM)**

A special memory chip that can be erased and reprogrammed.

**Event Log**

A file, maintained by the ACSEL, that contains messages describing library and ACSLS events.

**Event Logger**

See ACS Event Logger.

**external label identifiers**

A six-character alphanumeric label on the outside edge of a cartridge used to identify a physical tape cartridge. It may consist of uppercase letters A through Z, numerals 0 through 9, \$, #, and blanks.

**Fast Load**

With Fast Load enabled, an FC initiator that issues a mount operation receives a successful response once the operation has been validated and accepted by ACSLS, but before cartridge movement begins.

**full installation**

A complete software installation required for new customer sites or for existing sites where a new library has been installed.

**HLI**

Host/Library Interface. One way that the ACSLS communicates with a library.

**HLI-attached**

Libraries that are connected to the ACSLS through the HLI. These libraries can be connected through a serial interface (serial-attached) or through a TCP/IP interface (TCP/IP-attached).

**home location**

The cell associated with a given cartridge.

**Host Software Component (HSC)**

Software running on an IBM mainframe that controls multiple libraries as a library server.

**ID**

Identifier or identification.

**Initial Program Load (IPL)**

A process that activates a machine reset, initiates wake up diagnostics (from EPROMs) and loads functional code.

**inline diagnostics**

Routines that test components of a subsystem while operating on a time-sharing basis with the functional microcode in the subsystem component.

**in-transit cartridges**

Cartridges between their source and destination locations. Cartridges are considered in-transit if they are in pass-thru ports, robot hands, or playground.

**I/O**

Input/Output.

**IPC**

Interprocess Communication.

**IPL**

See Initial Program Load.

**journal**

A sequential log of changes made to the database since the last checkpoint.

**LAD**

Lock Access Door.

**LAN**

See local area network.

**large CAP (LCAP)**

A 40-cartridge CAP with the storage cells arranged in four removable magazines of ten cells each. The magazines appear as a single column of 40 cells to the host software.

**LCAP**

See large CAP.

**LCU**

See Library Control Unit.

**LED**

See Light Emitting Diode.

**library**

A library is composed of one or more ACSs, attached tape drives, volumes in the ACSs, and the ACSLS software that controls and manages the ACSs.

**LibAttach**

A software component, resident on a Windows NT client system, that translates and routes messages between client applications and the CSI.

**library configuration options**

Customer can specify the number of ACSs in the library and the connections between each ACS and the server system.



**library control component**

Software which controls the mounting and dismounting of cartridges in the ACS.

**library control processor**

Properly configured computer hardware that, with the addition of appropriate software, supports the operation of the Library Control Software.

**library control system**

The library control platform loaded with library control software (ACSL).

**library control software**

The software components of ACSL including the library control component, the Client System Interface and Library Utilities.

**Library Control Unit**

The portion of the LSM that controls the picking, mounting, dismounting, and replacing of data and cleaning cartridges.

**library drive**

A cartridge transport attached to an LSM that is connected to, and controlled by, a client system. Library drives interact with the LCU during automated tape cartridge mount and dismount operations. Library drives interact with a client application during tape data transfer operations. Library drives are individually addressable by the ACSLM and are individually accessible by client applications. See Cartridge Transport.

**library errors**

Errors that occur because the library is offline, has suffered hardware failure, is unavailable, and so forth.

**Library Management Unit (LMU)**

The portion of an ACS that manages LSMs, allocates their resources, and communicates with ACSL.

**Library Storage Module (LSM)**

An ACS structure that provides the storage area for cartridges, cartridge drives, CAPs, and the robot necessary for moving them.

**light emitting diode (LED)**

A light emitting device that uses little energy and is used mainly to indicate on/off conditions.

**LMU**

See Library Management Unit.

**local area network (LAN)**

A computer network in which any component in the network can access any other component. This is the type of interface between an LMU and attached LSM's.

**LSM**

See Library Storage Module.

