

StorageTek Automated Cartridge System Library Software

管理者ガイド

リリース 8.4

E68227-01

2015 年 10 月

StorageTek Automated Cartridge System Library Software

管理者ガイド

E68227-01

Copyright © 2015, Oracle and/or its affiliates. All rights reserved.

このソフトウェアおよび関連ドキュメントの使用と開示は、ライセンス契約の制約条件に従うものとし、知的財産に関する法律により保護されています。ライセンス契約で明示的に許諾されている場合もしくは法律によって認められている場合を除き、形式、手段に関係なく、いかなる部分も使用、複写、複製、翻訳、放送、修正、ライセンス供与、送信、配布、発表、実行、公開または表示することはできません。このソフトウェアのリバース・エンジニアリング、逆アセンブル、逆コンパイルは互換性のために法律によって規定されている場合を除き、禁止されています。

ここに記載された情報は予告なしに変更される場合があります。また、誤りが無いことの保証はいたしかねます。誤りを見つけた場合は、オラクルまでご連絡ください。

このソフトウェアまたは関連ドキュメントを、米国政府機関もしくは米国政府機関に代わってこのソフトウェアまたは関連ドキュメントをライセンスされた者に提供する場合は、次の通知が適用されます。

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

このソフトウェアまたはハードウェアは様々な情報管理アプリケーションでの一般的な使用のために開発されたものです。このソフトウェアまたはハードウェアは、危険が伴うアプリケーション (人的傷害を発生させる可能性があるアプリケーションを含む) への用途を目的として開発されていません。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用する際、安全に使用するために、適切な安全装置、バックアップ、冗長性 (redundancy)、その他の対策を講じることは使用者の責任となります。このソフトウェアまたはハードウェアを危険が伴うアプリケーションで使用したこと起因して損害が発生しても、Oracle Corporation およびその関連会社は一切の責任を負いかねます。

Oracle および Java はオラクルおよびその関連会社の登録商標です。その他の社名、商品名等は各社の商標または登録商標である場合があります。

Intel, Intel Xeon は、Intel Corporation の商標または登録商標です。すべての SPARC の商標はライセンスをもとに使用し、SPARC International, Inc. の商標または登録商標です。AMD, Opteron, AMD ロゴ、AMD Opteron ロゴは、Advanced Micro Devices, Inc. の商標または登録商標です。UNIX は、The Open Group の登録商標です。

このソフトウェアまたはハードウェア、そしてドキュメントは、第三者のコンテンツ、製品、サービスへのアクセス、あるいはそれらに関する情報を提供することがあります。適用されるお客様と Oracle Corporation との間の契約に別段の定めがある場合を除いて、Oracle Corporation およびその関連会社は、第三者のコンテンツ、製品、サービスに関して一切の責任を負わず、いかなる保証もいたしません。適用されるお客様と Oracle Corporation との間の契約に定めがある場合を除いて、Oracle Corporation およびその関連会社は、第三者のコンテンツ、製品、サービスへのアクセスまたは使用によって損失、費用、あるいは損害が発生しても一切の責任を負いかねます。

目次

はじめに	41
ドキュメントのアクセシビリティについて	41
1. 概要	43
ACSLS について	43
ACSLS HA について	43
acssa および acsss ユーザー ID	43
acsss マクロ	44
cmd_proc の使用	44
cmd_proc ウィンドウ	44
cmd_proc の使用: 行モードと比較した場合のカーソルモード	44
カーソルモードで保持される制限された履歴	45
行モードでのコマンドと混在したステータスメッセージ	45
カーソルモードの cmd_proc では 80 文字より長い行を表示できません	46
cmd_proc を中断および再開する方法	46
cmd_proc の終了	46
cmd_proc の起動	47
リモートログイン	47
cmd_proc のキーボードショートカット	47
cmd_proc の入力および出力のリダイレクト	48
追加の cmd_proc ウィンドウでの入力ファイルの指定	48
追加の cmd_proc ウィンドウへの出力のリダイレクト:	49
アイドル状態への ACSLS の移行	49
ACSLS の再起動	50
ACSLS のディレクトリ構造	50
2. ACSLS の起動およびモニタリング	53
ACSLS の起動	53

ACSLS のモニタリング	54
ACSLS の停止	54
Solaris での SMF タイムアウト	54
ACSLS の起動ポリシー	55
追加の起動時間	55
ACS の目的の (オフライン) 起動状態	55
3. ACSLS ライセンスキー	57
4. ACSLS GUI	59
ACSLS GUI の起動	60
GUI のユーザーとパスワードの管理	60
GUI の概要	61
GUI の「Welcome」画面	61
マストヘッド	61
アラームのカラーコード	62
System Preferences	63
ナビゲーションツリー	64
Overview and Status	64
Configuration and Administration	65
Log Viewer	65
System Events	65
Logical Library Configuration	65
Tape Library Operations	66
Tape Libraries & Drives	66
ACSLS 8.4 GUI で除外されたシステム操作	67
ファイアウォールと GUI	67
Solaris の場合	67
Linux の場合	68
HTTPS 用の自己割り当てデジタル証明書構成	69
5. ライブラリハードウェアの設置と構成	71
ライブラリハードウェアへの接続のインストール	71

SCSI mchanger デバイスドライバの追加	71
SL8500 または SL3000 デュアル TCP/IP サポートの有効化	72
ライブラリハードウェアの構成または再構成	72
acsss_config を使用したライブラリハードウェアの構成	73
ACS の番号の変更	81
6. ACSLS の動作を制御する変数の設定	83
ACSLs 機能構成メニューへのアクセス	84
動的変数および静的変数の確認および変更	86
CSI チューニング変数の設定	86
イベントロギング変数の設定	90
一般的な製品動作変数の設定	94
アクセス制御変数の設定	104
自動バックアップ変数の設定	106
アクセス制御情報の再作成	107
イベント通知設定の定義	107
ライブラリハードウェアの構成または更新	108
システムイベントの電子メール通知の登録	109
7. アクセス制御	111
ボリュームアクセス制御	111
ボリュームアクセス制御の有効化	111
ユーザーのボリュームへのアクセスが許可されるほかのユーザーの定義	113
ボリュームの所有権の確立	116
所有権ポリシー	116
所有権の確認	117
ボリュームアクセスのサマリー	117
コマンドアクセス制御	118
コマンドアクセス制御の有効化	119
クライアント ID のユーザー名への関連付け	119
どのユーザーがどのようなコマンドを使用できるかの定義	120
コマンドアクセス制御の許可および禁止ファイルのコマンド名	120

アクセス制御メッセージのロギング	122
8. ライブラリ管理	123
ACS 番号の指定	123
テープドライブの移動、追加、または取り外し時の ACSLS の再構成	124
ACSLS に構成されているテープドライブを更新する	124
ライブラリの監査	125
監査のしくみ	126
監査を実行するタイミング	126
監査間隔	127
ACSLS への SCSI またはファイバ接続ライブラリの追加	128
拡張ストア機能の使用	128
混合メディアライブラリの管理	130
ACSLS の混合メディア設定を表示する	131
SCSI 接続 LSM の混合メディアの制限事項	131
スクラッチ優先順位の設定	132
ユーザー定義の混合メディアファイル	132
システム定義の混合メディアファイル	133
ACSLS が混合メディアファイルを使用するしくみ	133
スクラッチ優先順位ファイルを定義する	134
デュアル TCP/IP 接続の管理	135
デュアル LMU 構成の管理	135
デュアル LAN クライアント構成の管理	137
プライマリおよびセカンダリ LAN の IP アドレスを登録する	138
マルチホーム ACSLS サーバー用に 2 つ目の Ethernet ポートをインストールする	138
TCP/IP 接続のタイムアウト間隔を設定する	139
イベント通知への登録	140
所定の状態への自動回復	140
現在の状態および所定の状態	141
現在の状態は下位へカスケードされ、所定の状態はそうならない	141
ライブラリが一時的に使用不可になっている場合のマウントおよびマウント解除のキューへの追加と再試行	142

テープドライブの移動、追加、または取り外し時の ACSLS の再構成	143
ACSLS に構成されているテープドライブを更新する	144
メディア検証	145
メディア検証プールにドライブを追加する	145
メディア検証プールからドライブを削除する	145
9. カートリッジの管理	147
LSM の設定	148
CAP の使用	148
CAP のタイプ	148
CAP の状態	150
CAP のモード	150
CAP の優先順位	152
CAP 情報の表示	153
カートリッジの挿入	153
挿入リクエストの終了	154
カートリッジの取り出し	155
CAP の回復	156
一般的な CAP の回復手順	156
CAP の回復を行う前に挿入と取り出しを完了する	156
ハングした CAP を強制的にオフラインにしてからオンラインに変更す ることで回復する	156
アクセスドアを開いたあとの CAP の回復	157
自動挿入に使用する CAP のロックが解除されない	157
手動挿入に使用する CAP のロックが解除されない	157
取り出しに使用する CAP のロックが解除されない	158
L1400、L700、L700e、または L180 ライブラリ内の CAP のロックを解除する ための回復手順	159
手動挿入に使用する CAP のロックが解除されない	159
自動挿入に使用する CAP のロックが解除されない	159
取り出しに使用する CAP のロックが解除されないため、空にできな い	159
新しい再アクティブ化されたカートリッジへのポリシーの自動適用	160

クリーニングカートリッジ属性の自動割り当て	160
watch_vols ポリシー	160
クリーニングカートリッジ	161
ACSLS による自動クリーニング	161
クリーニングカートリッジの最大使用回数	162
クリーニングカートリッジの手動の定義	163
クリーニングカートリッジのモニタリング	163
クリーニングカートリッジの挿入	164
使用済みのクリーニングカートリッジの取り出し	165
ドライブの手動クリーニング	165
ファイバ接続ライブラリ上のカートリッジのクリーニング	166
ドライブがクリーニングされていないときの対処	167
自動クリーニングが有効になっていることを確認する	167
ドライブ用のクリーニングカートリッジが存在していることを確認する	167
SL8500 または SL3000 ライブラリの SL コンソールを使用して自動ク リーニングが無効になっていることを確認する	168
クリーニングカートリッジに questionable とマークされているかどうかを チェックする	169
スクラッチカートリッジの管理	169
スクラッチプールおよびスクラッチカートリッジ情報の表示	170
ライブラリへのスクラッチカートリッジの追加	171
スクラッチプールのリバランス	171
スクラッチプールの削除	172
スクラッチプールを空にする	172
単一のプールの削除	172
すべての空のプールの削除	173
スクラッチカートリッジのマウント	173
単一メディア環境	173
混合メディア環境	174
カートリッジのスクラッチ解除	174
不在カートリッジと取り出し済みカートリッジのサポートの使用	175
不在、取り出し済み、および見つからないカートリッジ	176
カートリッジ (ボリューム) のステータスレポート	176

カートリッジの回復	177
見つからないカートリッジ	178
不在カートリッジと取り出し済みカートリッジ	179
カートリッジが見つからない	179
カートリッジが見つかった	179
手動ボリューム削除ユーティリティーの使用	180
期限切れカートリッジの特定	180
カートリッジの寿命のパーセンテージ	181
アクセス数	182
ACSL5 のカートリッジマウント回数の詳細	182
カートリッジの保証期限と寿命のしきい値	183
アクティブな LSM からの、もっとも長い間アクセスされていないカートリッジの移動	183
空きセルが少なすぎる LSM と空のセルを含む LSM を特定する	184
アクセス日で LSM 内のカートリッジを確認する	185
簡単にソートできるアクセス日が報告されることを確認する	185
LSM 内のカートリッジの最終アクセス日の分布を調べる	185
いっぱいになった LSM から空のセルを含む LSM にカートリッジを移動する	186
移動するカートリッジのリストを作成する	186
空き領域のある LSM にカートリッジを移動する	186
無効になっている LSM のドライブへのカートリッジの手動ロード	187
10. データベース管理	189
使用されるユーティリティー	190
データベースのエクスポート	190
以前のリリースへのエクスポートの前のサポートされていないテープライブラリ、ドライブ、およびカートリッジの取り外し	192
Linux にエクスポートする前の論理ライブラリの削除	192
ディスクファイルへのエクスポート	192
テープへのエクスポート	193
データベースのインポート	194
ディスクファイルからのインポート	195

テープからのインポート	198
ファイバ用の mchanger の新しいプラットフォームへの移行	199
古い ACSLS サーバーのファイバ接続ライブラリに関する詳細の記録	199
ファイバ接続ライブラリの mchanger 名を変更するための ACSLS の再構成	200
インポートされたデータベースとライブラリ構成の検証	201
自動データベースバックアップ	202
テープへの手動バックアップの実行	202
ACSLS サーバーに接続された、指定されたテープデバイスへのバックアップ	203
UNIX ファイルへのバックアップ	204
回復と復元	204
データベースの最新のバックアップへの復元	205
障害が発生したサーバーからの回復	206
ACSLS 制御ファイルの復元	206
11. レポートおよびロギング	209
カスタムボリュームレポートの作成	209
カスタムボリュームレポート	210
カスタムボリュームレポートの例	211
ロギングボリューム統計レポートの作成	212
12. ユーティリティーリファレンス	215
概要	216
レガシー起動/停止スクリプト	217
ユーティリティーコマンド	217
acs_renumber.sh	217
acsss マクロ	219
形式	220
オプション	220
bdb.acsss	221
形式	222
オプション	222
使用方法	222

動的構成 (config) ユーティリティ	223
動的構成の制限事項	224
避けるべき操作	225
config acs	225
形式	226
新しい ACS の追加	226
既存の ACS の再構成	227
config acs の制限	228
config drives	228
形式	229
使用方法	229
config lsm	229
形式	230
使用方法	230
config lsm の制限	230
config ports	231
形式	231
使用方法	231
config ports の制限	232
db_export.sh	232
形式	232
オプション	232
使用方法	233
db_import.sh	233
形式	233
オプション	233
使用方法	234
del_vol	234
形式	235
オプション	235
使用方法	236
例	236
drives_media.sh	236

形式	236
オプション	237
ejecting.sh	237
形式	237
オプション	238
ejecting.sh のログ	240
free_cells.sh	241
形式	241
オプション	241
例	241
getHba.sh	244
形式	244
get_license_info	246
greplog	246
形式	246
オプション	246
使用方法	247
install_scsi_Linux.sh	247
形式	247
lib_type.sh	248
形式	248
licensekey.sh	248
moving.sh	248
形式	249
使用方法	249
例	256
probeFibre.sh	258
形式	258
オプション	258
rdb.acsss	259
形式	259
メニューオプション	259
参照:	263

showDevs.sh	264
形式	264
オプション	264
showDrives.sh	264
形式	265
stats_report	265
形式	265
使用方法	266
userAdmin.sh	267
形式	268
volrpt	269
形式	269
オプション	269
使用方法	270
例	272
watch_vols	275
形式	276
使用方法	276
例	276
13. コマンドリファレンス	279
一般的なコマンド構文	281
コンポーネントのタイプと識別子	282
一般的なコマンドエラーメッセージ	285
コマンド	285
audit	285
形式	286
使用方法	287
例	289
コマンド領域メッセージ	290
表示領域メッセージ	292
cancel	292
形式	292

オプション	293
例	295
コマンド領域メッセージ	296
表示領域メッセージ	296
clear lock	297
形式	297
オプション	297
使用方法	297
例	297
コマンド領域メッセージ	298
表示領域メッセージ	299
define pool	299
形式	299
オプション	299
使用方法	300
例	300
コマンド領域メッセージ	300
表示領域メッセージ	301
delete pool	301
形式	301
オプション	301
使用方法	302
例	302
コマンド領域メッセージ	303
dismount	303
形式	303
オプション	303
使用方法	304
例	305
コマンド領域メッセージ	306
表示領域メッセージ	308
eject	308
形式	308

オプション	308
使用方法	310
例	310
コマンド領域メッセージ	311
表示領域メッセージ	313
enter	313
オプション	314
使用方法	315
例	315
コマンド領域メッセージ	316
表示領域メッセージ	318
idle	318
形式	318
オプション	319
使用方法	319
例	319
コマンド領域メッセージ	320
表示領域メッセージ	320
lock	321
形式	321
オプション	321
使用方法	321
例	322
コマンド領域メッセージ	322
表示領域メッセージ	323
logoff	323
形式	324
オプション	324
使用方法	324
例	324
コマンド領域メッセージ	324
表示領域メッセージ	324
mount	324

形式	324
オプション	325
使用方法	325
例	326
コマンド領域メッセージ	326
表示領域メッセージ	328
mount *	328
形式	328
オプション	328
使用方法	329
例	330
コマンド領域メッセージ	331
表示領域メッセージ	333
move	334
形式	334
オプション	334
使用方法	334
例	335
コマンド領域メッセージ	335
query コマンド	336
形式	336
コマンド領域メッセージ	336
表示領域メッセージ	337
query acs	337
形式	337
オプション	337
使用方法	337
例	338
query cap	339
形式	339
オプション	339
使用方法	339
例	341

query clean	341
形式	341
オプション	342
使用方法	342
例	343
query drive	343
形式	343
オプション	343
使用方法	343
例	345
query lmu	345
形式	346
オプション	346
使用方法	346
例	349
query lock	349
形式	349
オプション	349
使用方法	350
例	351
query lsm	351
形式	351
オプション	351
例	353
query mount	353
形式	353
オプション	353
使用方法	354
例	355
query mount *	356
形式	356
オプション	356
使用方法	356

例	358
query pool	358
形式	358
オプション	359
使用方法	359
例	360
query port	360
形式	360
オプション	360
使用方法	360
例	361
query request	361
形式	361
オプション	361
使用方法	362
例	362
query scratch	363
形式	363
オプション	363
使用方法	363
例	364
query server	364
形式	364
オプション	365
使用方法	365
例	366
query volume	366
形式	366
オプション	366
使用方法	366
例	367
set コマンド	368
形式	368

コマンド領域メッセージ	368
表示領域メッセージ	368
set cap mode	369
形式	369
オプション	369
使用方法	369
例	369
コマンド領域メッセージ	370
set cap priority	371
形式	371
オプション	371
使用方法	371
例	372
コマンド領域メッセージ	372
表示領域メッセージ	373
set clean	373
形式	373
オプション	373
使用方法	373
例	374
コマンド領域メッセージ	374
表示領域メッセージ	375
set lock	375
形式	375
オプション	375
使用方法	375
例	376
コマンド領域メッセージ	376
表示領域メッセージ	376
set owner	377
形式	377
オプション	377
使用方法	377

例	377
コマンド領域メッセージ	377
表示領域メッセージ	377
set scratch	378
形式	378
オプション	378
使用方法	378
例	378
コマンド領域メッセージ	379
表示領域メッセージ	380
show	380
形式	380
オプション	380
使用方法	380
例	381
コマンド領域メッセージ	381
表示領域メッセージ	381
start	382
形式	382
オプション	382
使用方法	382
例	382
コマンド領域メッセージ	382
表示領域メッセージ	383
switch lmu	383
形式	383
オプション	383
使用方法	384
例	384
コマンド領域メッセージ	384
表示領域メッセージ	386
unlock	386
形式	386