**LSM ID**

A unique identifier for an LSM. The LSM ID consists of the ACS ID and the LSM number.

**media validation**

Feature that verifies T10000 cartridges using a designated pool of T10000C and T10000D drives. The validation provides a success or failure result for each tape cartridge tested

**missing cartridge**

A cartridge that is in the database, but couldn't be found. If a recorded possible location for the cartridge could not be examined due to an offline LSM or a drive not communicating, the cartridge is marked missing instead of absent. The cartridge status is changed to `STATUS_VOLUME_MISSING`.

**Multiple TCP/IP**

Using TCP/IP connections to multiple libraries to provide redundant communication paths between the host software (ACSL or HSC) and an SL8500 library complex.

**network adapter**

Equipment that provides an electrical and logical interface between a network and specific attached equipment.

**Network Interface (NI)**

An interface between the server system and the client systems that maintains network connections and controls the exchange of messages. The NI is resident on the server system and each client system.

**NI**

See Network Interface.

**OCR**

Optical character recognition.

**ONC**

Open network computing.

**Open Systems Interconnection (OSI)**

A software architecture model of the International Organization for Standardization. The OSI model provides standards for the interconnection of data processing systems.

**OSI**

See Open Systems Interconnection.

**OSLAN**

Open Systems Local Area Network.

**Partition**

Partition of a library cell's, cartridges, drives, and CAPs assigned to a physical partition that is managed by ACSLS as a separate ACS.

**Pass-Thru Port (PTP)**

Mechanism that enables a cartridge to be passed from one LSM to another in a multiple LSM ACS.

**PCAP**

See priority CAP.

**playground**

A reserved area of special cells (within an LSM) used for storing diagnostic cartridges and cartridges found in-transit upon power-on and before initialization of the LSM is completed.

**pool**

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

**POST**

Power-on self-test.

**priority CAP (PCAP)**

A single-cartridge CAP used for priority entry and ejection of cartridges.

**processing errors**

Errors that result from processing or network communication failures.

**PROM**

Programmable read-only memory.

**PTP**

See Pass-Thru Port.

**RDBMS**

Relational database management system.

**redo log files**

Backup files used to restore the ACSLS database.

**relational database**

A database that is organized and accessed according to relationships between the data items; relationships are represented by tables.

**Redundant Electronics**

The optional SL8500 Redundant Electronics (RE) feature provides failover protection in enterprise libraries. RE uses a two sets of library controller cards. At any given time, one set is active and the other set is standby. The active library controller can failover to the standby

in response to a command from ACSLS or the SL Console. Automatic failover can be initiated by the library if a library card fails.

**ROM**

Read-only memory.

**RPC**

Remote Procedure Call.

**SCAP**

See standard CAP.

**scratch**

An attribute of a tape cartridges, indicating that it is blank or contains no useful data.

**SCSI**

Small computer serial interface.

**Serial-attached**

See HLI-attached.

**server system**

The part of the library that is the residence for ACSLS, now referred to as the Library Control System. The Library Control System acts as an interface between a library and client systems.

**server system user**

A person who invokes ACSLS commands, utilities, or procedures on the server system. Server system users are generally site and maintenance personnel (for example, library operators, tape librarians, system administrators, CSEs, and systems personnel).

**servo**

A system that uses feedback to control a process.

**silo**

A commonly used term for an LSM. See Library Storage Module.

**SIMM**

Single inline memory module.

**SLOT**

see cell.

**SQL**

See structured query language.

**SRN**

See service request number.

**SSI**

See Storage Server Interface.

**SSR**

Software Support Representative.

**Standard CAP (SCAP)**

A 21-cartridge CAP with the storage cells arranged in three rows of seven fixed cells.

**Storage Server Interface (SSI)**

A software component, resident on a client system, that translates and routes messages between client applications and the CSI.

**structured query language (SQL)**

A language used to define, access, and update data in a database.

**StorageTek Library Console**

the operator panel software application used for the StreamLine libraries.

**system resource variable**

Used to control the amount of system resources used by ACSLS.

**system unit**

The Library Control Platform.

**tape library management system (TLMS)**

A type of client application.

**TCP**

Transmission Control Protocol.

**TLMS**

See tape library management system.

**TOD**

Time of day.

**UDP**

User Datagram Protocol.

**UNIX**

An operating system originally developed by Bell Laboratories (now UNIX Systems Laboratories, Inc.) and used by a variety of computer systems.

**unsolicited messages**

Messages that indicate an error or notify you when a particular routine action can be taken.

**UOC**

Usable on codes.

**upgrade installation**

Performed when installing a new version of ACSLS at an existing customer site.

**user**

selectable features and options variables-Used to define various user-selectable features and options.

**validation errors**

Errors that result from format and syntax validation performed by cmd\_proc.

**venter**

Virtual enter. Entering an unlabeled cartridge with a virtual label.

**virtual label**

A logical volume ID (volser) that can be assigned to a cartridge when its physical label is missing or unreadable.

**volser**

Volume Serial Number.

**volume**

A data or cleaning cartridge.

**volume access control**

Limits access to volumes, usually by the client.

**volume identifier**

A six-character string that uniquely identifies a data or cleaning cartridge to the database.

**volume serial number (volser)**

A synonym for external label identifier.

**WTM**

write tape mark.

**XDR**

External data representation.



**XML**

Extensible Markup Language. A universal format for structured documents and/or data on the Web.

# Index

## A

---

absent cartridge, [8-20](#)

access control

enabling

ownership.assignments file, [7-1](#)

access control information, rebuilding, [5-18](#)

accessibility, [P-1](#)

ACS states, [12-45](#)

ACSLs

audit, [7-3](#)

commands, [12-1](#)

database recovery, [9-12](#)

directory structure, [1-7](#)

dual-LAN client configuration, [7-10](#)

extended store feature, [7-4](#)

mixed-media restrictions for SCSI-attached

LSMs, [7-7](#)

mixed-media support, [7-6](#)

reconfiguring

general product behavior, [5-8](#)

logging options, [5-6](#), [5-16](#), [5-17](#)

scratch preferences, [7-7](#)

utilities, [11-1](#)

ACSLs commands

audit, [12-5](#)

cancel, [12-11](#)

clear lock, [12-14](#)

define pool, [12-16](#)

delete pool, [12-18](#)

dismount, [12-19](#)

eject, [12-23](#)

enter, [12-27](#)

idle, [12-31](#)

lock, [12-32](#)

logoff, [12-35](#)

mount, [12-36](#)

mount \*, [12-38](#)

move, [12-42](#)

query, [12-44](#)

query acs, [12-45](#)

query cap, [12-47](#)

query clean, [12-49](#)

query drive, [12-50](#)

query lmu, [12-51](#)

ACSLs commands (*continued*)

query lock, [12-54](#)

query lsm, [12-56](#)

query mount, [12-57](#)

query mount \*, [12-59](#)

query pool, [12-61](#)

query port, [12-63](#)

query request, [12-64](#)

query scratch, [12-65](#)

query server, [12-66](#)

query volume, [12-67](#)

set cap mode, [12-69](#)

set cap priority, [12-71](#)

set clean, [12-73](#)

set commands, [12-69](#)

set lock, [12-74](#)

set owner, [12-76](#)

set scratch, [12-76](#)

show, [12-78](#)

start, [12-80](#)

switch lmu, [12-81](#)

unlock, [12-83](#)

uptime, [12-85](#)

vary, [12-85](#)