オプション	386
使用方法	386
例	387
コマンド領域メッセージ	387
表示領域メッセージ	388
vary	388
形式	388
オプション	388
使用方法	389
例	391
コマンド領域メッセージ	392
表示領域メッセージ	395
venter	395
形式	396
オプション	396
使用方法	396
例	397
コマンド領域メッセージ	397
表示領域メッセージ	398
14. display コマンドリファレンス	399
display コマンドオプションの使用	400
ワイルドカード文字の使用	400
例	401
範囲の使用	401
例	402
形式	402
オプション	402
80 文字より長い行を表示するときに行モードで cmd_proc を起動する	403
コマンド	403
display cap	403
形式	404
フィールド	404

オプション	404
例	407
display cell	407
形式	407
フィールド	407
オプション	408
例	409
display drive	409
形式	409
フィールド	409
オプション	409
例	414
display lock	415
形式	415
フィールド	415
オプション	415
例	416
display lsm	416
形式	417
フィールド	417
例	420
display panel	420
形式	420
フィールド	420
オプション	421
例	422
display pool	422
形式	422
フィールド	422
オプション	422
例	424
display port	424
形式	424

フィールド	424
オプション	424
例	426
display volume	426
形式	427
フィールド	427
オプション	427
例	432
15. lib_cmd	435
はじめに	435
コマンド	435
パス名	435
概要	436
lib_cmd の使用	436
オプション	437
サブコマンド	437
バッチモードでの lib_cmd の使用	445
A. ACSLS のバックアップおよび回復ツール	447
ACSLS バックアップツール	447
自動バックアップ	447
手動バックアップ	448
手動でのデータベースのエクスポート	448
ACSLS 回復ツール	449
rdb.acsss の使用	450
db_import.sh の使用	450
障害のシナリオ	451
データベースが破損した	451
間違ったライブラリに対して acsss_config を実行した	452
サーバー障害 – 新しいハードウェアでの同じサーバーの再構築	452
サーバー障害 – 新しいハードウェアでの別の ACSLS サーバーの再構築	452

B. エンタープライズライブラリの接続オプション	453
概要	453
ACSLS とライブラリの通信ステータスの表示	454
デュアル TCP/IP サポート	454
要件	455
構成	455
シナリオ 1 - 推奨される構成	456
シナリオ 2	457
シナリオ 3	458
シナリオ 4	458
カスタマイズされたルーティングテーブルのエントリをリブート後も保持する	460
スクリプトの作成	460
ブート時に初期化されるカスタムルートの追加	461
ルーティングコマンドの削除	461
マルチ TCP/IP サポート	462
冗長電子装置	463
ACSLS での RE のサポート	464
マウントおよびマウント解除の照会と再試行	465
単一ライブラリのための switch lmu	465
C. SL8500 の ACSLS サポート	467
複数の SL8500 へのマルチ TCP/IP を使用した接続	468
SL8500 のすべてのコンポーネントが動作していることの確認	468
SL8500 内部アドレスと ACSLS アドレスについて	469
SL コンソールを使用したアドレス変換	471
テープドライブの場所	472
パーティションからセルを取り外す前のカートリッジの移動	473
SL8500 CAP	473
一括 CAP	474
ACSLS で一括 CAP を処理するための SL8500 のアップグレード	475
挿入と取り出しの目的を示すカスタム SL コンソールメッセージ	477

回転式 CAP	478
挿入または取り出し操作	479
一部の ACSLS クライアントでの挿入、取り出し、および監査の操作	479
エレベータおよび PTP アクティビティを最小限に抑える	480
ワークロードをサポートするためのテープドライブの構成	480
カートリッジの場所の管理	481
見つからないカートリッジを見つける	482
SL8500 のオフラインへの変更	482
SL コンソールではなく ACSLS を使用して SL8500 コンポーネントをオフラインに変更する	483
SL8500 コンポーネントを ACSLS に対してオフラインに変更する場合	483
アクセスドアを開く前	483
CAP が動作していない場合	483
サービス安全ドアを閉じたとき	484
サービス安全ドアを使用するときこれらの ACSLS コマンドおよびユーティリティを使用しない場合	484
動的構成 (config) ユーティリティの使用	485
SL8500 の拡張	486
内容	486
SL8500 モジュールの図:	488
拡張 SL8500 の監査	489
パススルーポートを使用した SL8500 の接続	491
SL8500 PTP 接続を設置する前に	492
新しい SL8500 の追加	493
左側への新しい SL8500 の追加	493
新しい ACSLS 構成の動的な構成	494
右側への新しい SL8500 の追加	494
右側に SL8500 を追加するときの考慮事項	494
新しい ACSLS 構成の動的な構成	494
ACS のマージの手順	497
右から左に番号が付けられた ACS のマージ	497
右から左に番号が付けられた ACS をマージする手順	497
左から右に番号が付けられた ACS のマージ	498

左から右に番号が付けられた ACS をマージする手順	498
PTP の取り外しと ACS の分割	499
左側に SL8500 から新しい ACS を作成して ACS を分割する - 考えられるシナリオ	500
ACS を分割するための ACSLS の手順	500
分割の右側への新しい ACS の追加	501
D. SL3000 の ACSLS サポート	503
ACSLS サポート	503
SL3000 向けに ASSLS を構成する前に	504
ACSLS への SL3000 の接続	504
すべての SL3000 コンポーネントが動作していることの確認	505
ACSLS に対する SL3000 の構成	505
SL3000 の監査	505
SL3000 アドレス指定	506
CAP 番号付け	506
SL3000 モジュール	507
新しいパネルタイプ	508
内部 SL3000 アドレス指定について	508
AEM の使用	509
アクセスドア	509
安全ドア	509
CAP の操作	510
中断不要のメンテナンス	511
SL3000 CAP ID の別名処理	511
パーティションからセルを取り外す前のカートリッジの移動	513
SL3000 CAP の動作	513
見つからない SL3000 カートリッジを見つける	513
SL3000 のオフラインへの変更	514
SL コンソールではなく ACSLS を使用して SL3000 コンポーネントをオフラインに変更する	514
SL3000 コンポーネントを ACSLS に対してオフラインに変更する場合	514
アクセスドアを開く前	514

CAP が動作していない場合	514
動的構成 (config) ユーティリティの使用	515
新しいカートリッジアドレスの監査	515
ACSLS のデュアル TCP/IP	516
E. SL500 の ACSLS サポート	517
ACSLS 接続	517
ACSLS と SL500 ライブラリの相違点	517
ライブラリの構成	517
ライブラリの場所の特定	518
アドレス指定スキーム	518
パネル	518
行番号	518
列番号	518
ドライブのアドレス	519
ACSLS の制限事項	519
SL500 ライブラリ設定の構成	520
SL500 CAP の動作	520
ライブラリの監査	521
これが新しいライブラリである場合	521
モジュールの追加、取り外し、またはスワップ後	521
ドアを通じた手動によるカートリッジの追加または取り外し後	522
ライブラリ構成設定の変更後	522
F. SL150 の ACSLS サポート	523
ACSLS 接続	523
ACSLS と SL150 ライブラリの相違点	523
ライブラリの構成	523
ライブラリの場所の特定	524
アドレス指定スキーム	524
パネル	524
行番号	524

列番号	524
ドライブのアドレス	525
ACSLS の制限事項	525
SL150 ライブラリ設定の構成	526
SL150 CAP の動作	526
取り出し中に開かない CAP (メールスロット)	527
ライブラリの監査	528
これが新しいライブラリである場合	529
モジュールの追加、取り外し、またはスワップ後	529
メールスロットを通じた手動によるカートリッジの追加または取り外し後	529
ライブラリ構成設定の変更後	529
G. StorageTek Virtual Tape Library の ACSLS のサポート	531
サポートされている構成	531
VTL の動作	532
ACSLS に対する VTL の構成	533
前提条件	533
インストール	533
H. 論理ライブラリのサポート	537
論理ライブラリについて	537
利点	537
制限	539
論理ライブラリの作成	539
物理 ACS の指定	539
論理ライブラリの属性の指定	540
1 つ以上の物理ドライブの割り当て	541
1 つ以上の物理ボリュームの割り当て	541
クライアント接続の指定	543
高速ロード	544
論理ライブラリの削除	544
トラブルシューティング	545

クライアントが割り当て済みの論理ライブラリを表示できない場合はどうしたらよいですか。	545
クライアントは正しく接続されているが、引き続き論理ライブラリが表示されない場合はどうしたらよいですか。	545
クライアントでの論理ライブラリの構成で問題が発生している場合はどうしたらよいですか。	546
FC ポートをターゲットモードで構成する方法	546
I. ライブラリのパーティション分割	549
SL8500 と SL3000 の両方のパーティション分割に共通した点	550
パーティション分割のガイドライン	551
新しい構成を計画する	551
SL3000 または SL8500 のパーティション分割時の混乱を最小限に抑える	552
構成における変更	552
停止を最小限に抑える	552
問題	552
ライブラリのパーティション分割またはパーティション ID の変更	553
ライブラリの再パーティション分割	554
パーティション分割されている ACS をパーティション分割されていない ACS に変更する	555
ACS パーティション ID の表示	556
CAP の動作	556
パーティション分割されたライブラリ	556
パーティション分割されたライブラリでの CAP の専用化	557
パーティション分割されたライブラリでの CAP の共有	557
別のホストが CAP を必要とする場合	557
共有された CAP の CAP 優先順位の割り当て	558
CAP の予約	558
CAP 予約の終了	558
特定のセルへのカートリッジの移動	559
J. トラブルシューティング	561
ACSLS イベントログ	561

イベントログの管理	562
<i>greplog</i> を使用したイベントログの検索	562
形式	562
オプション	562
例	563
追加のログ	563
トレースログの管理	564
Java コンポーネントのログ	565
鍵監視ポイント	565
ACSLS の起動に関する問題の診断	566
ライブラリの接続テスト	566
testports	566
testlmutcp	567
testlmu	567
pinglmu.sh	567
probescsi.sh	568
probeFibre.sh	568
showDevs.sh	568
クライアントの接続テスト	569
ブリッジドライブで接続したファイバライブラリ内の CAP がロックされる	569
Oracle Support 用の診断情報の収集	570
ACSLS と SELinux (Security-Enhanced Linux)	570
ACSLS の SELinux ポリシーモジュールのアンインストール	570
SELinux の強制を管理する	571
GUI が動作していることの検証	573
GUI のトラブルシューティングのヒント	574
K. ACSLS クライアントアプリケーションのインストール時の問題	577
Solaris 11 での ACSAPI クライアントのインストール	577
L. ライブラリのパフォーマンス	581
十分な並行マウントおよびマウント解除リクエストの送信	581

複数の SL8500 への接続	582
LSM 間のパススルーアクティビティの最小化	583
マウント解除中のカートリッジのフロート	583
LSM 内の空のストレージセルの保持	584
マウントおよびマウント解除のタイムアウト	584
ライブラリへのカートリッジの挿入	585
ライブラリからのカートリッジの取り出し	585
ACSAPI リクエストと ACSLS コマンドを使用したパススルーアクティビティの 最小化	585
特定のカートリッジのマウント	585
query mount	586
mount	586
ACSLS によって選択されたスクラッチカートリッジのマウント	586
query mount scratch (query mount * と呼ばれます)	586
mount scratch (query mount * と呼ばれます)	587
これらの ACSAPI リクエストおよび ACSLS コマンドの使用	588
M. ファイアウォールセキュリティオプション	591
ファイアウォールの背後での ACSLS の実行	591
セキュリティ領域への対処	592
RPC	592
セキュリティ	592
通信コンポーネント	592
ファイアウォールセキュリティオプションの利点	593
ACSLS サーバー側	593
ACSLS サーバーポートの制限	593
クライアント側 (CSC)	594
クライアントポートの制限	594
メリット	595
ファイアウォールセキュリティ機能をオンにして変数を設定する	595
ACSLS 変数	595
ACSLS 変数の表示および設定	598
ACSAPI クライアントシステム変数	598

CSC ツールキット 2.3 での新しい変数	599
クライアント上の環境変数の表示および設定	600
ファイアウォールセキュリティーソリューションのシナリオ	601
ACSLS サーバー側のみでのファイアウォールセキュリティー	601
クライアント側のみでのファイアウォールセキュリティー	603
ポートマッパーを使用した ACSLS サーバー側とクライアント側の両方での ファイアウォールセキュリティー	604
ポートマッパーを使用しない ACSLS サーバー側とクライアント側の両方で のファイアウォールセキュリティー	606
ACSLS サーバーでファイアウォールセキュリティーをオンにする	607
ACSLS サーバーでファイアウォールセキュリティーをオフにする	610
ファイアウォールセキュリティーの構成	611
ファイアウォールセキュリティー通信のトラブルシューティング	612
よくある質問	614
N. CSCI	621
CSCI の概要	621
CSCI 接続	621
アーキテクチャーおよびサブシステム	621
エラーメッセージ	622
一般的なエラーメッセージ	622
環境変数	623
O. メディア管理	625
概要	625
制限事項	625
例	625
解決方法	627
ACSLS によって報告される記録密度を使用した一般的なメディアの管理	627
手順	628
ACSLS プールを使用した、異なる密度で記録された一般的なメディアの管理	630
手順	630

P. XAPI クライアントインタフェース	635
ACSLS サーバーへの XAPI クライアントインタフェース	635
ACSLS XAPI サービス	635
XAPI の変数	636
Q. ACSLS でのアクセシビリティ機能	641
ユーザーインタフェース	641
ACSLS (テキストのみ) コマンド行インタフェース (CLI)	641
プライマリ管理インタフェース: Unix シェル機能	642
ACSLS グラフィカルユーザーインタフェース (GUI)	642
ACSLS GUI でのアクセシビリティのプロビジョニング	643
ACSLS でアクセシビリティモードを構成するために必要な特殊な手順	643
GUI ツリーメニュー	643
用語集	645
索引	655

図の一覧

4.1. ACSLS グラフィカルユーザーインターフェース	61
4.2. ACSLS の「Overview and Status」ページ	64
B.1. 推奨される構成	456
B.2. ACSLS と SL8500 または SL3000 でパブリックネットワークを使用	457
B.3. 2 つのネットワークインターフェースを備えた SL8500 または SL3000	458
B.4. ACSLS HA	459
B.5. マルチ TCP/IP を使用した ACSLS	463
B.6. マルチ TCP/IP およびデュアル TCP/IP を使用した ACSLS	463
B.7. RE を使用した ACSLS	464
B.8. RE とマルチ TCP/IP を使用した ACSLS	464
B.9. RE とデュアル TCP/IP を使用した ACSLS	465
B.10. RE とデュアル TCP/IP およびマルチ TCP/IP	465
C.1. SL8500 用に ACSLS を構成する前の ACSLS サーバーを伴う SL8500	467
C.2. ソフトウェアおよび物理ドライブの番号付け	473
C.3. SL8500 顧客拡張モジュール	488
C.4. 4 つの接続された SL8500 ライブラリ	492
C.5. 4 つの接続された SL8500 ライブラリ	493
C.6. マージ対象の ACS	497
C.7. 目的の構成: 単一の ACS	497
C.8. 既存の構成: 3 つの個別の ACS	498
C.9. 目的の構成: 単一の ACS	498
C.10. 既存の構成: 1 つの ACS	500
C.11. 目的の構成: 2 つの ACS	500
D.1. SL3000 アドレス指定	506
L.1. LSM 間のパススルーアクティビティの最小化	583
M.1. ACSLS サーバー側のみでのファイアウォールセキュリティ	601
M.2. クライアント側のみでのファイアウォールセキュリティ	603
M.3. ポートマッパーを使用した ACSLS サーバー側とクライアント側の両方でのファイ アウォールセキュリティ	605
M.4. ポートマッパーを使用しない ACSLS サーバー側とクライアント側の両方でのファイ アウォールセキュリティ	606

N.1. CSCI アーキテクチャーおよびサブシステム	622
N.2. トークンリングインタフェースを使用した CSCI システム	622

表の一覧

1.1. cmd_proc のキーボードショートカット	48
1.2. ACSLS のディレクトリ構造	50
7.1. ボリュームアクセスが有効	118
7.2. ボリュームアクセスが有効	118
7.3. コマンドアクセスが有効	121
7.4. コマンドアクセスが有効	121
8.1. 監査のスコープが監査間隔に与える影響	127
8.2. サポートされている LSM の平均監査時間	127
8.3. ACSLS が混合メディアファイルを使用するしくみ	133
9.1. CAP のタイプ	148
9.2. CAP の状態	150
9.3. CAP のモード	151
9.4. CAP の優先順位	152
9.5. カートリッジの挿入コマンド	154
9.6. 寿命のしきい値	183
13.1. ACSLS コンポーネントのタイプと識別子	282
13.2. audit に有効なコンポーネント	286
13.3. clear lock に有効なコンポーネント	297
13.4. lock に有効なコンポーネント	321
13.5. query lock に有効なロックのタイプ	350
13.6. unlock に有効なコンポーネント	386
13.7. vary に有効なコンポーネント	388
13.8. vary offline の結果	389
13.9. vary offline force の結果	390
13.10. vary online の結果	391
14.1. 手動介入を待機するドライブとライブラリ	414
C.1. アドレス指定の説明	469
E.1. ドライブのアドレス指定の例	519
F.1. ドライブのアドレス指定の例	525
J.1. GUI のトラブルシューティングのヒント	574
N.1. CSCI サーバーサブシステムの環境変数	623

例の一覧

B.1. IPv4 ルーティングテーブル	457
----------------------------	-----

はじめに

StorageTek Automated Cartridge System Library Software (ACSL) は、StorageTek 自動テープライブラリを制御する Oracle の StorageTek UNIX サーバーソフトウェアです。StorageTek ACS 製品ファミリは、完全に自動化されたテープカートリッジベースのデータストレージおよび情報検索システムで構成されます。StorageTek ACSL では、各種オペレーティングシステムで動作するワークステーションからメインフレーム、スーパーコンピュータまでの範囲に及ぶさまざまなクライアントシステムへのネットワークアクセスがサポートされています。

このガイドは、StorageTek ACSL の管理を担当する個人向けです。次のことに関して実用的な知識をすでに持っていることが前提となっています。

- UNIX ファイルおよびディレクトリの構造
- 使用しているプラットフォームに対応した UNIX コマンドおよびユーティリティを使用する方法
- UNIX システムファイル
- 一般的な UNIX システム管理者タスク (root としてのログオン、UNIX アプリケーションへのユーザーアクセスの設定など) を実行する方法

ドキュメントのアクセシビリティについて

オラクルのアクセシビリティについての詳細情報は、Oracle Accessibility Program の Web サイト (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>) を参照してください。

Oracle Support へのアクセス

サポートをご契約のお客様には、My Oracle Support を通じて電子支援サービスを提供しています。詳細情報は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>) か、聴覚に障害のあるお客様は (<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>) を参照してください。