ACSLs utilities

acs\_partition.sh, [11-3](#)

acs\_renumber.sh, [11-4](#)

acsss, [11-7](#)

bdb.acsss, [11-8](#)

config acs, [11-11](#)

config drives, [11-13](#)

config lsm, [11-14](#)

config ports, [11-15](#)

db\_export.sh, [11-16](#)

db\_import.sh, [11-17](#), [11-18](#)

del\_vol, [11-19](#)

drives\_media.sh, [11-21](#)

ejecting.sh, [11-21](#)

free\_cells.sh, [11-24](#)

getHba.sh, [11-25](#)

greplog, [11-27](#)

moving.sh, [11-28](#)

probeFibre.sh, [11-36](#)

rdb.acsss, [11-36](#)

showDevs.sh, [11-39](#)

ACSLs utilities (*continued*)

- showDrives.sh, [11-40](#)
- stats\_report, [11-40](#)
- userAdmin.sh, [11-42](#)
- volrpt, [11-43](#)
- watch\_vols, [11-48](#)

acssa user ID, [1-1](#)

acsss command, [11-7](#)

acsss user ID, [1-1](#)

acsss\_config, [5-1](#), [M-1](#)

- enabling or disabling auto-clean, [8-14](#)
- for dynamic changes to drive configurations, [11-13](#)
- logging volume statistics report, [10-3](#)
- menu, [4-5](#), [5-2](#)
- run after adding, removing, or swapping modules, [E-6](#)
- stats report, [11-41](#)

adding scratch volumes, [8-17](#)

addresses, translating, [D-5](#)

AEM, [D-5](#)

architecture, CSCI, [M-1](#)

audit command, [12-5](#)

audit, when to perform, [7-3](#)

auditing the library, [7-3](#)

- audit intervals, [7-4](#)
- running an audit, [7-3](#)

automatic database backup, [9-11](#)

## B

---

backup

- automatic, [9-11](#)
- manual to tape, [9-11](#)
- specified tape device attached to ACSLS server, [9-12](#)
- UNIX file, [9-12](#)

Backup and Recovery Tools

- automatic backups, [A-1](#)
- disaster scenarios
  - database corrupted-need to restore database to working state, [A-4](#)
  - ran acsss\_config against wrong library, [A-4](#)
  - server failure - rebuilding another ACSLS server with new hardware, [A-5](#)
  - server failure - rebuilding the same server with new hardware, [A-4](#)
- manual backups, [A-2](#)
- manual database exports, [A-2](#)
- using db\_import.sh, [A-3](#)
- using rdb.acsss, [A-3](#)

bdb.acsss utility, [11-8](#)

## C

---

cancel command, [12-11](#)

CAP

- bulk, [C-6](#)
- cap\_state, [12-47](#)
- displaying information about, [8-5](#)
- modes, [8-3](#)
- priorities, [8-4](#)
- set mode, [12-69](#)
- set priority, [12-71](#)
- states, [8-2](#)
- types, [8-2](#)

cartridge

- recovery, [8-22](#)
- status reporting, [8-21](#)

cartridges

- absent, [8-20](#)
- applying policies, [8-9](#)
- ejected, [8-20](#)
- ejecting, [8-7](#)
- entering, [8-5](#)
- expired, [8-24](#)
- loading into disabled LSM, [8-29](#)
- scratch, [8-16](#)

cell, display, [13-7](#)

checkGui.sh, [I-10](#)

cleaning cartridges, [12-73](#)

- checking, [12-49](#)

cleaning tape path, manual mode, [8-13](#)

cleaning transports, [8-10](#)

- manually, [8-13](#)

cleaning transports in SCSI-attached LSMs, [8-14](#)

clear lock command, [12-14](#)

cmd\_proc

- logoff, [12-35](#)
- shortcut keys, [1-4](#)
- using, [1-2](#)

command processor

- shortcut keys, [1-4](#)
- using, [1-2](#)

command syntax, [12-2](#)

commands, ACSLS, [12-1](#)

config acs, [11-11](#)

config drives utility, [11-13](#)

config lsm, [11-14](#)

configuration

- updating, [4-3](#)

Configuration

- configuring your library hardware, [5-19](#)
- defining event notification settings, [5-18](#)
- menu, [5-2](#)
- rebuilding access control information, [5-18](#)
- setting access control variables, [5-16](#)
- setting automatic backup variables, [5-17](#)