第1章 概要

この章では、ACSLS の概要について説明します。

ACSLS について

Automated Cartridge System Library Software (ACSLS) は、StorageTek テープライブラリを制御する Oracle の StorageTek サーバーソフトウェアです。ACS (Automated Cartridge System) は、PTP (Pass-Thru-Port) 経由で接続されたテープライブラリのグループです。ACSLS はネットワーク経由のコマンド処理を介して、1 つ以上の ACS に保存された情報にアクセスし管理します。このソフトウェアには、システム管理コンポーネント、クライアントシステムのアプリケーションへのインタフェース、およびライブラリ管理機能が備わっています。

ACSLS HA について

ACSLS HA は、コンポーネントやサブシステムに障害が発生した場合に、中断なしのテープライブラリの制御サービスを保証するためにデュアル冗長性、自動回復、および自動フェイルオーバー回復を提供するハードウェア構成およびソフトウェア構成です。

ZFS ファイルシステムを持つ Solaris 11 で ACSLS 8.4 を実行するための詳細については、*ACSLS-HA* インストール、構成、および運用のガイドを参照してください。このバージョンでは、ユーザー定義のファイルシステムへの ACSLS ソフトウェアのインストールがサポートされます。

acssa および acsss ユーザー ID

このセクションでは、acssa および acsss ユーザー ID について説明します。

- *acssa* ログインは、*cmd_proc* へのアクセス、ライブラリ制御操作のコンソールユーザーインタフェース、および限定された一連の ACSLS ユーティリティーを提供します。

acssa の一般的なシェル環境には、*cmd_proc* を実行している 1 つまたは複数のウィンドウおよび ACSLS イベントログの実行中の末尾をモニタリングするウィンドウが含まれています。*acssa* ログイン環境は、これらの両方のリソースへのアクセスを提供します。

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

- acsss ログインは、これらと一般的な保守、構成、データベースのバックアップと復元、シェルユーティリティー、および一般的な診断のためのほかのすべての管理ユーティリティーへのアクセスを提供します。

acsss マクロ

acsss コマンドは、ACSL S アプリケーションに関連付けられている複数のサービスを操作する起動、停止、およびステータスのマクロです。セクション2章「[ACSL S の起動およびモニタリング](#)」および「[acsss マクロ](#)」を参照してください。

cmd_proc の使用

このセクションでは、*cmd_proc* について説明します。

cmd_proc ウィンドウ

次の例は、*acssa* としてログインしたときに表示される *cmd_proc* ウィンドウを示しています。カーソルモードでは、*cmd_proc* ウィンドウは、上部のセクションがメッセージ領域、および下部のセクションがコマンド領域である分割された画面です。ACSL S コマンドはプロンプトに入力します。

コマンドを受け入れるには、ACSL S が実行されている必要があります。「-q」オプションを指定して *cmd_proc* を起動すると、この最初の query server リクエストが抑制されます。

```
cmd_proc -q
-----ACSL S x.x.x-----
ACSSA>query server
2008-01-23 15:41:42

Server Status
Identifier      State   Free Cell Audit Mount Dismount Enter Eject
                Count   C/P    C/P    C/P    C/P    C/P
                run    234    0/0    0/0    0/0    0/0    0/0
```

cmd_proc の使用: 行モードと比較した場合のカーソルモード

ACSL S の *cmd_proc* は簡単に使用できるインタフェースであり、リクエストの処理中も一般的なサーバステータス情報を通知し続けます。*cmd_proc* のデフォルトモードはカーソルモードです。これは、ほとんどの端末タイプで適切に機能する汎用インタフェースであり、標準の 24 行、80 文字のウィンドウを使用します。カーソルインタフェースは、画面が 2 つのセクションに分

割されており、*STDERR* に送られたメッセージがウィンドウの上半分に表示され、*STDOUT* に送られたメッセージが下半分に表示されます。

ACSL S の *cmd_proc* をデフォルトのカーズモードで使用すると、一般的なサーバステータスメッセージがウィンドウの上部に表示され、ユーザー固有のやり取りが下部に表示されます。

カーズモードで保持される制限された履歴

カーズの 1 つの短所は、ユーザーと ACSLS サーバのやり取りの履歴を保持する機能が制限されていることです。それらのやり取りの領域は、24 行のウィンドウの下半分に制限されます。

この短所は、*cmd_proc* を行モードで使用すると克服できます。

```
cmd_proc -l
```

行モードでは、ユーザーはスクロールウィンドウのすべての利点を活用でき、やり取りの履歴はスクロール可能な端末バッファにロールアップされ、バッファのサイズにのみ制限されます。

行モードでのコマンドと混在したステータスメッセージ

行モードの動作の主な短所は、*STDOUT* および *STDERR* を個別の領域に分割できないことです。両方のソースの出力テキストは、画面上の同じスポット (リクエストを入力しようとしている端末の単一のカーソル行) に送信されます。

実行している *cmd_proc* セッションがシステム上の唯一のセッションである場合は、これは問題ではない可能性があります。ただし、アクティブな操作が ACSLS で進行中の忙しい本番環境では、ACSL S リクエストを入力している行と同じ行にステータス情報が出力されるウィンドウで作業することは、不満を感じないとしても困難である場合があります。

入力している行に表示されるシステムステータス情報は無視しても安全ですが、その情報をほかにリダイレクトできます。システムメッセージを別の出力先にリダイレクトするには、次のように行モードの *cmd_proc* を実行できます。

```
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.
```

作業中にシステムステータス情報を表示するには、2 番目のシェルウィンドウを開いて、ステータスメッセージを送信したファイルの実行中の末尾を表示します。

```
tail -f /tmp/SysChatter.out
```

意図した `cmd_proc` 操作を実行するには、`STDERR` を `/dev/null` にリダイレクトします。

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

カーソルモードの `cmd_proc` では 80 文字より長い行を表示できません

カーソルモードの `cmd_proc` コマンドは 80 文字より長い行を表示できず、80 文字より長い行を表示しようとする、`cmd_proc` ウィンドウがハングします。

この状況が発生した場合は、`Ctrl+c` および `Ctrl+d` を使用すると `cmd_proc` ウィンドウを解放できます。

すべての `query` およびその他のコマンドの出力は 1 行あたり 80 文字未満であり、表示コマンドによってすべてのレコードについて報告されるデフォルトのフィールドは 80 文字未満である必要があります。ただし、多くのオプションのフィールドを表示すると、行が 80 文字より長くなる場合があります。

多くのオプションのフィールドを表示する場合は、`cmd_proc` を (`-l` オプションを使用して) 行モードで起動することをお勧めします。たとえば、`cmd_proc` を `cmd_proc -l` として起動して、`display drive * -f volume type state serial_num wwn` を実行します。

`cmd_proc` を中断および再開する方法

UNIX コマンドを実行するために `cmd_proc` を中断して、`cmd_proc` を再開できます。`cmd_proc` を手動で起動する必要があります。`cmd_proc` で開始した進行中のリクエストは、`cmd_proc` が中断されている間も完了するまで続行されます。

`cmd_proc` を中断および再開するには、次の手順に従います。

1. `cmd_proc` の実行中に、`Ctrl+z` を押します。
2. UNIX のシェルプロンプトが表示されます。

目的の UNIX 操作を実行します。

3. `cmd_proc` を再開するには、UNIX の `fg` コマンドを入力します。

`cmd_proc` の終了

1. `cmd_proc` を実行しているときは、進行中のすべてのアクティビティーが完了して、`ACSSA>` プロンプトが返されるまで待ちます。

2. `cmd_proc` を終了するには、`logoff` コマンドを入力します。

```
logoff
```

3. `cmd_proc` セッションが終了します。

cmd_proc の起動

`cmd_proc` は `/etc/termcap` に定義されている任意の端末タイプから起動できます。カーソルモードで実行している場合、端末の表示サイズは 24x80 以上である必要があります。

`cmd_proc` セッションは ACSLS から独立したモードで実行されます。ACSLS の起動せずに `cmd_proc` セッションを開始すると、コマンドに応答はありません。ACSLS が実行されていないときに `cmd_proc` がコマンドを実行しようとして、ソケット通信エラーが表示されることがあります。

リモートログイン

ACSLS サーバーへのリモートアクセスは、SSH クライアントを持つシステムから使用できます。`ssh` クライアントは、Solaris、Linux、および MacOS を含むほとんどの POSIX 準拠のオペレーティングシステムのシェルの標準機能です。Windows 環境では、SSH クライアントソフトウェア (PuTTY、WinSCP、同様の商用アプリケーションなど) をインストールする必要があります。

ユーザー `acssa` としてリモートから ACSLS サーバーにアクセスするには、次のコマンドを入力します。

```
$ ssh acssa@hostname
```

ここで、`hostname` は ACSLS サーバーのホスト ID です。

`acssa` の一般的な削除環境には、`cmd_proc` を実行している 1 つ以上の SSH ログインシェル、および ACSLS イベントログの実行中の末尾をモニタリングする別のシェルが含まれています。

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

cmd_proc のキーボードショートカット

次の表は、`cmd_proc` のキーボードショートカット (<CTRL>+ キーストロークの組み合わせ) について説明しています。

表1.1 cmd_proc のキーボードショートカット

キーの組み合わせ	アクション	注記
<i>Ctrl+c</i>	最後の <i>cmd_proc</i> コマンドを取り消します。	<i>Ctrl+c</i> は、 <i>cancel</i> コマンドのキーボードショートカットです。 <i>cancel</i> コマンドの詳細については、「cancel」を参照してください。
<i>Ctrl+d</i>	<i>cmd_proc</i> プロンプトに戻ります。	現在のコマンドが完了している場合は、 <i>Ctrl+d</i> に効果はありません。現在のコマンドが処理されている場合、それは完了しますが <i>cmd_proc</i> は応答メッセージを表示しません。ACSSS プロンプトで現在のコマンドを入力し終わっていない場合、 <i>Ctrl+d</i> を押すとコマンドが削除されます。
<i>Ctrl+h</i>	コマンド行の前の文字を削除します。	ほとんどのキーボードでは、Enter キーまたは backspace キーも使用できます。
<i>Ctrl+i</i>	<i>cmd_proc</i> の表示をリフレッシュします。	この機能は、現在の <i>cmd_proc</i> の表示が通信回線上のノイズによって崩れている場合に役立ちます。
<i>Ctrl+r</i>	現在のコマンド行をリフレッシュします。	この機能は、現在のコマンド行の表示が通信回線上のノイズによって崩れている場合に役立ちます。
<i>Ctrl+r</i>	現在のコマンド行を削除します。	なし
<i>Ctrl+z</i>	<i>cmd_proc</i> を中断して、シェル環境にエスケープします。	<i>cmd_proc</i> を再開するには、 <i>C</i> シェルの <i>fg</i> コマンドを入力します。

cmd_proc の入力および出力のリダイレクト

入力ファイルを使用すると、*cmd_proc* を起動するときに、コマンドを自動的に入力 (*enter*) できます。たとえば、次の入力ファイルはカートリッジをマウントおよびマウント解除することによって ACSLS を検証します。

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

追加の cmd_proc ウィンドウでの入力ファイルの指定

cmd_proc を起動するには、次のコマンドを入力します。

```
cmd_proc -q < filename
```

また、`cmd_proc` を起動し、入力ファイルを指定して、出力を別のファイルにリダイレクトすることもできます。入力ファイルおよび出力ファイルを使用すると、`cmd_proc` の起動時に一連のコマンドを実行し、結果を参照できます。たとえば、次のファイルは、1つの入力ファイルのみを使用した `cmd_proc` を示す前の例で実行されたコマンドの結果を示しています。

```
ACSSA> query drive 0,0,0,0
1998-06-30 18:23:08
Identifier State Status Cartridge Type
0,0,0,0 online available 9840
ACSSA> query volume JPL1400
1998-06-30 18:23:09
Identifier Status Current location
JB1400 home 0,0,3,0,0
ACSSA> mount JPL1400 0,0,0,0
ACSSA> Mount: JB1400 mounted on 0,0,0,0
ACSSA> dismount JPL1400 0,0,0,0 force
ACSSA> Dismount: Forced dismount of JB1400 from 0,0,0,0
ACSSA> logoff
ACSSA
```

追加の `cmd_proc` ウィンドウへの出力のリダイレクト:

追加の `cmd_proc` を起動するには、入力ファイルを指定して出力をリダイレクトします。

1. `acssa` または `acsst` としてログインして、UNIX 端末ウィンドウを開きます。
2. `cmd_proc` を起動するには、次のコマンドを入力します。

```
cmd_proc -q < file1 > file2
```

ここで、`file1` は入力ファイルであり、`file2` は出力が送信されるファイルです。

デフォルトでは、`cmd_proc` の表示領域メッセージは `stderr` に書き込まれますが、これらのメッセージをリダイレクトすることもできます。次に例を示します。

```
cmd_proc -q < file1 > file2 2>> file2
```

アイドル状態への ACSLS の移行

この手順は、ACSLS をアイドル状態にすることによってリクエストの処理を中断する場合に使用します。通常、この手順は ACSLS を停止する前に使用されますが、それを使用して ACSLS のリクエストの処理を一時的に停止することもできます。

ACSLS をアイドル状態に移行するには、次の手順に従います。

`cmd_proc` から `idle` コマンドを入力します。

ACSLS は現在のすべてのリクエストを処理し、新しいすべてのリクエストを拒否して、アイドル状態になります。

ACSL S の再起動

この手順は、ACSL S を実行状態にすることによってリクエストの処理を再開する場合に使用します。通常、アイドル状態から ACSLS を削除するには ACSLS を再起動します。

ACSL S を再起動するには、次の手順を実行します。

cmd_proc から次のコマンドを入力します。

```
start
```

ACSL S がリクエスト処理を再開します。

ACSL S のディレクトリ構造

次の表は、ディレクトリ、サブディレクトリ、および ACSLS ディレクトリ構造内でもっともよく使用されるファイルおよびシェルスクリプトのリストを示しています。

ACSL S パスには 3 つの変数が使用されます。それらの機能を次に示します。

- `$installDir`

これは、ベースインストールディレクトリであり、デフォルトでは `/export/home/` です。

- `$ACS_HOME`

`$installDir/ACSSS/` にあり、これは `acsss` ユーザー ID のホームディレクトリであり、ACSL S 製品がインストールされています。

`$ACS_HOME` はデフォルトでは `/export/home/ACSSS` です。

- `$ACSDB_BKUP`

これは、ACSL S バックアップが保存されるディレクトリです。

表1.2 ACSLS のディレクトリ構造

ディレクトリ	内容
<code>\$installDir</code> (デフォルトでは <code>/export/home/</code>)	ベースインストールディレクトリ。
<code>\$installDir/SSLM</code>	ACSL S の java コンポーネント (ACSL S GUI および SMCE (論理ライブラリの操作) を含む) のホーム
<code>\$installDir/SSLM/AcsIlsDomain</code>	ACSL S の Web ベースの GUI アプリケーションのホームディレクトリ。

ディレクトリ	内容
<code>\$installDir/wlinstall</code>	バンドルされている WebLogic アプリケーションサーバーパッケージおよび関連するインストールスクリプト。
<code>\$installDir/Oracle</code>	バンドルされていない WebLogic のホームディレクトリ。
<code>\$installDir/acsls_thirdPartySoftware</code>	サードパーティーのライセンス情報および関連する再発行されたソースコードのコレクション。
<code>\$ACS_HOME (\$installDir/ACSSS)</code> (デフォルトでは <code>/export/home/ACSSS/</code>)	<code>acsss</code> ユーザー ID のホームディレクトリ。ACSLs のホームディレクトリでもあります。(ACS_HOME 環境変数はこのディレクトリを指しています。)
<code>\$ACSDB_BKUP</code> (デフォルトでは <code>/export/backup/</code>)	データベースのバックアップ
<code>\$ACS_HOME/config/</code>	ACSLs 構成ファイルが含まれています。
<code>\$ACS_HOME/data/external/</code>	アクセス制御、混在したメディア、およびカートリッジのレポートで使用されるカスタマイズされたファイルが含まれています。
<code>\$ACS_HOME/data/external/access_control/</code>	アクセス制御のサンプルおよびカスタマイズされたファイルが含まれています。
<code>\$ACS_HOME/data/internal/</code>	ACSLs 内部構成ファイル。変更しないでください。
<code>\$ACS_HOME/diag/bin</code>	診断ファイルおよびシェルスクリプトが含まれています。
<code>\$ACS_HOME/lib/</code>	実行時に必要な ACSLS のインストール済みの共有ライブラリが含まれています。
<code>\$ACS_HOME/log/</code>	ACSLs のイベントログおよびユーティリティーのイベントログファイルが含まれています。
<code>\$ACS_HOME (\$installDir/ACSSA/)</code> (デフォルトでは <code>/export/home/ACSSA/</code>)	<code>acssa</code> のホームディレクトリ。
<code>\$installDir/acscdb/</code> (デフォルトでは <code>/export/home/acscdb/</code>)	データベースのホームディレクトリ。
<code>\$LOG_PATH</code>	これは <code>\$ACS_HOME/log</code> と同等です。このディレクトリには、 <code>acsss_event.log</code> および ACSLS 操作に関連するその他の役立つログが含まれています。

第2章 ACSLS の起動およびモニタリング

ACSLS をインストールして、接続されるライブラリを構成したら、*acsss enable* コマンドを使用してアプリケーションを有効にできます。*acsss* マクロは、ACSLS に関連付けられている複数のサービスを操作し、それらを適切な順序で起動および停止したり、全体的なシステムステータスの概要ビューを表示したりします。

インストール内容によりませんが、ACSLS アプリケーションは Solaris または Linux システムにインストールされる最大 7 つのサービスで構成された集合体です。

- *acsdb* - ACSLS ライブラリデータベースを保守します。
- *acsls* - ライブラリ操作を実行するライブラリ制御ソフトウェア。
- *weblogic* - ACSLS GUI の Web サーバー。
- *surrogate* - java サービスと *acsls* の間の通信リンク。
- *rmi-registry* - 名前付き java オブジェクトおよびメソッドの検索サービス。
- *smce* - 論理ライブラリの SCSI メディアチェンジャーエミュレーション。
- *stmf* - 論理ライブラリのターゲットモードフレームワーク。

最初の 2 つのサービスは、すべてのインストールで共通しています。*weblogic*、*surrogate*、および *rmi-registry* サービスは、ACSLS GUI がインストールされている場所に存在します。*smce* および *stmf* サービスは、論理ライブラリのサポートが構成されている Solaris システムに存在します。これらのサービスのすべては、ACSLS ユーザーによって 1 つのマクロ *acsss* を使用して操作されます。

ACSLS の起動

root として、次のコマンドを実行して ACSLS を起動します。

```
acsss enable
```

このコマンドは ACSLS を起動するデフォルトの方法です。依存関係を確認し、さまざまな ACSLS サービスと ACSLS GUI を適切な順序でアクティブ化します。サービスは、システムのリポート後に自動的に起動されるように構成されています。

ACSLS のモニタリング

さまざまな ACSLS サービスの簡易的なステータスレポートを表示するには、次のコマンドを実行します。

```
acsss status
```

ACSLS の停止

ACSLS の停止は完全な停止ではなく、*acsls* および *smce* サービスが無効化されたあとでも、データベースと GUI ログインセッションのアクティブの状態が維持されるため保守作業を実行できます。ACSLS とデータベースを停止するには、次の手順を使用します。