Configuration (*continued*)  
 setting CSI tuning variables, [5-3](#)  
 setting event logging variables, [5-6](#)  
 setting general product behavior variables,  
[5-8](#)

configuring  
 tape drives  
 layouts, [C-5](#)

CSCI  
 architecture, [M-1](#)  
 environmental variables, [M-3](#)  
 error messages, [M-2](#)  
 overview, [M-1](#)

## D

---

database  
 automatic backup, [9-11](#)  
 backup to specified tape device, [9-12](#)  
 backup to UNIX file, [9-12](#)  
 deleting volumes, [11-19](#)  
 exporting, [9-2](#)  
 importing, [9-5](#), [A-1](#)  
 installing, [9-1](#)  
 manual backup to tape, [9-11](#)  
 recovering and restoring, [9-12](#)  
 recovery, [9-12](#)  
 restoring to most recent backup, [9-13](#)  
 verifying, [9-10](#)  
 verifying and library configuration, [9-10](#)

db\_export.sh utility, [9-2](#), [11-16](#)  
 db\_import.sh utility, [11-17](#), [11-18](#)

define pool command, [12-16](#)

del\_vol utility, [11-19](#)

delete pool command, [12-18](#)

deleting scratch pools, [8-18](#)

diagnostics

get\_diags, [I-8](#)

dismount command, [12-19](#)

display cap command, [13-4](#)

display cap information, [8-2](#)

display cell command, [13-7](#)

display command references, [13-1](#), [14-1](#)

display commands, [13-2](#)

display cap, [13-4](#)

display cell, [13-7](#)

display drive, [13-8](#)

display lock, [13-13](#)

display lsm, [13-14](#)

display panel, [13-17](#)

display pool, [13-18](#)

display port, [13-20](#)

display volume, [13-21](#)

display drive command, [13-8](#)

display lock command, [13-13](#)

display lsm command, [13-14](#)

display panel command, [13-17](#)

display pool command, [13-18](#)

display port command, [13-20](#)

display volume command, [13-21](#)

displaying scratch information, [8-17](#)

drive

clear lock, [12-14](#)

configuration, [11-13](#)

display, [13-8](#)

lock, [12-32](#), [12-54](#), [12-74](#)

state, [12-50](#)

status, [12-50](#)

unlock, [12-83](#), [12-85](#)

Dual TCP/IP Support on SL3000, [D-12](#)

Dual TCP/IP Support on SL8500, [B-2](#)

dual-LAN client configuration, [7-10](#)

dual-LMU, [12-51](#)

dynamic configuration

config acs, [11-11](#)

config drives, [11-13](#)

config lsm, [11-14](#)

config ports, [11-15](#)

dynamic configuration utilities, [11-10](#)

dynamic variables

setting access control, [5-16](#)

setting automatic backup, [5-17](#)

setting CSI, [5-3](#)

setting event logging, [5-6](#)

setting event notification, [5-18](#)

setting general product behavior, [5-8](#)

## E

---

eject command, [12-23](#)

ejected cartridge, [8-20](#), [8-23](#)

ejected volumes, [8-7](#)

enabling access control, ownership.assignments  
 file, [7-1](#)

enter command, [12-27](#)

enter command, manual vs automatic, [12-28](#)

entering cartridges, [8-5](#)

event log, [5-6](#)

event log, search, using greplog, [9-1](#)

event notification, [7-12](#)

Expired Cartridges, [8-24](#)

exporting

database to disk file, [9-3](#)

database to tape, [9-4](#)

exporting the database, [9-2](#)

Extended Store, [7-4](#), [K-3](#)

extended store feature, [7-4](#)

---

## F

Fast Load, [G-6](#)

firewall security

ACSLs, [L-1](#)

benefits, [L-2](#)

communications, [L-17](#)

communications concerns, [L-2](#)