ACSLS を停止するには、次のコマンドを使用します。

```
acsss disable
```

Solaris での SMF タイムアウト

Solaris SMF ユーティリティは、各サービスが完全に有効になるための時間設定を割り当てます。*acsls* サービスの場合、この時間制限はライブラリ構成 (LSM の数、ドライブ数、および CAP の数) を基準に計算されます。大きいライブラリ構成は小さい構成よりも ACSLS が回復するために時間がかかるため、大きい構成にはより長い SMF タイムアウト時間が割り当てられます。

まれに、障害のある LSM を起動するたによけいに時間がかかり、許可された SMF の時間制限を超える場合があります。タイムアウト時間が期限切れになると、SMF は処理を再起動します。このアクションでは、起動シーケンスが無限ループに陥り、起動が困難な状況で ACSLS の回復が妨げられることがあります。

特殊なファイル *acsls_startup_policy* は、このような状況で使うことが意図されています。このファイルは *\$ACS_HOME/data/external* ディレクトリにあり、構成すると、起動時の回復のために追加の時間が付加されたり、SMF 起動シーケンス中に特定の ACS の回復が除外されたりします。詳細な構成手順は、*acsls_startup_policy* のヘッダーの備考に含まれています。このファイル内の起動パラメータを調整することによって、異常なライブラリ起動状態によって発生する ACSLS の起動の問題を回避できます。

詳細は、「[ACSLS の起動に関する問題の診断](#)」を参照してください。

ACSL S の起動ポリシー

このファイルは ACSLS の起動時に適用される通常の起動パラメータを変更します。デフォルトの起動時の値を変更する場合は、慎重に分析し、Oracle ACSLS ソフトウェアサポートに相談することをお勧めします。

追加の起動時間

このパラメータは、Solaris 上の *acsls* サービスの SMF 起動タイムアウトに適用されます。*acsls* 起動タイムアウトは、現在のライブラリ構成によって自動的に計算されます。より多くの LSM、ドライブ、および CAP を持つライブラリには、より長いタイムアウトが指定されます。このタイムアウトは、ライブラリ構成が変更されると自動的に調整されます。計算された値は次のコマンドを実行することによって表示できます。

```
acsss timeout
```

自動計算されたタイムアウトが十分ではない場合、前の起動シーケンスを完了するための十分な時間が経過する前に、SMF 機能が介入して *acsls* サービスが再起動されることがあります。

起動シーケンスに追加の時間を与えると、そのような SMF の介入を防ぐことができますが、短所もあります。追加する時間が長すぎると、注意する必要がある可能性がある構成の問題点が隠されてしまうことがあります。通常のタイムアウト時間を長くすると、起動時の重大な問題または回復不可能な問題をオペレータに警告する SMF の機能に遅れが生じます。

acsls start のシーケンスが完了するための時間 (分) を追加するには、次の行の「=」のあとに整数値を指定します。

```
additional_startup_time=0 # Minutes
```

ACS の目的の (オフライン) 起動状態

ACSL S が起動すると、すべてのライブラリリソースが最後に設定された目的の状態になります。目的の状態がオンラインである場合、ACS をオンラインにする処理には、指定された ACS の物理ライブラリリソースが構成のデータベースイメージに対してチェックおよび検証される回復期間が発生します。この処理は、ライブラリ構成のサイズおよび異常な状況が存在するかどうかに応じて、1 分未満から数分かかります。

ACS のこの回復時間を回避するには、その ACS の目的の状態を指定して、関連付けられているポートをオフラインにします。そのようなアクションを行うと *acsls* SMF サービスが迅速

にオンラインステータスになりますが、実際の ACS を変更してポートをオンラインにするために、その後手動によるアクションが必要になります。

ACS の目的の起動状態を設定し、ポートをオフラインにするには、`$ACS_HOME/data/external/` ディレクトリ内の `acs1s_startup_policy` ファイルの該当する行の先頭からコメント文字 (`#`) を削除します。

たとえば、次の行を変更して

```
# ACS0_desired_startup_state_is_offline
```

次のようにします。

```
ACS0_desired_startup_state_is_offline
```

第3章 ACSLS ライセンスキー

StorageTek ACSLS バージョン 7.3.1 より、ACSLs ではソフトウェアによる使用権ライセンスの強制は使用されなくなり、ACSLs で有効なライセンスキーかどうかチェックされなくなりました。システムコンソールまたは `acsss_event.log` に、間もなく期限が切れるライセンスキーやライブラリ容量ライセンスに関するメッセージが表示されなくなりました。

次のユーティリティーは、有効なライセンスキーの設定およびチェックが機能しなくなりました。

- `licensekey.sh`
- `get_license_info.sh`

第4章 ACSLS GUI

ACSL 8.0 で導入された Web ベースの ACSLS グラフィカルユーザーインターフェース (GUI) は、ライブラリ操作の処理とモニタリング、および論理ライブラリの管理と操作のためのグラフィカルコンソールを備えています。この GUI を使用する手順と詳細情報については、オンラインヘルプを参照してください。

GUI は次を備えています。

- 柔軟性と使いやすさが大幅に拡大された代替のライブラリコンソール。これは、レガシー `cmd_proc` で使用できるほとんどの操作を備えています (例外は「[ACSL 8.4 GUI で除外されたシステム操作](#)」に示されています)。
- 論理ライブラリのサポート。

論理ライブラリの詳細については、「[論理ライブラリについて](#)」を参照してください。

- ライブラリの管理および操作。論理ライブラリの管理に関連する新しい操作とともに、レガシー `cmd_proc` 操作のほとんどを実行できます。
- テープライブラリコンポーネントのリアルタイムモニター。
- 物理構成および論理構成をナビゲートするツリーブラウザ。
- 各画面から表示できるリアルタイムのアラート。

アラートは、ハードウェアの損傷、データの破損、またはアプリケーションソフトウェアの破損につながる可能性のある状況を知らせます。常にそれが属する情報の前に表示されます。

- ユーザーが指定した条件でボリュームとドライブをフィルタする機能。
- システムイベントとシステムログの、ユーザーがカスタマイズしたビュー。
- オンラインヘルプ。

論理ライブラリを作成、管理、または削除するには、ACSL 8.4 グラフィカルユーザーインターフェース (GUI) を使用する必要があります。GUI は自動的に ACSLS 8.4 インストールパッケージに組み込まれています。論理ライブラリの詳細については、[付録H「論理ライブラリのサポート」](#)を参照してください。

ACSLs GUI の起動

ACSLs GUI を制御するソフトウェアは、ACSSS サービスが有効になると自動的に起動します。ACSLs GUI は、Solaris プラットフォーム上の WebLogic 内のその他のアプリケーションの間で配備されます。

ACSLs GUI にログインするには:

1. ブラウザを開き、サーバーのホスト名または IP アドレスを含む URL を送信します。

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

または

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

ホストシステムの完全修飾ホスト名または IP アドレスを使用することをお勧めします。WebLogic で URL を完全に解決できない場合は、一部のページ (ACSLs のヘルプページを含む) が正しく表示されない可能性があります。

ポート 7001 で http を使用している場合は、ポート 7002 で https を使用するように WebLogic によって自動的に再ルーティングされます。

WebLogic ではセキュアな https プロトコルが使用されているため、サイトのセキュリティ証明書が登録されていないために、信頼されないことを示す警告がブラウザに表示される可能性があります。確実に URL がローカルの ACSLS サーバーである場合は、続行しても安全です。この時点で、ログイン画面が表示されます。ブラウザがセキュリティ証明書を要求する場合は、[「HTTPS 用の自己割り当てデジタル証明書の構成」](#)を参照してください。

2. ログイン画面で、有効な ACSLS ユーザー ID と対応するパスワードを入力します。これらは、WebLogic をインストールするとき、または `userAdmin.sh` を使用して設定するユーザー ID です (`acsls_admin` など)。
3. 正常にログインすると、ACSLs GUI の「Welcome」ページが表示されます。

GUI のユーザーとパスワードの管理

ACSLs GUI のユーザーとパスワードを追加したり管理したりするには、`userAdmin.sh` のメニュー方式のユーティリティを使用します。[「userAdmin.sh」](#)を参照してください。

GUI の概要

次のセクションでは、GUI の基本的な概要について説明します。

GUI の「Welcome」画面

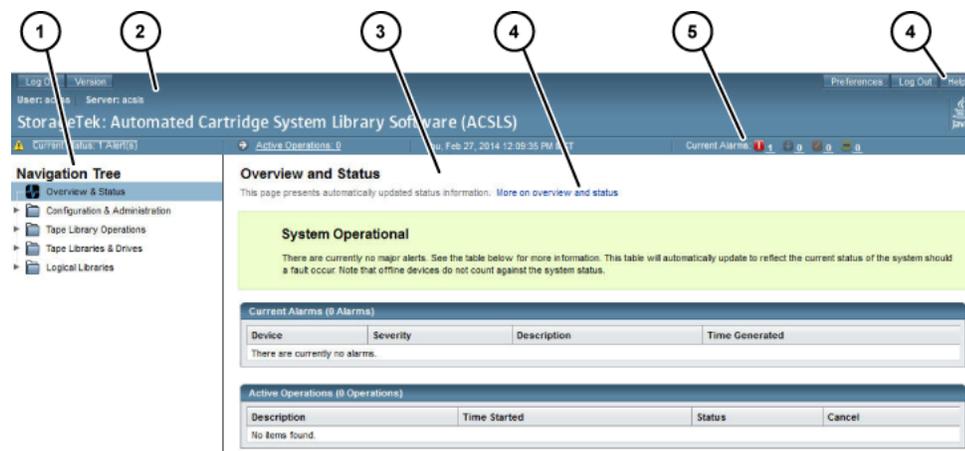
ACSLS GUI の「Welcome」画面には 3 つのウィンドウが表示されます。

- 上部フレームのマストヘッド
- 左側のフレームのナビゲーションツリー
- 右側のフレームの「Overview and Status」ページ

ブラウザへの GUI の初期ロード後に、右側のフレームが表示されない場合は、ブラウザをリフレッシュしてください。

次の図は、ACSLS GUI を示しています。

図4.1 ACSLS グラフィカルユーザーインターフェース



凡例:

1. ナビゲーションツリー
2. マストヘッド
3. ページ
4. ヘルプリンク
5. アラーム

マストヘッド

マストヘッドには次のボタンがあります。

- Log Out

「Log Out」ボタンはユーザーを GUI からログアウトします。

セキュリティ向上のため、GUI からログアウトするたびにブラウザ履歴をクリアすることをお勧めします。

- Version

「Version」ボタンは、現在の ACSLS のバージョンとコピーライト表示を表示します。

- Preferences

「Preferences」ボタンは、GUI システム設定を指定できるページを開きます。

- Logout

「Logout」ボタンは、現在のセッションを終了します。

- Help

「Help」ボタンは、オンラインヘルプシステムを起動します

- Active Operations

「Active Operations」ボタンは、進行中の ACSLS 操作の数を表示します。これは、アラームとアクティブな操作に関する詳細が表示される「Overview and Status」ページへのハイパーリンクです。

- Current Status

「Current Status」ボタンは、アクティブなアラートの数を表示します。これは「Overview and Status」ページに移動できるハイパーリンクです。

- Date and Time of Login

「Date and Time of Login」は、現在の GUI セッションの最初のログイン時間を示すタイムスタンプです。

- Current Alarms

「Current Alarms」ボタンは、現在のシステムアラームの数と重大度を示すダッシュボードビューです。現在のアラームの詳細は、「Overview and Status」ページで要約されています。

アラームのカラーコード

次の表では、アラームのカラーコードについて説明します。

色	重大度レベル	影響を受けるデバイス
赤色	クリティカル	これは通常の ACSLS 操作を著しく阻害する状況を示します。クリティカルのアラームは、根本的な問題に対する早急な対応が必要なときに表示されます。結果として生じる影響によって、通常の ACSLS ライブラリ操作がシステム全体の停止を引き起こします。
黒色	停止	このアラームは、ACSLs 内の 1 つ以上のサブシステムがオフラインであることを示します。オフラインステータスは、意図的な手動操作によることもあれば、ほかの要因によることもあります。通常は、関連する技術的要因の存在が、黄色 (マイナー) のアラームまたはオレンジ色 (メジャー) のアラームのアイコンによって示されます。
オレンジ色	メジャー	メジャーのアラームは、ソフトウェアか、ACSLs またはその GUI の機能を有効にするベースとなるコンポーネントで認識された誤動作を示します。
黄色	マイナー	この色は、システムが完全には機能していないことを意味します。システムが初期化中の可能性もあり、動作していない 1 つ以上の分離したサブシステムが存在する可能性もあります。この中断は、一時的なものとして、またはそれが表す機能が分離していて全体的なライブラリ制御操作に不可欠ではないときに見られます。

マストヘッドのアラームは、そのカテゴリに少なくとも 1 つのアクティブなアラームが存在しないかぎり、色で点灯しません。アラームの実際の数、色のアイコンの隣にある数値テキストに表示されます。アクティブなアラームがない場合、この数字はゼロ (0) を示しています。

色の識別が困難なユーザーのために、マウスをそれぞれの色のアイコンの上に置いて確認できる「ツールチップ」機能があります。小さいポップアップウィンドウに標準テキストで、そのグラフィカルアイコンの意味が表示されます。

System Preferences

このページにはマストヘッドの「Preferences」ボタンからアクセスします。このページには、個人の設定に応じてシステムの動作を変更できる多数のドロップダウンメニューがあります。

- Default Tree Menu - 展開または縮小のメニュー形式を指定します。
- Log Page Size - ログビューアでページ上に表示するイベントの数を指定します。
- Alert Update Interval - ブラウザがシステムアラートに対してサーバーを検証する必要がある頻度を指定します。GUI の応答時間が遅い場合は、この間隔を大きくできます。
- Number of Days to Retain Events - 「System Events」ページに表示する履歴の数を指定します。
- Enable Fast Load for SCSI Clients - 有効なマウントリクエストを受信した直後に SCSI クライアントに成功の応答を返すには (yes)、ロボットマウント操作が実際に完了するまで待機するには (no) を指定します。

ナビゲーションツリー

ナビゲーションツリーは常に ACSLS GUI の左側のフレームに配置されます。縮小ツリーには、次へのナビゲーションリンクが表示されます。

- 「Overview and Status」ページ
- 「Configuration and Administration」ページ
- 「Tape Library Operations」ページ
- 「Tape Libraries and Drives」ページ
- 「Logical Libraries」ページ

Overview and Status

このページは、マストヘッドのアラームアイコンで示されるサービスの具体的な誤動作または中断を識別するときに進む最初の場所です。このページの上部には、システム全体の状況の重大度を反映するステータスのサマリーが表示されます。

図4.2 ACSLS の「Overview and Status」ページ

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.			

ステータスサマリーの下に 2 つのボックスが表示されます。それらの機能を次に示します。

- Current Alarms

「Current Alarms」には、影響を受けるデバイス、重大度、説明、およびイベントが発生した時間が表示されます。重大度レベルは次のとおりです。

- 「System Critical」は、赤色のアイコンと赤色の背景色で示されます。これは通常の ACSLS 操作を著しく阻害する状況を示します。
- 「System Degraded」は、黄色のアイコンと黄色の背景色で示されます。これはシステムが完全には機能していないことを意味します。システムが初期化中の可能性もあり、動作していない 1 つ以上の分離したサブシステムが存在する可能性もあります。

- 「System Operational」は、緑色の背景色で示されます。これは、すべてのサブシステムが機能していて、システムコンポーネントでエラーが検出されていないことを示します。
- アクティブなアラーム

「Active Operations」の表には、現在進行中のライブラリ操作がリストされます。それぞれの操作は、簡潔な説明、アクションが開始された時間、現在のステータス、および操作を取り消すためのラジオボタンのオプションとともにリストされます。

Configuration and Administration

「Configuration and Administration」セクションには、ACSL Sシステムをモニターできるページと、論理ライブラリ構成を管理できるその他のページへのリンクが含まれています。ツリーのこのリーフを展開すると、次のサブメニューが表示されます。

- Log Viewer
- System Events
- Logical Library Configuration

Log Viewer

デフォルトの「Log Viewer」ページには、システムイベントをモニターするさまざまなログ (*acsss_event.log*、*smce_trace.log* など) の実行中の末尾が表示されます。このページには、ログの履歴をスクロールして先頭に戻ることができるナビゲーションボタンがあります。

System Events

個々のライブラリの操作はすべて、「System Events」ログに記録されます。このログの各レコードには、イベントのタイムスタンプ、イベントのタイプ、およびイベントの説明が含まれています。

Logical Library Configuration

ツリーのこのリーフを展開すると、論理ライブラリの構成および管理のためのさまざまなオプションが開きます。

- 新しい論理ライブラリの作成
- 新しい論理ライブラリの編集
- 論理ライブラリへのボリュームの割り当て
- 論理ライブラリからのボリュームの割り当て解除

- クライアント接続の管理。

論理ライブラリの詳細については、[付録H「論理ライブラリのサポート」](#)を参照してください。

Tape Library Operations

テープ操作により、次のことができます。

- ACS、LSM、またはパネルの監査
- ボリュームのマウント
- ボリュームのマウント解除
- ボリュームの挿入
- ボリュームの取り出し

Tape Libraries & Drives

テープライブラリとドライブについての次の詳細情報が表示されます。

- 処理を停止または起動できるライブラリ操作
- 表示する機能:
 - ライブラリサーバーの物理的な状態
 - 過去 1 時間以内のマウントと過去 24 時間以内のマウントをグラフで示す現在のアクティビティ
 - 次を含む物理コンポーネント:
 - ACS の合計数
 - 容量
 - 合計ボリューム
 - 合計ボリューム (タイプ別)
 - 合計空きセル数
 - セルの使用状況のグラフ表示
 - 合計 LSM
 - 合計 LSM (タイプ別)
 - 合計ドライブ
 - 合計ドライブ (タイプ別)
 - CAP の合計数
 - 論理ライブラリのリスト

ACSL S 8.4 GUI で除外されたシステム操作

ほとんどのインストールユーティリティー、診断、およびその他のシェルコマンドユーティリティーは、ACSL S GUI コンソールのスコープ外にあります。レガシー `cmd_proc` のいくつかの操作は、リリース 8.4 の ACSL S GUI には実装されていません。これらには次が含まれます。

- プールの定義、削除、照会、または表示
- パネルの表示
- サーバーのアイドルまたは起動
- ボリュームのロックまたはロック解除
- ドライブのロックまたはロック解除
- スクラッチボリュームのマウント
- LMU の照会または切り替え
- `query mount` の実行
- スクラッチボリュームの設定
- ロックの設定またはクリア
- ポートの変更、照会、または表示
- `venter`

ファイアウォールと GUI

ACSL S サーバーでファイアウォール保護が有効になっている場合は、ファイアウォールソフトウェアを使用して明示的なポリシーが構成されていないかぎり、ACSL S GUI へのリモートユーザーのアクセスに影響します。