RPC concerns, [L-1](#)

scenarios, [L-8](#)

security concerns, [L-2](#)

setting variables, [L-4](#)

turning off, [L-15](#)

turning on, [L-13](#)

float, [K-3](#)

frequently asked questions, [L-19](#)

---

## G

GUI

starting, [3-1](#)

troubleshooting tips, [I-11](#)

---

## H

HLL-PRC, description of, [C-3](#), [D-5](#)

---

## I

idle command, [12-31](#)

importing

database from disk file, [9-6](#)

db\_import.sh, [11-17](#), [11-18](#)

importing the database, [9-5](#), [A-1](#)

installing Solaris, preparation

Solaris 8, [6-1](#)

internal addresses

description, [C-3](#), [D-5](#)

invoking access control, ownership.assignments  
file, [7-1](#)

IP addresses, registering for primary and  
secondary LANs, [7-10](#)

---

## K

keyboard shortcuts, cmd\_proc, [1-4](#)

---

## L

lib\_cmd, [14-1](#)

Library Hardware

configuring, [4-1](#)

library server utilities, general information, [11-1](#)

LMU, switch, [12-81](#)

loading a CAP, [8-2](#)

loading cartridges into a disabled LSM, [8-29](#)

lock

clear, [12-14](#)

command, [12-32](#)

lock command, [12-32](#)

logical libraries, [G-1](#)

benefits, [G-1](#)

command line interface, [14-1](#)

creating, [G-2](#)

deleting, [G-6](#)

lib\_cmd, [14-1](#)

limitations, [G-2](#)

logoff command, [12-35](#)

LSM

display, [13-14](#)

populating, [8-1](#)

query, [12-56](#)

---

## M

managing a mixed-media library, [7-6](#)

managing scratch volumes, [8-16](#)

manual enter, [12-27](#)

manual volume delete utility, [8-23](#), [11-19](#)

media management, [N-1](#), [N-4](#)

media validation, [7-16](#)

miscellaneous ACSLS files, recovery, [9-14](#)

missing cartridge, [8-21](#), [8-22](#)

mixed media

files, [7-7](#)

scratch preferences, [7-8](#)

library, managing, [7-6](#)

mount, [12-40](#)

restrictions for SCSI-attached LSMs, [7-7](#)

scratch preferences, [7-7](#)

mount \* command, [12-38](#)

mount command, [12-36](#)

mounting scratch volumes, [8-19](#)

move, [12-42](#)

move command, [12-42](#)

move volume, [12-42](#)

moving.sh, [11-28](#)

multi-homed host, [7-11](#)

---

## N

network IP addresses, registering for primary and  
secondary LANs, [7-10](#)

---

## O

optimization guidelines

cartridges, [C-10](#)

## P

---

- panel, row, column addressing, [D-5](#)
- Partitioning, [H-1](#)
  - changing partition IDs, [H-4](#)
  - changing partitioned ACS to un-partitioned, [H-5](#)
  - guidelines, [H-2](#)
  - library, [H-4](#)
  - re-partitioning a library, [H-4](#)
- pass-thru, [K-1](#)
- pool
  - define, [12-16](#)
  - delete, [12-18](#)
  - display, [13-18](#)
  - query, [12-61](#)
- port verification, [13-20](#)
- ports
  - adding, [11-16](#)
  - information about, [13-20](#)

## Q

---

- query acs command, [12-45](#)
- query cap command, [12-47](#)
- query clean command, [12-49](#)
- query commands, [12-44](#)
- query drive command, [12-50](#)
- query lmu command, [12-51](#)
- query lock command, [12-54](#)
- query lsm command, [12-56](#)
- query mount \* command, [12-59](#)
- query mount command, [12-57](#)
- query pool command, [12-61](#)
- query port command, [12-63](#)
- query request command, [12-64](#)
- query scratch command, [12-65](#)
- query server command, [12-66](#)
- query volume command, [12-67](#)
- Queue Mounts and Dismounts, [7-14](#)

## R

---

- rdb.acsss, [11-36](#)
- rdb.acsss utility, [11-36](#)
- rebalancing scratch pools, [8-17](#)
- reconfiguring ACSLS
  - logging options, [5-6](#), [5-16](#), [5-17](#)
- Reconfiguring ACSLS when tape drives are moved, added, or removed, [7-14](#)
- recovery
  - from failed server, [9-14](#)
- redundant electronics, [B-9](#)
- Redundant Electronics, [B-9](#)
  - in query lmu, [12-51](#)

- Redundant Electronics (*continued*)
  - in switch lmu, [12-81](#)
- registering for event notification, [7-12](#)
- request, query, [12-64](#)
- restore, [9-12](#)
  - miscellaneous ACSLS files, [9-14](#)
- restoring database
  - to most recent backup, [9-13](#)
- restoring miscellaneous ACSLS files, [9-14](#)
- Retry Mounts and Dismounts, [7-14](#)

## S

---

- scratch pool
  - defining, [12-16](#)
  - deleting, [12-18](#)
  - displaying, [13-18](#)
  - mount scratch, query, [12-57](#)
  - rebalancing, [8-17](#)
- scratch, query, [12-65](#)
- SCSI-attached
  - adding, [7-4](#)
- server
  - failure recovery, [9-14](#)
  - idle, [12-31](#)
  - starting, [12-80](#)
- server state, query, [12-66](#)
- server utilities, general information, [11-1](#)
- set cap mode command, [12-69](#)
- set cap priority command, [12-71](#)
- set clean command, [12-73](#)
- set lock command, [12-74](#)
- set owner command, [12-76](#)
- set scratch command, [12-76](#)
- setting event notification, [5-18](#)
- show command, [12-78](#)
- SL3000
  - ACSLs Support, [D-1](#)
  - aliasing CAP IDs, [D-7](#)
  - auditing, [D-3](#)
  - auditing new cartridge addresses, [D-11](#)
  - before configuring, [D-2](#)
  - CAP behavior, [D-9](#)
  - CAP numbering, [D-4](#)
  - connecting to ACSLS, [D-2](#)
  - finding missing cartridges, [D-9](#)
  - internal addressing, [D-5](#)
  - modules, [D-4](#)
  - new panel types, [D-5](#)
  - using dynamic configuration, [D-11](#)
  - varying offline, [D-10](#)
- SL4000
  - ACSLs Support, [F-1](#)
  - AEM bulk operations, [F-9](#)
  - auditing new cartridge addresses, [F-8](#)

SL4000 (*continued*)

- before configuring, [F-2](#)
- CAP numbering, [F-5](#)
- changing username and password, [F-3](#)
- finding missing cartridges, [F-7](#)
- modules, [F-3](#)
- panel types, [F-4](#)
- using dynamic configuration, [F-7](#)
- varying offline, [F-6](#)

SL500

- ACSLS Connectivity, [E-1](#)
- address schemes, [E-2](#)
- auditing the library, [E-5](#)
- CAP behavior, [E-4](#)
- column numbers, [E-2](#)
- drive addresses, [E-2](#)
- library configurable settings and ACSLS, [E-3](#)
- Library Differences, [E-1](#)
- panels, [E-2](#)
- row numbers, [E-2](#)

SL8500

- adding new SL8500 to right, [C-21](#)
- adding new SL8500 to the left, [C-21](#)
- CAP behavior, [D-9](#)
- caps, [C-6](#)
- connecting to multiple SL8500s, [B-8](#), [C-2](#)
- Dual TCP/IP, [B-2](#), [D-12](#)
- expansion, [C-15](#)
- Expansion
  - auditing, [C-17](#)
- internal addresses and ACSLS addresses, [C-3](#)
- managing cartridge locations, [C-11](#)
- minimizing elevator and PTP activity, [C-10](#)
- missing cartridges, [C-12](#), [D-9](#), [F-7](#)
- partitioning, [H-1](#)
- pass-rhtu-ports
  - adding new SL8500s, [C-21](#)
- pass-thru ports, [C-19](#)
- pass-thru-ports
  - merging ACSs, [C-24](#)
  - removing PTPs, [C-26](#)
  - splitting ACSs, [C-26](#)
- supporting tape workloads, [C-11](#)
- tape drive locations, [C-5](#)
- translating internal addresses, [C-4](#)
- vary offline, [C-12](#)
- verify SL8500 components are operational, [C-2](#), [D-2](#)