Solaris の場合

`ipf` および `ipfilter` のマニュアルページを参照してください。`ipfilter` ファイアウォールは、`root` で次のコマンドを使用して有効化 (無効化) します。

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

- `ipfilter` の現在のステータスを確認するには:

```
svcs ipfilter
```

ファイアウォールポリシーは、ファイル `/etc/ipf/ipf.conf` で定義できます。ローカルホスト上のコンポーネント間 (ACSLs と WebLogic 間など) の自由な通信を許可するには、以下の例のような文を含めます。

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

- リモートの Web ベースのブラウザによる ACSLS GUI のアクセスを許可するポリシーを指定するには、ポート 7001 および 7002 を開く必要があります。

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

ACSLs ACSAPI クライアントの場合は、ACSLs で使用中のポートを検出する必要があります。UNIX シェルから次のコマンドを使用します。

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

ポート ID は、表示の最終フィールドにリストされます。これらのそれぞれに「pass in quick」文を追加する必要があります。RPC ポートマッパーポート 111 に「pass in quick」文を追加する必要がある場合もあります。

ACSLs のファイアウォールの詳細情報については、[付録M「ファイアウォールセキュリティオプション」](#)を参照してください。

提示されたルールセットの最後の文 `block in from any` は、前の文で特に許可されていないかぎり、どのトラフィックもホストに到達しないと仮定しています。

Linux の場合

iptables のマニュアルページを参照してください。iptables ファイアウォールは、ユーザーの `root` で次のコマンドを使用して有効化 (または無効化) します。

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

- iptables のステータスを確認するには:

```
service iptables status
```

iptables のポリシーファイルは `/etc/sysconfig/iptables` です。ACSLs GUI へのリモート http/https アクセスを許可するポリシーを追加するには、以下の例のような文を使用して、ポート 7001 と 7002 に例外が含まれるように、このファイルを更新してください。

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

iptables の詳細については、「[GUI が動作していることの検証](#)」を参照してください。

ACSL S ACSAPI クライアントの場合は、ACSL S で使用中のポートを検出する必要があります。Linux シェルから次のコマンドを使用します。

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

ポート ID は、表示の最終フィールドにリストされます。iptables ポリシーにこれらの各ポートの例外を追加する必要があります。RPC ポートマッパーポート 111 に例外文を指定する必要があります。

ACSL S のファイアウォールの詳細情報については、[付録M「ファイアウォールセキュリティオプション」](#)を参照してください。

HTTPS 用の自己割り当てデジタル証明書の構成

ACSL S サーバーに WebLogic をインストールすると、自動的に簡易 512 ビット公開鍵が使用可能になり、クライアントブラウザとの基本的な https 交換がサポートされます。通常、追加の構成作業が必要になることはありません。ただし、Microsoft Internet Explorer などの一部のブラウザでは、1024 ビット以上の長さの鍵が必要になります。

Internet Explorer や FireFox バージョン 39 以上では、サードパーティーのデジタル署名認証局によって検証された証明書を採用していない https サーバーで使用するために WebLogic セットアップ手順が必要になります。

手順については、『[ACSL S 8.4 インストールガイド](#)』の「HTTPS 用の自己署名デジタル証明書の構成」を参照してください。

第5章 ライブラリハードウェアの設置と構成

ライブラリハードウェアの設置と構成には、次のタスクが含まれます。

- 「[ライブラリハードウェアへの接続のインストール](#)」

ライブラリハードウェアへの接続のインストールには、次が含まれます。

- SCSI mchanger デバイスドライバの追加
- オプションのデュアル TCP/IP サポートの有効化 (SL8500 または SL3000 の場合)

- 「[acsss_config を使用したライブラリハードウェアの構成](#)」

これにより次ができるようになります。

- ライブラリの通信を確立します
- サポートするライブラリの数と、それらをパーティション分割するかどうかを構成します
- ライブラリへの接続が SCSI/ファイバ、TCP/IP、シリアル、または SCSI/ファイバかどうかと、使用する形式を設定します
- ライブラリハードウェアを再構成します

注:

サポートされているライブラリ、ドライブタイプ、メディアタイプの現在のリストと、ドライブとメディアの互換性については、『[ACSL S 製品情報ガイド](#)』を参照してください。

ライブラリハードウェアへの接続のインストール

このセクションでは、ライブラリハードウェアへの接続のインストールについて説明します。

SCSI mchanger デバイスドライバの追加

SCSI メディアチェンジャー (mchanger) は、ACSL S と Solaris オペレーティングシステム上の SCSI またはファイバ接続のライブラリ間で通信を行うデバイスドライバです。Linux では、ACSL S はネイティブの sg ドライバを使用します。

Solaris および Linux の両方で、ACSL S は、ACSL S に対してライブラリハードウェアを構成するときに指定する `/dev/mchanger` リンクも作成します。

SCSI ライブラリのサポートの追加は、インストールプロセスのオプション部分です。ただし、新しい SCSI ライブラリを、または単に SCSI サポートを追加するために、既存の ACSLS インストールに追加する必要がある場合もあります。いつでも必要なドライバをインストールして mchanger リンクを作成できます。

1. ACSLS サーバーに *root* ユーザーとしてログインし、パスワードを入力します。
2. インストールディレクトリに移動します。

```
cd $ACS_HOME/install
```

3. SCSI ライブラリが機能していて、ACSL S サーバーに物理的に接続されていることを確認します。

注:

(Solaris) マルチパスハードウェアが意図的に SL500 ライブラリ用に構成されている場合を除き、ライブラリ接続バスで使用される「fp」ドライバなどの親のデバイスドライバのマルチパス I/O を無効にすることが重要です。SL500 に正常に接続することが困難な場合は、<driver>.conf ファイル (通常は /kernel/drv/fp.conf) でマルチパス I/O が無効にされていることを確認する必要があります。

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

4. 適切なドライバインストールスクリプトを呼び出します。

- Solaris

```
./install_scsi_sol.sh
```

- Linux

```
./install_scsi_Linux.sh
```

SL8500 または SL3000 デュアル TCP/IP サポートの有効化

SL8500 または SL3000 ライブラリがインストールされていて、デュアル TCP/IP サポートを有効にする場合は、「[デュアル TCP/IP サポート](#)」を参照してください。また、これらのセクションでは、作成する必要があるカスタマイズ済みのルーティングテーブルのエントリについても説明します。

ライブラリハードウェアの構成または再構成

ライブラリハードウェアの構成または再構成には 2 つの方法があります (新規または変更されたライブラリかテープドライブ、またはその両方)。

- *acsss_config*

このコマンドは、ACSL S が停止しているときに実行する必要があります。*acsss_config* を使用します。

- ライブラリハードウェアの初期 (最初の) 構成の場合
 - ACS を削除するため
 - ライブラリへのポート接続を変更または削除するため
 - ファイバまたは SCSI 接続のライブラリを再構成するため
- 動的構成 (*config*) ユーティリティ

ACSL S が実行中の間は、このユーティリティを実行します。上記の除外されたものを除くすべての変更には、*config* を使用します。*config* ユーティリティは、

- 新しいライブラリを追加します
- TCP/IP 接続のライブラリ構成を更新します
- ポート接続、LSM、CAP、およびドライブを追加します

acsss_config を使用したライブラリハードウェアの構成

acsss_config を使用すると、ライブラリハードウェアを構成または再構成できます。

- 各 ACS には少なくとも 1 つの CAP が必要です。これは、別のパーティションと共有される CAP にできます。
- ACSLS システム全体に対して少なくとも 1 つのドライブが構成されている必要があります。

たとえば、ACSL S が 4 つのライブラリをサポートしている場合は、3 つのライブラリにはゼロ (0) 個のドライブを含めることができます。ただし、4 番目のライブラリには、少なくとも 1 つのドライブが含まれている必要があります。

- ライブラリハードウェアを構成します

はじめてライブラリハードウェアを構成するには、*acsss_config* ユーティリティを使用する必要があります。*acsss_config* ユーティリティはメニュー方式であり、オプション 8 を選択すると、はじめてのハードウェアの構成、ライブラリの削除、またはライブラリへのポート接続の変更などのタスクを実行できます。

注:

ACSL S では、ライブラリを構成または再構成するときに ACS 番号をスキップできます (オプション 6 を参照)。acsss_config と動的構成は、両方とも ACS 番号の指定とスキップをサポートします。たとえば、9310 の ACS 0 から SL8500 の ACS 1 に移行したあと、残りの SL8500 ACS 1 でドライブおよびボリューム ID の番号を付け直さなくても ACS 0 を削除できます。

- ライブラリハードウェアを再構成します

acsss_config を使用してすべてのライブラリの構成情報を更新する前に、現在の構成を記録してください。現在のライブラリに割り当てられている ACS 番号を変更すると、すべてのボリュームが不在とマークされ、ドライブのアドレスすべてが変更されます。query lmu all コマンドからの出力を保存して、現在の ACS 番号とポート接続を記録します。

注:

既存の物理 ACS に対するクライアント接続がある論理ライブラリがまだ存在しているときに、acsss_config でその ACS を構成から削除すると、SMCE サブシステムが保守状態になることがあります。

これを回避するには、acsss_config を使用してライブラリ構成から ACS を削除する前に、ACSL S GUI (または、ACSL S 8.2 以降のリリースでは lib_cmd CLI) を使用して、関連付けられたすべての論理ライブラリを削除する必要があります。関連付けられたライブラリを最初に削除すると、関連するすべての情報が確実に正しく削除されます。

ライブラリの初期構成後に、config ユーティリティを使用して、ACSL S を停止せずに ACS、LSM、またはテープドライブを動的に追加または再構成できます。

詳細と手順については、「[acsss マクロ](#)」を参照してください。

注:

ライブラリの物理構成が変更されるたびに、acsss_config または config を使用して ACSL S データベースを更新する必要があります。データベースに定義されている構成がライブラリに定義されている構成に一致しない場合、ACSL S は正しく実行されません。

例:

データベースの更新が必要な構成変更には、次が含まれます。

- ACS、LSM (SL500 などの SCSI 接続の LSM を含む)、パススルーポート (PTP)、トランスポート、またはスタンバイ LMU の追加または削除

注意:

新しい ACS または LSM のインストール後は、ACSLs を再構成する前に、接続されている各ライブラリおよび LSM が完全に (ハードウェア) 構成され、電源が投入され準備が整っていることを確認してください。そうしないと、*acsss_config* または *config* でライブラリが正しく構成されません。

注:

追加したライブラリを ACSLS HA でモニターするためには、*ha_acs_list.txt* ファイルにエントリを作成する必要があります。詳細については、「[ACSLs HA について](#)」を参照してください。

- 。サーバーシステムと LMU 間のポート接続の追加または削除

acsss_config を起動するには:

1. *acsss* としてログインします。
2. ライブラリハードウェアを再構成している場合は、この手順に従います。

ACSLs を停止する前に *query lm u all cmd_proc* コマンドを使用して既存の ACS 番号、パーティション ID、およびポート接続を表示し、この情報を記録します。

3. ACSLS が実行されている場合は停止します。
4. 構成ユーティリティを実行します。

acsss_config

「*ACSLs feature configuration*」画面が表示されます。

注:

構成ユーティリティ *acsss_config* では、すべてのライブラリ、LSM、およびトランスポートが完全に構成され、電源が投入されて準備が整っていないかぎり、ライブラリが正しく構成されません。

次のメニューが表示されます。

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:

注:

オプション 1-7 の説明については、6章「[ACSLs の動作を制御する変数の設定](#)」を参照してください。

5. オプション 8 を選択します。

次のプロンプトで構成の設定を進めます。

- プロンプト: *Configure library communications? (y/n):*

ACSLs サーバーとライブラリ間の通信を確立または更新するには、*y* と入力します。これには、ライブラリへのポート接続の追加、削除、または変更と、ライブラリパーティションの追加、削除、または変更が含まれます。

n を選択すると、現在のライブラリの構成がリフレッシュされます。

ACS を追加または削除したり、ライブラリへの接続を追加、変更、または削除したりする必要がない場合は、ライブラリ通信の構成をバイパスしてハードウェア構成を単にリフレッシュすることが、もっとも簡単で最良のオプションです。ライブラリへの現在のすべての接続の再定義をスキップし、ACSLs データベースに記録されているライブラリ構成をリフレッシュするだけです。これにより、ライブラリへの接続を省略したり誤って指定したりするリスクが排除されます。

- プロンプト: *Library server database exists and will be overwritten, continue? (y or n): y*

このプロンプトは、ライブラリ通信があらかじめ確立済みで既存のデータベースが存在する場合のみ表示されます。

このプロンプトに対して *y* と入力すると、構成が変更されます。

6. サポートする ACS の数を指定します。

- プロンプト: *Number of ACSs to be supported:*

サイトでサポートする ACS の数 (1 から 32 まで) を入力します。少なくとも「1 つ」の ACS が必要です。1 つまたは複数の ACS をサポートするハードウェアが設置されている必要があります。

注:

パススルーポートで接続された L700e ライブラリのペアは、1 つの ACS としてカウントされます。

連続する ACS 番号を割り当てずにライブラリを構成または再構成するときは、ACS の番号付けをスキップできます。

7. 各 ACS 番号を指定します。

- プロンプト: *Please enter the first ACS number [default: 0]:*

この ACS の番号を入力します。

注意:

既存の ACS を再構成するときは、それらに現在割り当てられているのと同じ ACS 番号を指定してください。

そのあと、ACS ごとのデバイス接続を定義するよう求められます。

8. ライブラリのパーティションを指定して定義します。

- プロンプト: *Is ACS #n in a partitioned SL8500 or SL3000? (y or n)*
 - *y* と入力すると、ACS のパーティション ID を尋ねられます。

このパーティション ID は SL コンソール上のパーティション ID に一致する必要があります。

- これがパーティション分割されたライブラリまたは SCSI/ファイバ接続のライブラリではない場合は、*n* と入力します。

ACSLs は、パーティション分割された SCSI/ファイバ接続ライブラリ (SL500 など) をサポートしていません。また、パーティション分割された SCSI/ファイバ接続ライブラリには、パーティション ID はありません。

9. ACS ごとにデバイス接続 (ポート) を定義します。

- プロンプト: *Number of connections to ACS #n*

このプロンプトは構成されている ACS ごとに表示され、各 ACS の通信ポートの番号を設定します。接続は次のとおりです。

- SCSI 接続

注:

バススルーポートで接続された L700e SCSI ライブラリのペアがある L1400 には、2 つの接続 (各 L700e に 1 つ) があります。

- ACSLS とライブラリ間のシリアルまたは TCP/IP 接続。少なくとも 2 つの接続が推奨されます

複数の SL8500 ライブラリに接続する場合は、最大 15 個の接続が許可されます。

ACS は物理的に接続する必要があります。1 から 15 の 10 進数を入力してください。

- プロンプト: *Device or host - ACS#n, connection #n:*

接続ごとにデバイスまたはホストを入力してください。

注意:

必ず正しいホスト名または IP アドレスを指定してください。間違ったライブラリに接続しないでください。

ACS に指定したすべての接続が、実際に同じ ACS に接続されていることを確認します。

TCP/IP 接続の LMU への接続を指定するには、次のいずれかを入力します。

- IP アドレス
- ホスト名
- 完全修飾ホスト名

注:

ホスト名を指定する場合は、このホスト名を、SL8500 または SL3000 ライブラリ、または 9330 LMU に入力された IP アドレスにマップする必要があります。このホスト名と IP アドレスのマッピングはサイト固有です。通常これは、*/etc/hosts* ファイル、ドメインネームサーバー、NIS、または NIS+ によって行われます。

例:

SL8500、SL3000、または 9300 ライブラリの TCP/IP デバイス名の例

```
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: 192.168.174.31
```

注:

ACSLs は SL3000 への TCP/IP 接続のみをサポートしています。ACSLs はファイバ接続の SL3000 をサポートしていません。9330 ACS への接続は、シリアルまたは TCP/IP のいずれかにはできますが、両方にはできません。

例:

4400 または 9300 ライブラリにシリアルデバイス名の例

```
Device or host - ACS #0, device #0: /dev/ttya
```

```
Device or host - ACS #0, device #1: /dev/ttyb
```

ファイバまたは SCSI 接続のライブラリは、mchanger デバイスを介して接続されます。

例:

SCSI ライブラリのデバイス名の例

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

パススルーポートで接続された L700e SCSI ライブラリのペアがある L1400 には、2 つの接続 (各 L700e に 1 つ) があります。

例:

L700e ライブラリのペアのデバイス名の例

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

```
Device or host - ACS #1, connection #1: /dev/mchanger3
```

定義する ACS が複数ある場合は、それらの ACS 番号を指定するように求められます (手順 6 を参照)。

10. ACSLS データベース内の構成を作成または更新します。

- プロンプト: *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):

y と入力します。

続行する前に、ライブラリが完全に構成されていることと、すべてのサブシステムが機能していて電源が投入されていることを確認します。

この手順は、ポート接続の追加または削除を除くすべての構成変更でデータベースを更新します。

- *プロンプト: Library server database exists and will be overwritten, continue? (y or n):*

y と入力します。スクリプトがライブラリ構成を構築すると、各 LSM の各パネルに次のメッセージが表示されます。

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

また、このスクリプトはライブラリ構成レポートを生成し、それを次のファイルに追加します。

\$ACS_HOME/log/acsss_config.log

11. オプションで、Solaris 上に TCP/IP 以外のクライアントを構成します。

- *プロンプト: Configure client system interfaces? (y or n):*

システムからクライアントシステムインタフェースを構成するかどうかを確認されたら、回答します (*y* または *n*)。

OSLAN プロトコルで ICL クライアントを使用している場合を除き、*n* と入力します。

y と入力すると、次のメッセージが表示されます。

CSI SELECTION TABLE

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

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

y の場合は、次のメッセージが表示されます。

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

ONC/ RPC は常に選択されます。

注:

`acsss_config` は、データベースを自動的にバックアップしてから終了します。

12. `acsss enable` を実行して、サーバーを起動します。

イベントログを調べると、すべてが動作していることを確認できます。

サーバーシステムがリブートされると、ACSL S が自動的に起動します。

13. ライブラリの監査を実行します

監査は、ライブラリカートリッジの実際のインベントリに一致するように ACSLS データベースを更新します。

これで、ACSL S によるライブラリ操作の準備が整いました。ACSL S のコマンドを入力する場合は、`acssa` としてログインします。

ACS の番号の変更

既存の ACS の番号を変更する必要がある場合、または望ましい場合は、[「acs_renumber.sh」](#)を参照してください。

第6章 ACSLS の動作を制御する変数の設定

ACSLS には、静的変数と動変数の両方があります。

- 静的変数

静的変数は、変数について説明し、新しい値を設定できるプロンプトの文に示されます。

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

注:

ACSLS の静的変数を変更したあとに新しい値を使用するには、ACSLS を停止して再起動する必要があります。

- 動変数

ACSLS の動変数への変更は、ACSLS 操作中にその変数が次に参照されるときに有効になります。

ACSLS 変数を表示および更新する簡単な方法

- すべての ACSLS 変数 (静的および動的) の現在の設定を表示します。

```
dv_config -d
```

- 変数の現在の設定をファイルに保存します。

```
dv_config -d > filename
```

- ファイルを表示します。

```
vi filename
```

- 動的オプションの値を出力します。

```
dv_print
```

- 1つの変数の現在の設定を、変数について説明するプロンプトとともに表示し、設定を変更できるようにします。

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

ここでは:

- `-p`

プロンプトでは、変数に新しい値を指定できます。

- `<variable_name>`

目的の変数名をここに挿入します。

- `-u`

変数に変更された場合は、共有メモリーの動的変数の値を更新します。`-u` オプションは静的変数には使用しません。

注:

変数のより完全な説明を表示するには、プロンプトで疑問符 (?) を入力します。プロンプトがリフレッシュされ、変数を変更できます。

`acsss_config` メニューを使用して、ACSL S の動作を制御する変数を設定することもできます。オプション 1-8 を使用して、次の操作を実行します。

- [「CSI チューニング変数の設定」](#)
- [「イベントロギング変数の設定」](#)
- [「一般的な製品動作変数の設定」](#)
- [「アクセス制御変数の設定」](#)
- [「自動バックアップ変数の設定」](#)
- [「アクセス制御情報の再作成」](#)
- [「イベント通知設定の定義」](#)
- [「ライブラリハードウェアの構成または更新」](#)
- [「システムイベントの電子メール通知の登録」](#)

ACSL S 機能構成メニューへのアクセス

ACSL S をインストールまたはアップグレードすると、ほとんどのユーザー環境ではその環境に基づいてシステムのデフォルトがすでに設定されています。ただし、必要に応じて、オプション 1-8 を使用してこれらの設定を変更できます。オプション 1 から 7 を使用すると、動的変数および静的変数を変更できます。静的変数を有効にするには、ACSL S を再起動する必要があります。

ライブラリハードウェアを追加および定義するには、オプション 8 を使用します。このオプションは、ACSLS をはじめてインストールした場合、または、たとえば、新しいライブラリを追加する必要がある場合に使用します。

各プロンプトでヘルプを表示するには、? を選択します。

`acsss_config` を起動するには、次の手順に従います。

1. CDE ログインに到達するまで CDE を終了します。
2. `acsss` としてログインします。
3. 構成スクリプトを実行します。

`acsss_config`

「*ACSLS feature configuration*」画面が表示されます。

すべての LMU、LSM、およびトランスポートが完全に構成され、オンにされて準備ができている状態ではない場合、構成スクリプト `acsss_config` は失敗します。

次のメニューが表示されます。

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:

動的変数および静的変数の確認および変更

オプション 1 - 8 を使用すると、次の動的変数および静的変数を変更できます。

CSI チューニング変数の設定

CSI (Client System Interface) は、ACSL S とほかのサーバーのクライアントの間の通信を処理します。これは、各クライアントの通信を処理する方法を設定します。1 つのクライアントとの通信が失われても、ほかのクライアントは影響を受けず、それらの通信は中断せずに続行されます。複数の CSI を ACSLS で実行できます。

各プロンプトでヘルプを表示するには、? を選択します。

オプション 1 では、次の変数を設定または変更できます。

- **CSI_CONNECT_AGETIME**

プロンプト: *Maximum age in seconds of pending requests in CSI request queue [172800]*

動的変数。この設定は、応答していないクライアントリクエストを ACSLS が待機する期間を指定します。

有効なエントリは 600 から 315360000 秒です。デフォルトは 172800 秒です。

- **CSI_RETRY_TIMEOUT**

プロンプト: *Number of seconds between successive retries [4]*

デフォルトは 4 秒です。

動的変数。このオプションは、CSI がネットワーク接続を確立するための各試行の間に待機する最小の時間を秒単位で指定します。

CSC と CSI の間でタイミングの問題が発生する場合は、この値を変更してください。

- **CSI_RETRY_TRIES**

プロンプト: *Number of retries for the CSI before a timeout condition occurs [5].*

動的変数。このオプションは、CSI がメッセージを送信するために行う試行の数を指定します。指定された再試行回数内で接続を確立できない場合は、保留中のメッセージが破棄されます。デフォルトでは 5 回再試行します。

- `CSI_TCP_RPCSERVICE`

プロンプト: *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].*

静的オプション。このオプションは CSI が TCP RPC サーバーとして動作する場合に指定します。デフォルトは true です。

- `CSI_UDP_RPCSERVICE`

プロンプト: *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].*

静的オプション。このオプションは CSI が UDP RPC サーバーとして動作する場合に指定します。クライアントが UDP 経由で ACSLS と通信するには、デフォルトを受け入れる必要があります。デフォルトは TRUE です。

このオプションを有効にするには、ACSLS を再起動する必要があります。

- `CSI_MULTI_HOMED_CL`

プロンプト: *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]*

このオプションを使用すると、パケットヘッダーの IP アドレスがパケットの送信元の IP アドレスと同じではない受信パケットをサーバー上の CSI が処理できます。クライアントが ACSLS に送信するリクエストパケットのアドレスは、クライアントアプリケーションが「hostname」の呼び出しを行い、それに続けて `get hostbyname` 検索を行うことによって導出されます。これにより、ローカルホストの IP アドレスがリクエストパケットに設定されます。

通常、これは機能します。ただし、例外があります。

- ほとんどの場合、ACSLS のクライアントには、単一の固定されたホストアドレスがあります。ただし、クライアントシステムに複数のネットワークインタフェースがあり、複数の IP ポートのいずれかを使用して ACSLS にリクエストを送信することがあります。
- 外部からアクセスできない内部のプライベート IP アドレスを持つ NAT の背後にクライアントシステムがある可能性があります。

- クライアントが VPN 経由で ACSLS サーバーに接続されていて、クライアントの IP アドレスが ACSLS が応答する必要がある IP アドレスではありません。

これらの場合、ACSL S はクライアントアプリケーションでローカルホストのアドレス以外の IP アドレスに応答する必要があります。CSI_MULTI_HOMED_CL 変数は、このような状況に対処するために設計されています。有効なセクションは次のとおりです。

- FALSE

これは、ACSL S 操作のデフォルト設定です。この場合、ACSL S はクライアントに応答を返すときに、常にローカルホストの IP アドレスに応答します。

- TRUE

ACSL S はリクエストパケットの IP アドレスを無視します。代わりに、ACSL S は、クライアントに応答するときに、ACSAPI リクエストパケットに関連付けられた受信 RPC データグラムの IP ヘッダーにあるアドレスを使用します。つまり、リクエストの送信元の IP アドレスです。アクセス制御が有効になっている場合は、この代替 IP アドレスを internet.addresses ファイルに指定する必要があります。

- CSI_USE_PORTMAPPER

プロンプト: *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].*

デフォルトは *IF_DUAL_LAN_NOT_ENABLED* です。静的オプション。有効なオプションは次のとおりです。

- ALWAYS - CSI がクライアントにメッセージを送信できない場合、ポートマッパーが常に照会されます。
- NEVER - CSI がクライアントにメッセージを送信できない場合、ポートマッパーが照会されることはありません。クライアントがポートマッパーをサポートしていない場合は、このオプションを選択します。
- *IF_DUAL_LAN_NOT_ENABLED* - デュアル LAN のサポートが有効になっていない場合にのみ、ポートマッパーが照会されます。デュアル LAN のサポートが有効になっている場合は、クライアントがポートマッパーをサポートしていないと見なされます。このオプションを選択すると、製品動作の下位互換性が確保されます。

- SURROGATE_PROCESSES

プロンプト: *Number of ACSSURR persistent processes that should be started [0]:*

有効なエントリ: 0 または 1。LM ゲートウェイがインストールされていない場合は、0 を入力します。

- **SURROGATE_PORT**

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

この変数は、ライブラリ管理 (LM) ゲートウェイにのみ適用されます。有効なエントリ: 50300 - 99999。

- **SURROGATE_TIMEOUT**

プロンプト: *Number of seconds to wait for data packets to be read on surrogate/gateway sockets:*

この変数は、ライブラリ管理 (LM) ゲートウェイにのみ適用されます。有効なエントリ: 1 - 600。

- **SURROGATE_QUEUE_AGE**

プロンプト: *Number of minutes to wait before deleting a stale queue entry [5].*

この変数は、ライブラリ管理 (LM) ゲートウェイにのみ適用されます。有効なエントリ: 5 - 60。

- **START_CSCI_PROCESS**

プロンプト: *Automatically start CSCI at ACSLS startup (TRUE/FALSE) [FALSE]:*

この変数は、ACSLs の起動時に CSCI プロセスを自動的に起動するかどうかを指定します。デフォルト値の FALSE は、CSCI が ACSLS とともに開始されないことを意味します。ACSLs とともに CSCI プロセスを起動する場合は、この変数に TRUE を設定します。

- **CSI_FIREWALL_SECURE**

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

動的変数。この設定は ACSLS への受信リクエストのための単一のユーザー定義のポートのファイアウォールセキュリティの定義を有効または無効にします。値は次のとおりです。

- False - ACSLS サーバー上のポートを制限しない場合。

- True - これはデフォルトです。ACSL S サーバーはセキュリティー保護されたファイアウォールの背後で動作します。

- **CSI_INET_PORT**

プロンプト: *Port number used by the CSI to receive incoming ACSLS requests [30031].*

この変数は、1 つ以上のクライアントからの受信 TCP リクエストのために CSI によって使用される単一のユーザー定義のポートを指定します。

この変数は、**CSI_FIREWALL_SECURE** に True を設定することによって、ファイアウォールでセキュリティー保護された CSI が有効にされている場合にのみ使用します。

デフォルト: 30031

有効なエントリ: 1024 - 65535 (ただし、50003 以外)

- **CSI_FAILED_RESPONSE_LIMIT**

プロンプト: *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]*

ほとんどのリクエストの場合、CSI がクライアントに 1 つの応答を送信できないと、そのリクエストの残りのすべての応答が削除されます。これにより、応答を受信できないクライアントに応答を送信しようとして、CSI の動作が止まってしまうことを防ぎます。

監査、挿入、取り出し、またはイベント登録のリクエストの場合、CSI は、最初に失敗したあとも、リクエストへの応答を返すことを試み続けます。**CSI_FAILED_RESPONSE_LIMIT** は、CSI がリクエストへの残りのすべての応答を削除する前に、リクエストに応答を返す試行が失敗する数を指定します。これにより、通信に問題があるか、クライアントがアクティブでなくなったときに、CSI の可用性が保護されます。

これらの値は、クライアントと CSI の間でタイミングの問題が発生している場合にのみ変更してください。詳細は、『ACSL S 管理者ガイド』を参照してください。

1 - 9 の数値を入力します。

イベントロギング変数の設定

各プロンプトでヘルプを表示するには、? を選択します。

オプション 2 では、次のイベントログを設定または変更できます。

- **EVENT_FILE_NUMBER**

プロンプト: *Number of Event Log files to retain [9]:*

このオプションは、保持する追加のイベントログファイルの数を指定します。このオプションを有効にして、現在のイベントログファイルのサイズがしきい値のサイズに達すると、ログが別のファイルに自動的にコピーされます。指定された数のファイルに達すると、もっとも古いファイルのデータが上書きされます。

追加のイベントログファイルを保持しない場合は **0** を入力します。単一のイベントログファイルは手動で切り捨てられるまで拡大し続けます。この場合、イベントログファイルのサイズがしきい値のサイズに達すると、イベントログが名前変更または削除されるまで、要求外の「Event log full」というメッセージが周期的に表示されます。ACSL S をインストールまたはアップグレードしたときのデフォルト値は 9 個です。

この機能を有効にして、保持するログファイルの数を指定するには、**1** から **9** の間の数値を入力します。イベントログがしきい値のサイズに達すると、データが *event0.log* ファイルに移動されます。次回、しきい値のサイズに達すると、*event0.log* のデータが *event1.log* に移動され、イベントログのデータが *event0.log* に移動されます。このサイクルは指定された保持するファイルの数に達するまで継続され、その時点でもっとも古いファイルからデータが破棄されます。

- **LOG_PATH**

プロンプト: *Changes to the logging directory will not take effect until the product is restarted. What directory should logging information be in [\$ACS_HOME/log]:*

ログファイルが配置されるディレクトリを選択します。インストールされた状態では、ACSL S は *\$ACS_HOME/log* ディレクトリに情報を記録します。通常の使用では、この変数の値は変更されません。*\$ACS_HOME* が含まれているファイルシステムにディスク容量の問題がある場合は、代替パスを指定できます。指定するパスは絶対パス (/ または *\$ACS_HOME* で始まるパス) にする必要があります。

この変数を有効にするには、ACSL S 製品を再起動する必要があります。

- **LOG_SIZE**

プロンプト: *Maximum library server event log size in Kilobytes (=1000 bytes).*

このオプションは、イベントログのしきい値サイズを K バイト (1,000 バイト) 単位で指定します。負ではない数値を入力します。ACSL S をインストールまたはアップグレードしたときのデフォルト値は 500 です。

有効なエントリ: 32-2147483

- **TIME_FORMAT**

プロンプト: *Date/time format for all logs [%Y-%m-%d%H:%M:%S]:*

このオプションは、イベントログおよびトレースログの日付と時間の情報を出力するために使用する形式を指定します。この形式は、C 言語の機能 *strftime* によって使用される形式と同じです。

- **LM_RP_TRAIL**

プロンプト: *Should library manager request process auditing be turned on? [TRUE]:*

TRUE を設定した場合、このオプションはすべての ACSLS リクエストの監査トレールを有効にします。各リクエストは、タイムスタンプ、およびリクエストしたクライアントまたはユーザーインタフェースの名前とともに記録されます。その結果のログは、*rpTrail.log* ファイルにあります。

- **RP_TRAIL_LOG_SIZE**

プロンプト: *Maximum size in Kbytes for the rpTrail Log [1000]:*

このオプションは、*rpTrail* ログのしきい値のサイズを K バイト単位で指定します。負数は入力できません。ACSL S をインストールまたはアップグレードしたときのデフォルト値は 1000K バイト (1M バイト) です。*rpTrail.log* がこのサイズを超えると、圧縮およびアーカイブされ、後続のリクエストの監査のための新しい *rpTrail.log* が作成されます。

10 - 10000 の値を入力します。

- **RP_TRAIL_FILE_NUM**

プロンプト: *Number of rpTrail archive files to retain [10]:*

このオプションは、アーカイブされた *rpTrail* ログファイルを保持する数を指定します。現在の *rpTrail.log* のサイズがしきい値のサイズを超えると、ログは自動的に圧縮およびアーカイブされ、圧縮された時点で取得されたタイムスタンプ (yyyy-mm-dd_hh:mm) が名前に付けられます。アーカイブファイルは、「*log_archives*」ディレクトリに保存されま

す。アーカイブログが指定された数に達して、新しいログが追加されると、もっとも古いファイルが削除されます。

1 - 999 個のアーカイブログファイルを保持できます。

- `RP_TRAIL_DIAG`

プロンプト: *Log diagnostic time stamps for internal request processing? [FALSE]:*

このオプションは、監査トレールに診断情報を含めるかどうかを指定します。

TRUE を設定した場合、「QUEUED AT」、「FORKED/WRITTEN TO AT」、および「FINAL RESPONSE SENT TO <recipient> AT」のタイムスタンプがトレールログに含まれます。

- `XAPI_LOG_SIZE`

プロンプト: *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.]*

このオプションは、M バイト (ここでは「1048576 バイト」と定義) で表される、XAPI ログのしきい値サイズを指定します。負でない数を入力します。20 がこのオプションのデフォルト値です。

この変数を有効にするには、XAPI サーバーを再起動する必要があります。

- `XAPI_LOG_FILE_NUM`

プロンプト: *Number of XAPI Log archive files to retain [10].*

このオプションは、保持するアーカイブ XAPI ログファイルの数を指定します。現在の `vlog.file` サイズがしきい値サイズを超えると、0 から `n` の接尾辞が付加されてログファイルの名前が変更します。0 はもっとも新しいファイルで、`n` はもっとも古いファイルです。アーカイブファイルは、`API_WORK_PATH` ディレクトリに保存されます。指定された数のアーカイブログに達すると、新しいファイルがアーカイブディレクトリに追加されるごとにもっとも古いファイルがそのディレクトリから削除されます。1 つ以上 99 以下のアーカイブファイルを保持できます。1 から 99 の数値を入力して、保持するアーカイブログファイルの数を指定します。

この変数を有効にするには、XAPI サーバーを再起動する必要があります。

- `XAPI_TRACE_SIZE`

プロンプト: *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].*

このオプションは、M バイト (ここでは「1048576 バイト」と定義) で表される、XAPI トレースのしきい値サイズを指定します。負でない数を入力します。50 がこのオプションのデフォルト値です。

この変数を有効にするには、XAPI サーバーを再起動する必要があります。

- `XAPI_TRACE_FILE_NUM`

プロンプト: *Number of XAPI Trace archive files to retain.*

このオプションは、保持するアーカイブ XAPI トレースファイルの数を指定します。現在の `vtrace.file` サイズがしきい値サイズを超えると、0 から `n` の接尾辞が付加されてトレースファイルの名前が変更します。0 はもっとも新しいファイルで、`n` はもっとも古いファイルです。アーカイブファイルは、`XAPI_WORK_PATH` ディレクトリに保存されます。指定された数のアーカイブログに達すると、新しいファイルがアーカイブディレクトリに追加されるごとに、もっとも古いファイルがそのディレクトリから削除されます。1 つ以上 99 以下のアーカイブファイルを保持できます。1 から 99 の数値を入力して、保持するアーカイブログファイルの数を指定します。

この変数を有効にするには、XAPI サーバーを再起動する必要があります。

一般的な製品動作変数の設定

各プロンプトでヘルプを表示するには、? を選択します。

オプション 3 では、次の変数を設定または変更できます。

- `LIB_VOL_STATS`

プロンプト: *Enable Library Volume Statistics Gathering: (ON / OFF) [OFF]:*

このオプションは、ライブラリボリューム統計 (LVSTATS) 情報を挿入、取り出し、マウント解除、マウント、および監査の操作から収集するかどうかを指定します。LVSTATS の収集および報告セッションを開始するには ON を入力します。LVSTATS の収集および報告セッションを終了するには OFF を入力します。セッションが開始または終了したときに、適切なメッセージがログファイル `acsss_stats.log` に出力されます。このオプションを適切に使用すると、前記で一覧された個別の各操作がログファイル内で囲んで示されます。

- `VOL_STATS_FILE_NUM`

プロンプト: *Number of acsss_stats log files to retain [9]:*

このオプションは、追加の *acsss_stats* ログファイルを保持する数を指定します。このオプションを有効にして、現在の *acsss_stats* ログファイルのサイズがしきい値のサイズに達すると、ログが別のファイルに自動的にコピーされます。指定された数のファイルに達すると、もっとも古いファイルのデータが上書きされます。