Solaris

- backup and export database, [9-2](#)

start command, [12-80](#)

static variables

- setting CSI, [5-3](#)
- setting event logging, [5-6](#)

static variables (*continued*)

- setting event notification, [5-18](#)
- setting general product behavior, [5-8](#)

stats\_report utility, [11-40](#)

support

- get\_diags, [I-8](#)

switch lmu command, [12-81](#)

## T

---

TCP/IP timeout, [7-10](#)

translating addresses, [C-4](#), [D-5](#)

transports

- cleaning, [8-10](#)
- cleaning for SCSI attached LSMs, [8-14](#)
- cleaning manually, [8-13](#)
- displaying, [13-8](#)

troubleshooting

- ACSLS event log, [I-1](#)
- client connections, [I-7](#)
- database recovery, [9-12](#)
- drives not being cleaned, [8-14](#)
- GUI, [I-11](#)
- library connections, [I-5](#)
- pinglmu, [I-5](#)

tuning variables, [5-3](#)

## U

---

UNIX file backup, [9-12](#)

unlock command, [12-83](#), [12-85](#)

unscratching volumes, [8-20](#)

user, show, [12-78](#)

userAdmin.sh, [11-42](#)

using a cmd\_proc, [1-2](#)

Using display Command Options, [13-2](#)

using the CAP, [8-2](#)

utilities

- acs\_renumber.sh, [11-3](#), [11-4](#)
- acsss, [11-7](#)
- bdb.acsss, [11-8](#)
- config acs, [11-11](#)
- config drives, [11-13](#)
- config lsm, [11-14](#)
- config ports, [11-15](#)
- db\_export.sh, [11-16](#)
- db\_import.sh, [11-17](#), [11-18](#)
- del\_vol, [11-19](#)
- drives\_media.sh, [11-21](#)
- ejecting.sh, [11-21](#)
- free\_cells.sh, [11-24](#)
- getHba.sh, [11-25](#)
- greplog, [11-27](#)
- lib\_type.sh, [11-28](#)
- moving.sh, [11-28](#)

utilities (*continued*)

pinglmu, [1-6](#)  
probeFibre.sh, [11-36](#), [1-6](#)  
probescsi.sh, [1-6](#)  
rdb.acsss, [11-36](#)  
showDevs, [1-6](#)  
showDrives.sh, [11-40](#)  
stats\_report, [11-40](#)  
testlmu, [1-6](#)  
testlmutcp, [1-5](#)  
testports, [1-4](#), [1-5](#)  
userAdmin.sh, [11-42](#)  
volrpt, [11-43](#)  
watch\_vols, [11-48](#)

---

**V**

vary command, [12-85](#)  
verifying imported database, [9-10](#)  
volrpt utility, [11-43](#)

volume attributes, [9-5](#)  
volume reports, custom, [10-1](#)  
volume statistics, [5-6](#)  
volume statistics report  
    logging volume movement, [10-3](#)  
volume, querying, [12-67](#)  
volumes  
    absent, [8-20](#), [8-21](#)  
    adding scratch, [8-17](#)  
    delete, [8-23](#)  
    display, [13-21](#)  
    displaying scratch information, [8-17](#)  
    ejecting, [8-7](#)  
    managing scratch, [8-16](#)  
    mounting scratch, [8-19](#)  
    unscratching, [8-20](#)

---

**W**

watch\_vols utility, [11-48](#)