追加のログファイルを保持しない場合は 0 を入力します。単一の *acsss_stats* ログファイルは、手動で切り捨てられるまで拡大し続けます。この場合、*acsss_stats* ログファイルのサイズがしきい値のサイズに達すると、*acsss_stats* ログが名前変更または削除されるまで、要求外の「*acsss_stats log full*」というメッセージがイベントログに周期的に表示されます。ACSL S をインストールまたはアップグレードしたときのデフォルト値は 9 個です。

保持するログファイルの数を指定するには、1 から 9 までの数値を入力します。*acsss_stats* ログがしきい値のサイズに達すると、データが *vol_stats0.log* ファイルに移動されます。次回、しきい値のサイズに達すると、*vol_stats0.log* のデータが *vol_stats1.log* に移動され、*acsss_stats* ログのデータが *vol_stats0.log* に移動されます。このサイクルは指定された保持するファイルの数に達するまで継続され、その時点でもっとも古いファイルからデータが破棄されます。

- *VOL_STATS_FILE_SIZE*

プロンプト: *Maximum library server access_stats log size in Kbytes (=1000 bytes) [500]:*

このオプションは、*acsss_stats* ログのしきい値のサイズを K バイト単位 (ここでは「1000 バイト」として定義します) で指定します。32 以上の値を入力します。ACSL S をインストールまたはアップグレードしたときのこのオプションのサイズのデフォルト値は 500 (K バイト) です。

有効なエントリ: 32-10000

- *UNIFORM_CLEAN_USE*

プロンプト: *Select cleaning cartridge ordering method [VOLID_SORT]:*

有効なオプションは、次のとおりです。

- *VOLID_SORT* - クリーニングカートリッジをボリューム識別子によって順序付けします。この場合、次のクリーニングカートリッジに移動する前に、クリーニングカートリッジが使用限度の回数まで使用されます。このオプションを選択した場合、ACSL S は以前のリリー

スの ACSLS ソフトウェアと同じ順序でクリーニングカートリッジを使用して戻します。これは ACSLS をインストールまたはアップデートしたときのデフォルト値です。

- *LEAST_USED* - クリーニングカートリッジを使用回数によって順序付けします。このオプションを選択した場合、ACSLS は使用回数の逆順でボリュームリストをソートし、もっとも使用回数が少ないカートリッジが最初になるように戻します。これにより、クリーニングカートリッジの使用回数が均一に分散されます。
- *MOST_CAPACITY* - クリーニングカートリッジを残りの使用回数によって順序付けします。このオプションを選択した場合、ACSLS はクリーニングカートリッジの残りの使用回数に基づいてボリュームリストをソートし、残りの使用回数をもっとも多いカートリッジが最初になるように戻します。これにより、すべてのクリーニングカートリッジがほとんど同じ回数使用されます。

- *AUTO_CLEAN*

プロンプト: *Enable Transport Auto Clean Option (TRUE/FALSE) [TRUE]:*

有効なオプションは、次のとおりです。

- *TRUE* - トランスポートをクリーニングする必要がある場合、ACSLS は次のマウントの前に、トランスポートを自動的にクリーニングします。

SCSI またはファイバで接続されたライブラリ内のドライブのクリーニング操作は、ACSLS ではなくライブラリファームウェアによって管理されます。LSM コントロールパネルからドライブのクリーニングを有効にしてください。そうしない場合は、必要に応じて、ACSLS からクリーニングカートリッジを手動コマンドによって定期的にマウントできません。

- *FALSE* - 自動クリーニングを有効にしません

- *AUTO_START*

プロンプト: *Library Server startup state (RUN/IDLE) [RUN]:*

このオプションは、サーバーソフトウェアの開始時に回復が完了したあとの ACSLS の初期状態を指定します。有効なオプションは、次のとおりです。

- *RUN* - ユーザーリクエストは回復が完了した直後に処理されます。
- *IDLE* - 回復が完了した直後にユーザーリクエストが処理されないようにします。

このオプションに *IDLE* を設定すると、ユーザーにサーバーへのアクセスを許可する前に、デバイスの状態の変更または運用上のアクティビティを実行する必要がある場合に役に立つことがあります。

- **MAX_ACSMT**

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

有効なエントリは 1 - 5 です。

Oracle は、初期構成時にデフォルト値を受け入れ、必要に応じて値を変更することをお勧めします。この値を増やすと、パフォーマンスが向上することがあります。(8 個を超える LSM がある) 大規模な構成の場合、マウント/マウント解除のパフォーマンスを改善するために、この値を増やす必要があることがあります。

注意:

この数値の設定が高すぎる場合、ACSL S が正常に起動しないことがあります。この場合、この数値を少なくするか、ユーザー当たり許可される最大のプロセス数を増やします。

デフォルトは 2 です。

- **ENABLE_ACSQY**

プロンプト: *Enable QUERY persistent processes (TRUE/FALSE) [TRUE]:*

このオプションを使用すると、1 つ以上の永続的なプロセスとしてクエリーを実行できます。クエリーは永続的なプロセスまたはリクエストプロセスとして実行できます。初期構成時にデフォルト値を受け入れ、必要に応じて値を変更します。この値を増やすと、パフォーマンスが向上することがあります。有効なオプションは、次のとおりです。

- TRUE - 永続的なクエリープロセスを有効にします。

注意:

最小限の構成のシステムで、10 個を超えるクエリーの永続的なプロセスを使用すると、システムリソースが使用し尽くされて、ACSL S が起動できないことがあります。

- FALSE - 永続的なクエリープロセスを無効にします。クエリーはリクエストプロセスとして起動されません。

- **MAX_ACSQY**

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

このオプションは、永続的なクエリープロセスを作成する数を指定します。このオプションは、前述の手順で有効にされている場合にのみ有効です。有効な数値は 1 - 5 です。

Oracle は、初期構成時にデフォルト値 (2) を受け入れ、必要に応じて値を変更することをお勧めします。(8 個を超える LSM の) 大規模な構成の場合、パフォーマンスを向上させるためにこの値を増やす必要があることがあります。

注:

この数値の設定が高すぎる場合、ACSLs が正常に起動できないことがあります。この数値を少なくするか、ユーザー当たり許可される最大のプロセス数を増やします。

このオプションを有効にするには、ACSLs を再起動する必要があります。

- **MAX_ACS_PROCESSES**

プロンプト: *Changes to the maximum number of ACSLS processes will not take effect until the product is restarted. Number of ACSLS processes [40]:*

有効な数値は 32 - 100 です。

一時的なプロセスは、*mount*、*dismount*、*lock*、*unlock*、*clear_lock*、および *query_lock* を除くすべてのリクエストに対応できます。デフォルトの 40 個のプロセスは、非常に大規模な構成を除き、すべての ACSLS プロセスに対して機能します。デフォルトは 40 です。

注:

この値を変更する前に、サポートにアドバイスを求めることをお勧めします。

このオプションを有効にするには、ACSLs を再起動する必要があります。

- **TRACE_ENTER**

プロンプト: *Enable automatic enter tracing facility (TRUE/FALSE) [FALSE]:*

このオプションは、自動挿入操作の結果をイベントログに書き込むかどうかを指定します。有効なオプションは、次のとおりです。

- TRUE - 各自動挿入操作の終わりに、メッセージがイベントログに書き込まれます。これは、自動挿入操作中にボリュームが挿入されなかった理由を判断するための唯一の方法です。
- FALSE - この機能を無効にし、イベントログに書き込まれるメッセージの数を最小限に抑えるために役立ちます。

- **TRACE_VOLUME**

プロンプト: *Enable volume tracing facility (TRUE/FALSE) [FALSE]:*

このオプションは、データベースにボリュームを追加/削除するときに、要求外のメッセージを表示するかどうかを指定します。有効なオプションは、次のとおりです。

- TRUE - データベースにボリュームが追加または削除されるたびに、要求外のメッセージが表示されます。このメッセージが生成される可能性がある操作は、*audit*、*mount*、*dismount*、*enter*、*eject*、*recover*、および *vary* です。
- FALSE - この機能を無効にし、イベントログに書き込まれるメッセージの数を最小限に抑えるために役立ちます。

- **ABSENT_VOLUME_RETENTION_PERIOD**

- プロンプト: *Number of days to retain volumes that are identified as ABSENT or EJECTED in the database. [5]:*

動的変数。不在カートリッジまたは取り出し済みのカートリッジをデータベースに保持する日数 (0-999) を設定します。ボリュームを再挿入しないかぎり、ACSLs は、保持期間が期限切れになったときに、データベースからボリュームを自動的に削除します。

ボリュームを削除しない場合は、999 を入力します。ボリュームをすぐに削除するには、0 を入力します。

注意:

長い保持期間を設定すると、多数の不在ボリュームまたは取り出し済みのボリュームが保持されることがあります。これにより、データベースでより多くの領域が使用され、いっぱいになるのが早くなります。

- **ENABLE_STATUS_VOLUME_ABSENT**

プロンプト: *Report ABSENT and EJECTED volumes to ACSAPI clients. [FALSE]:*

有効なオプションは、次のとおりです。

- TRUE - 不在ボリュームおよび取り出し済みのボリュームを ACSAPI クライアントに報告します。
- FALSE - 不在ボリュームおよび取り出し済みのボリュームは ACSAPI クライアントに報告されません (以前のバージョンの ACSLS の動作と同様)。

- **ENABLE_STATUS_VOLUME_MISSING**

プロンプト: *Status for MISSING volumes reported to ACSAPI clients: TRUE=MISSING. FALSE=IN TRANSIT [FALSE]:*

このオプションは、見つからないボリュームについてステータスコードを ACSAPI クライアントに返すように設定します。有効なオプションは、次のとおりです。

- TRUE - 見つからないカートリッジを ACSAPI クライアントに報告します。
- FALSE - 見つからないボリュームは ACSAPI クライアントに報告されません (以前のバージョンの ACSLS の動作と同様)。

****ENABLE_INIT_ACSLM は TRUE である必要があります****

GUI および論理ライブラリがレガシー ACSLS プロセスとやり取りできるようにするには、この変数に TRUE を設定する必要があります。[TRUE]:

レガシー ACSLS のプロセスは、プロセス間通信に UNIX ベースのソケットのみを使用しました。この変数は、GUI および論理ライブラリをサポートするために、INET ベースのソケット通信を有効にします。ACSL 8+ 機能を有効にするには、ENABLE_INIT_ACSLM が true である必要があります。変更を有効にするには、ACSL 製品を再起動する必要があります。

- ALPHANUM_VOL_RANGES

プロンプト: *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]:

コマンドおよびユーティリティーで英数字のボリューム範囲のサポートをアクティブにするかどうかを指定します。有効なオプションは、次のとおりです。

- FALSE (デフォルト) - ボリューム範囲を指定する場合、vol_id の範囲の最初の文字が同じである必要があります、最後の可変文字が数字である必要があります。例: AAA000-AAA999
- TRUE - 英数字のボリューム範囲にすべての有効なボリューム ID が含まれている場合は、範囲に有効な vol_id を指定できます。有効なボリューム範囲は、数値 (0-9)、文字 (A-Z)、ドル記号 (\$)、シャープ記号 (#)、および空白 (先頭および末尾) の組み合わせで構成されます。

例: A1Z27BC-G\$123R

- EJECT_RESPONSE_ON_CAP_FULL

プロンプト: *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].*

このオプションは、CAP が使用されているときに、取り出しプロセスが中間応答を送信するか、MAX_ID のボリュームが取り出されるまで待つかどうかを指定します。MAX_ID 個のセルよりも多い CAP を持つ構成では、MAX_ID 個のボリュームが取り出されたときに中間応答が生成されるため、これは動作に影響しません。この設定は `cmd_proc` に影響を与えず、CAP が使用されている場合、`cmd_proc` は常に応答を受け取ります。

注意:

この変数の値を変更すると、中間取り出し応答に MAX_ID のボリュームが常に含まれていることを予期している ACSAPI クライアントに影響を与えます。

- **MOUNT_RETRY_DELAY**

プロンプト: *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]:*

ライブラリがビジー状態であるか、一時的に使用できないため、マウントおよびマウント解除のリクエストが失敗した場合、それらは自動的にキューに入れられます。リクエストが定期的に再試行されるか、ライブラリおよびドライブの可用性が再チェックされます。この変数は、このシーケンスの各再試行の間の時間間隔を分単位で指定します。

- **MOUNT_RETRY_TIME_LIMIT**

プロンプト: *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]:*

ライブラリが一時的に使用できない場合に、マウントおよびマウント解除のリクエストが失敗しないようにするために、それらは自動的にキューに入れられます。リクエストが定期的に再試行されるか、ライブラリおよびドライブの可用性が再チェックされます。この変数は、ACSL S がマウントまたはマウント解除のリクエストをキューに入れることを継続する最大時間です。

- **AUTO_CLEAN_RETRY_LIMIT**

プロンプト: *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].*

ドライブのクリーニングが必要になり、自動クリーニングが有効である場合、ACSL Sはそのドライブへの次のマウントの前に、ドライブをクリーンアップしようとします。クリーニングカートリッジが最大使用回数を超えるため、またはそれ以外の理由で使用できないために、クリーニング操作が失敗した場合、ACSL Sは別のクリーニングカートリッジを選択し、ドライブをクリーニングするためにマウントしようとします。

`AUTO_CLEAN_RETRY_LIMIT` は、リクエストされたデータカートリッジを続けてマウントする前に、ACSL Sがクリーニング操作を再試行する回数を制限します。

- `XAPI_PORT`

プロンプト: *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].*

このオプションは、クライアントからの受信 TCP リクエストに XAPI サーバーで使用されるポートを指定します。XAPI サーバーによって使用されるポートを定義する、1024 から 65535 の数値を入力します。ポート 50003 は指定しないでください。

- `XAPI_WORK_PATH`

プロンプト: *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]*

XAPI サーバーの作業ファイルが置かれるディレクトリを選択します。インストールされると、XAPI サーバーは、`$ACS_HOME/log/xapi` ディレクトリに情報を記録します。通常の使用では、この変数の値は変更されません。`$ACS_HOME` が含まれているファイルシステムでディスク容量の問題がある場合は、代替パスを指定できます。指定するパスは絶対パス（/または `$ACS_HOME` で始まるパス）にする必要があります。

この変数を有効にするには、XAPI サーバーを再起動する必要があります。

- `LIMIT_CAP_CONCURRENT_MOVES`

プロンプト: *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).*

各取り出しまたは挿入の操作で、ACSL S は複数のカートリッジを CAP に (または CAP から) 並行して移動します。(デフォルトでは、各 CAP で並行して 4 つが移動されます。)複数のライブラリが接続された ACS (ライブラリコンプレックス) で、多くの CAP が取り出しおよび挿入に同時に使用されている場合は、これにより、マウントおよびマウント解除を含むほかのリクエストのために使用可能なライブラリリソースを制限できます。

多数の CAP を挿入および取り出しに同時に使用するとき、マウント、マウント解除などのためにライブラリリソースを確保しておくには、`LIMIT_CAP_CONCURRENT_MOVES` に TRUE を設定します。

- `xapi_startup_file`

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

このオプションは、XAPI 起動ファイルの名前を指定します。このファイルは、`XAPI_WORK_PATH` ディレクトリに存在し、XAPI 起動パラメータを含みます。

この変数を有効にするには、XAPI サーバーを再起動する必要があります。

- `BULK_CAP_EJECT_HANDLE`

プロンプト: *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)*

取り出し中に各 Bulk CAP マガジン内の 1 つのスロットを空のままにするには、`BULK_CAP_EJECT_HANDLE` に TRUE を設定します。空のスロットはハンドルとして使用できます。レールの高さに応じて、下部スロットまたは上部スロットのいずれかが空のままになります。

- `ENTER_CLOSE_TO_DRIVES`

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

`ENTER_CLOSE_TO_DRIVES` が TRUE の場合、ACSL S は SL8500 に挿入されているカートリッジをテープドライブに近いストレージセルに移動します。これにより、その後、ドライブへのカートリッジのマウントが容易になります。

`ENTER_CLOSE_TO_DRIVES` が `FALSE` の場合、`ACSLs` は挿入されているカートリッジの新しいホームセルをラウンドロビン方式で割り当て、各パネルにホームセルを順番に割り当てます。

- `DISMOUNT_AWAY_FROM_DRIVES`

プロンプト: *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)*

`DISMOUNT_AWAY_FROM_DRIVES` が `TRUE` の場合、`ACSLs` は SL8500 内のマウント解除されている位置の新しいホームセルにフロートされているカートリッジをテープドライブから離れたストレージセルに移動します。これにより、テープドライブの近くに新しいカートリッジのためのスペースが残されます。

`DISMOUNT_AWAY_FROM_DRIVES` が `FALSE` の場合、`ACSLs` はマウント解除されている位置にフロートされているカートリッジの新しいホームセルをラウンドロビン方式で割り当て、各パネルの空のセルを順番に割り当てます。

- `XAPI_TAPEPLEX_NAME`

プロンプト: *Changes to the XAPI Tapeplex name will not take effect until the xapi server is restarted.*

Name of the XAPI Tapeplex []

このオプションは、XAPI Tapeplex の名前を指定します。この変数を有効にするには、XAPI サーバーを再起動する必要があります。

長さが 8 を超えないテキストを入力してください。

アクセス制御変数の設定

各プロンプトでヘルプを表示するには、`?` を選択します。

オプション 4 では、次のアクセス制御変数を設定または変更できます。

- `AC_CMD_ACCESS`

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

コマンドのアクセス制御がアクティブであるかどうかを指定します。これにより、各ネットワーククライアントが実行できる `ACSLs` コマンドを制御できます。

デフォルトの `false` は、アクセス制御を無効にします。

注:

この変更を有効にするには、ACSLS を停止してから再起動する必要があります。

- `AC_CMD_DEFAULT`

プロンプト: *Default access for commands (ACCESS/NOACCESS) [ACCESS].*

関連するアクセス制御リストがない場合の、コマンドへのデフォルトのアクセスを指定します。

デフォルトの `ACCESS` は、アクセス制御ファイルで提供されている構成情報を通じて拒否されていないかぎり、コマンドへのアクセスを許可します。

- `AC_VOL_ACCESS`

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

ボリュームのアクセス制御がアクティブであるかどうかを指定します。これにより、特定のボリュームにアクセスできる ACSLS クライアントを制御できます。このオプションを有効にするには、ACSLS を再起動する必要があります。

デフォルトの `false` は、ボリュームのアクセス制御を無効にします。

注:

この変更を有効にするには、ACSLS を停止してから再起動する必要があります。

- `AC_VOL_DEFAULT`

プロンプト: *Default access for volumes (ACCESS/NOACCESS) [NOACCESS].*

関連するアクセス制御リストがない場合の、ボリュームへのデフォルトのアクセスを指定します。

デフォルトの `NOACCESS` は、アクセス制御ファイルで提供されている構成情報を通じて付与されていないかぎり、ボリュームへのアクセスを許可しません。

- `AC_LOG_ACCESS`

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

コマンドまたはボリュームへのアクセスが拒否されたときに、メッセージをイベントログに記録するかどうかを指定します。コマンドまたはボリューム ID が記録され、このオプションを有効にした場合は、アクセス ID およびホスト ID が記録されます。

自動バックアップ変数の設定

各プロンプトでヘルプを表示するには、? を選択します。

オプション 5 では、次の自動バックアップ変数を設定または変更できます。

- プロンプト: *Would you like to modify the automatic backup settings? (y or n)*

変数: なし。自動バックアップ設定は crontab に保存されます。これらの設定を変更するには、*acsss_config* を使用します。

有効なオプションは、次のとおりです。

- Y - 「yes」の場合は、毎日バックアップするかどうかを尋ねられます。「yes」の場合は、*HH:MM* 形式で時間を入力する必要があります。

「no」の場合は、バックアップする曜日を選択する必要があります。*HH:MM* 形式で時間を入力します。

- N - SDM で管理されたデータベースバックアップファイルの保持期間に関する次の質問に移動します。

- *RETENTION_PERIOD*

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

このオプションは、カートリッジが再利用に適格ではないとして指定される前に、ストレージカートリッジ上のデータを DSM が追跡する日数を指定します。有効な日数: 4-30

保持期間はバックアップを保持する期間です。トランザクションログファイルは各バックアップとともに保持され、バックアップパーティションの必要な領域が増加します。保持期間を選択するときは、サイトに固有の次の条件を考慮してください。

- ライブラリのサイズとアクティビティー
- 特定の期間に実行される自動および手動のバックアップの数

注:

`bdb.acsss` ユーティリティを使用して手動バックアップを実行する場合は、ローカルディスクへの自動バックアップも行われるため、バックアップパーティション内のバックアップの数が増加します。

保持期間は各バックアップの間隔より長くしてください。たとえば、保持期間が4日間である場合は、各バックアップの間隔が3日を超えないように、週に少なくとも3回、自動または手動のバックアップを取得してください。

- 。バックアップおよびトランザクションログファイルを保持する期間。保持期間を長くすると、バックアップパーティションに必要な領域が増えることに注意してください。

アクセス制御情報の再作成

オプション6では、アクセス制御が有効であるか、変更された場合に、アクセス制御テーブルを再作成します。

イベント通知設定の定義

各プロンプトでヘルプを表示するには、?を選択します。

オプション7では、イベント通知設定を定義できます。

- `CLIENT_CHECK_MESSAGE_COUNT`

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

このオプションは、ACSL S が登録されているクライアントがまだ生存していることを確認する前に、クライアントに送信されるイベント通知メッセージの数および最短のチェック間隔 (次の質問で設定します) を指定します。

有効なエントリ: 10 - 1000 メッセージ

- `CLIENT_CHECK_MIN_INTERVAL`

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

ACSL S は、登録されているクライアントがまだ生存しているかどうかを定期的にチェックします。このチェックは、最短間隔およびイベント通知メッセージのカウントを超えたときに実行されます。

有効なエントリ: 1 - 600 分

- `CLIENT_CHECK_RESPONSE_TIME`

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

登録されているクライアントがまだ生存しているかどうかを ACSLS が定期的にチェックする場合、クライアントは指定された応答時間内にチェック登録リクエストで応答する必要があります。クライアントがこの期間以内に応答しない場合、クライアントのイベント通知登録は取り消されます。

有効なエントリ: 5 - 300 秒

- `CLIENT_CHECK_POLLING_INTERVAL`

プロンプト: *Enter the polling interval for confirming Event Notification client registration (in minutes) [60]:*

ACSLs は、登録されているクライアントがまだ生存しているかどうかを定期的にチェックします。指定されたポーリング間隔で、ACSLs はこのポーリング間隔内に登録、登録解除、またはチェック登録のリクエストを発行していないすべてのクライアントにクライアントチェック応答を送信します。

有効なエントリ: 10 - 1440 分。

静的変数を設定した場合は、`acs1s` サービスを再起動する必要があります。`acs1s disable` に続けて `acs1s enable` を使用します。

動的変数の場合は、これを行う必要はありません。

ライブラリハードウェアの構成または更新

ライブラリハードウェアの初期作成、新しいライブラリの追加、または既存のライブラリの再構成を行うには、`acs1s_config` (オプション 8) を使用する必要があります。ライブラリの初期構成後に、`config` ユーティリティを使用して、LSM やテープドライブの動的な追加、再構成、および削除、ACS の追加や再構成、およびライブラリへのポート接続の追加を行うことができます。詳細および手順については、[83 ページの6章「ACSLs の動作を制御する変数の設定」](#)を参照してください。

詳細は、[5章「ライブラリハードウェアの設置と構成」](#)を参照してください。

システムイベントの電子メール通知の登録

管理責務を持つユーザーは、システムのブートイベントや ACSLS-HA システムクラスタのフェイルオーバーイベントなど、システムイベントの自動電子メール通知を登録してもかまいません。

そのようなイベントを登録するには、ユーザーは、次のディレクトリのそれぞれのファイルに電子メールアドレスを追加する必要があります。

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

ヘッダーの注釈の下にある単一の行に、目的の各受信者の電子メールアドレスを入れます。それ以降、システムがブートするか、HA クラスタがスタンバイノードにフェイルオーバーするたびに、登録済みの各ユーザーは電子メールで通知されます。

この機能では、sendmail サービスが ACSLS サーバーで有効になっていること、およびネットワークファイアウォールの制限でデータセンターからの電子メール通信が許可されていることを前提としています。

第7章 アクセス制御

アクセス制御では、次のものが提供されます。

- ボリュームアクセス制御では、ボリュームを1つのクライアントアプリケーションに割り当てることができます。そのクライアントのボリュームへのアクセスをほかのクライアントに許可できます。
- コマンドアクセス制御では、管理者が、特定の ACSLS コマンドを特定のクライアントに割り当てることができます。

ボリュームアクセス制御とコマンドアクセス制御はどちらも、ACSAPI 経由でリクエストを送信するクライアントアプリケーションのユーザーに適用されます。

アクセス制御では、`cmd_proc` または ACSLS GUI を使用してライブラリリクエストを送信する管理ユーザーによるアクセスは制限されません。

ボリュームアクセス制御

有効になっている場合、特定のユーザーによって所有されているボリュームはそのユーザーか、または信頼できるほかのユーザーからのみアクセスできます。

ACSLS でボリュームアクセス制御をはじめて構成する場合は、次の手順に従います。

1. ACSLS でボリュームアクセス制御を有効にします。
2. クライアントアプリケーションをユーザー名に関連付けます。
3. そのユーザーのボリュームにほかのどのユーザーがアクセスできるかを定義します。
4. ボリュームの所有権を確立します。

ボリュームアクセス制御の有効化

ACSLS でボリュームアクセス制御を有効にするには:

1. 構成ユーティリティ `acsss_config` を実行します。

メインメニューが表示されます。

2. 「Option 4 - Set Access Control Variables」を選択します。

各変数が一度に1つずつ表示され、その現在の設定が表示されます。

3. 現在の設定またはデフォルト設定を受け入れるには、**Enter** をクリックします。
4. ユーティリティーにメッセージ *Access control is active for volumes* が表示されたら、「**[TRUE]**」を選択して **Enter** をクリックします。
5. ユーティリティーにメッセージ *Default access for volumes [ACCESS/NOACCESS]. . .* が表示されたら、次のいずれかを選択します。
 - 目標が特定のユーザーに対してアクセスを禁止し、ほかのすべてのユーザーに対してアクセスを許可することである場合は、「**[ACCESS]**」を選択します。

これには、特定のユーザーを *users.ALL.disallow* ファイルまたは特定の *users.COMMAND.disallow* ファイルにリストしておく必要があります。「[ユーザーのボリュームへのアクセスが許可されるほかのユーザーの定義](#)」を参照してください。

- 目標が特定のユーザーに対してアクセスを許可し、ほかのすべてのユーザーに対してアクセスを禁止することである場合は、「**[NOACCESS]**」を選択します。

これには、特定のユーザーを *users.ALL.allow* ファイルまたは特定の *users.COMMAND.allow* ファイルにリストしておく必要があります。「[ユーザーのボリュームへのアクセスが許可されるほかのユーザーの定義](#)」を参照してください。

ボリュームへのアクセスが拒否されたインスタンスをログに記録する場合は、そのプロンプトに応答して「**[TRUE]**」を選択します。

ボリュームアクセスを有効または無効にした場合は常に、変更を有効にするために ACSLS を再起動する必要があります。

Associating a client identity with a user name

すべてのクライアントアプリケーションが ACSLS リクエストパケットでユーザー ID を渡すわけではありません。クライアントがユーザー名で識別されない場合は、ユーザー ID を割り当てることができます。

1. *access_control* 構成ディレクトリに移動します。

```
$ACS_HOME/data/external/access_control.
```

2. *internet.addresses* という名前でファイルを作成するか、または *internet.addresses.SAMPLE* ファイルをコピーします。

3. このファイル内に、クライアントごとのレコードを作成します。各レコードには、クライアント IP アドレスと、それに続く対応するユーザー名の少なくとも 2 つのフィールドが含まれます。コメントのための追加フィールドを含めることができます。

次の例に示すように、各フィールドをスペースまたはタブで区切ります。

```
192.0.2.1 ulyssis payroll department
```

クライアントとユーザーの関連付けは、使用しているクライアントアプリケーションと同じ数だけ作成できます。

- クライアントアプリケーションが ACSLS リクエストでユーザー名を渡す場合は、`internet.addresses` ファイルが指定されている IP アドレスでユーザー名を認証し、両方のフィールドがリクエストパケット内の値と一致しない場合はアクセスを拒否します。複数のクライアントが共通のプラットフォームからホストされている場合は、このファイル内に同じ IP アドレスが複数回含まれている可能性があるため、このアドレスをその IP アドレスに正しく適用されるものと同じ数のユーザー名に関連付けることができます。
 - クライアントアプリケーションがリクエストでユーザー名を渡さない場合は、`internet.addresses` ファイルがそのクライアントのユーザー名を確立します。この場合は、どのクライアント IP アドレスにも 1 つのユーザー名しか関連付けることができません。
4. `internet.addresses` ファイルへの更新をすべて保存します。
 - `acsss_config` を実行します。
 - 「Option 6 - "Rebuild Access Control Information"」を選択します。

ACSLs は、この変更を動的に認識します。

TCP/IP を使用しない SNA および OSLAN クライアントの場合は、`access_control` ディレクトリ内の `lu62.names` または `adi.names` ファイルを参照してください。

ユーザーのボリュームへのアクセスが許可されるほかのユーザーの定義

ユーザーによって所有されているボリュームへのアクセスをほかのユーザーに許可するには:

1. `access_control` ディレクトリ内にファイル `users.ALL.allow` または `users.ALL.disallow` を作成します。

テンプレート `users.SAMPLE.allow` または `users.SAMPLE.disallow` をコピーできます。

- このファイルに所有者ごとのレコードを追加し、所有者のユーザー ID を左マージンに置きます。
- 影響を受けるユーザーを各所有者と同じ行に指定します。
- 次の例に示すように、各ユーザー名をスペースまたはタブで区切ります。

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

`users.allow` および `users.disallow` ファイルにリストされているユーザー名は、大文字小文字は関係なく、一意である必要があります。ユーザー名の大文字小文字は無視されます。

所有者と同じ行にリストされていないユーザーには、所有者のボリウムへのデフォルトの (`ACCESS` または `NOACCESS`) 関係が与えられます。

注:

同じコマンドまたは ALL に対する `users.COMMAND.allow` ファイルと `users.COMMAND.disallow` ファイルの両方に同じ `owner_ID` と `user_ID` のペアを含めることはできません。また、同じ `users.COMMAND.allow` および `users.COMMAND.disallow` ファイル内に重複した `owner_ID` と `user_ID` のペアを含めることもできません。これには、同じ行での同じ `user_ID` の繰り返しが含まれます。

1 人の所有者に対して許可されるユーザーの数が多すぎて 1 行に収まらない場合は、許可されるユーザーのリストを以降の行に続けることができます。各行が所有者 ID で始まる必要があります。

- オプションで、定義したボリウムアクセスポリシーの例外を確立できます。

ユーザーには一般に、アクセス制御の下にあるボリウムへのフルアクセスが許可されるか、またはどのアクセスも許可されません。ただし、ユーザーに、ほかのユーザーのボリウムへの特定の制限されたアクセスを許可できます。

たとえば、特定のユーザーによって所有されているボリウムのマウントまたはマウント解除はできないが、これらのボリウムの照会をすべてのユーザーに許可するポリシーを設定できます。例外は、アクセス制御によって影響を受けるすべてのコマンドに適用できます。

特定のコマンドに対するボリウムアクセスポリシーの例外を構成するには:

- `users.COMMAND.allow` または `users.COMMAND.disallow` ファイルを作成する必要があります (ここで、`COMMAND` は、許可または制限する特定のコマンドに置き換えられます)。

`users.COMMAND.allow` および `users.COMMAND.disallow` ファイルには、下に示されているものとまったく同じように指定された名前 (コマンド名は大文字) を持つコマンドコンポーネントが含まれている必要があります。コマンドのその他のバリエント (`QUERY_VOLUME` など) へのアクセスの制御はサポートされていません。

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

注:

- `MOUNT (1)` - `MOUNT` ポリシーは、`mount scratch` にも適用されます。ポリシーは、`mount readonly` には適用されません
 - `MOUNT_READONLY (2)` - `mount readonly` のためのポリシーは、`mount` とは別に定義されています。
 - 所有者 ID と許可されるユーザー ID のペアが重複してはいけないこと、および許可されるユーザーのリストを以降の行に続けることに関する上の考慮事項は、禁止されるユーザーのリストにも適用されます。
6. 定義するポリシーへの更新をすべて保存します。
 - `acsss_config` を実行します
 - 「Option 6 - "Rebuild Access Control Information"」を選択します。

ACSLs は、この変更を動的に認識します。

ボリュームの所有権の確立

ボリュームアクセス制御は、明示的な所有権を持つボリュームにのみ適用されます。ライブラリ内の所有されていないボリュームは、すべてのユーザーからアクセスできます。ボリューム所有権を明示的に設定するには、`cmd_proc` インタフェースを使用します。

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

空の文字列を使用すると、同様の方法で所有権を削除できます。

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

この操作によって、範囲内のすべてのボリュームから所有権がクリアされます。詳細は、「[set owner](#)」を参照してください。

ボリューム所有権は、`watch_vols` ユーティリティーで自動的に設定できます。詳細は、「[watch_vols](#)」を参照してください。

所有権ポリシー

所有権を自動的に設定および削除するためのポリシーは、ACSLs でも定義できます。たとえば、マウントされたすべてのスクラッチボリュームが、それをマウントしたユーザーによって所有されるようになるポリシーを設定できます。それ以降、そのボリュームはそのユーザーによって所有されます。同じポリシーを、そのボリュームがスクラッチステータスに戻された場合は常に所有権を削除するように拡張することもできます。ポリシーを記述して、挿入されたすべてのボリュームがデフォルトユーザーか、その挿入をリクエストしたユーザーか、またはそのボリュームが以前に所有されていた場合は以前の所有者に割り当てられるようにもできます。この機能によって、非常に高い柔軟性が提供されます。

所有権ポリシーは、`access_control` ディレクトリ内に存在する `ownership.assignments` ファイルで定義されます。このファイル内のポリシーを、`enter`、`automatic enter`、`set scratch`、または `mount scratch` 操作ごとに所有権を自動的に割り当てまたは割り当て解除するように設定できます。`ownership.assignments` ファイルでは、デフォルトの所有者を定義できます。あるボリュームでこれらのいずれかの操作が実行された場合は常に、その所有権を次に割り当てることができます。

- `Owner_default` (デフォルトの所有者)
- 同じ (以前の所有者)
- リクエスタ (現在のリクエストを発行したユーザー)

- 未所有 (ボリュームの所有権を取り消します)

注:

所有権ポリシーを定義するための手順については、*ownership.assignments* ファイルで詳細に説明されています。このファイルには、ボリューム所有権を設定するために使用できるコマンドの完全なリストが含まれています。

- 定義するポリシーへの更新をすべて保存します。
 - *acsss_config* を実行します
 - 「Option 6 - Rebuild Access Control Information」を選択します。

ACSLS は、この変更を動的に認識します。

所有権の確認

所有権を確認するには、*owner_id.volrpt* テンプレートを使用して *volrpt* を実行できます。

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

これにより、ライブラリ内のすべてのボリュームの表示が生成され、それぞれに関連付けられた所有者とともに一覧表示されます。

ボリュームアクセスのサマリー

ボリュームアクセス制御では、次のコマンドがサポートされています。

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

force オプションが StorageTek ACSLS にボリューム ID を無視し、ボリュームを無条件にマウント解除するよう指示するため、*dismount force* にアクセス制御は適用されません。

次の表は、ボリュームアクセス制御が有効になっているときに適用されるコンテキストについて要約したものです。

表7.1 ボリュームアクセスが有効

ボリュームに対するデフォルトアクセスは ACCESS	アクセスが許可される	アクセスが拒否される
アクセスは <i>cmd_proc</i> 経由である	X	
指定されたボリュームは所有されていない	X	
ユーザーはボリュームの所有者である	X	
ユーザーは <i>users.ALL.disallow</i> 内の所有者に関連付けられている		X
ユーザーが <i>users.ALL.disallow</i> 内の所有者に関連付けられていない場合	X	

表7.2 ボリュームアクセスが有効

ボリュームに対するデフォルトアクセスは NOACCESS	アクセスが許可される	アクセスが拒否される
アクセスは <i>cmd_proc</i> 経由である	X	
指定されたボリュームは所有されていない	X	
ユーザーはボリュームの所有者である	X	
ユーザーは <i>users.ALL.allow</i> 内の所有者に関連付けられている	X	
ユーザーが <i>users.ALL.allow</i> 内の所有者に関連付けられていない場合		X

コマンドアクセス制御

コマンドアクセス制御では、ACSL S 管理者が特定のクラスのコマンドを、ネットワーク全体にわたる特定のクライアントアプリケーションまたは特定のユーザーに制限できます。制御されたアクセスは、ACSAPI 経由で送信されるユーザーコマンドにのみ適用され、*cmd_proc* を使用してコマンドを送信するローカルユーザーには適用されません。

ACSL S でコマンドアクセス制御を構成するプロセスには、3つの手順が含まれます。

ACSL S でコマンドアクセス制御をはじめて構成する場合は、次の手順に従います。

1. ACSLS でコマンドアクセス制御を有効にします。
2. クライアント ID をユーザー名に関連付けます。
3. どのユーザーがどのようなコマンドを使用できるかを定義します。

コマンドアクセス制御の有効化

ACSL5 でコマンドアクセス制御を有効にするには、

1. 構成ユーティリティー `acsss_config` を実行します。

メインメニューが表示されます。

2. 「Option 4 - Set Access Control Variables」を選択します。

各変数が一度に 1 つずつ表示され、その現在の設定が表示されます。

3. 現在の設定またはデフォルト設定を受け入れるには、**Enter** をクリックします。
4. ユーティリティーにメッセージ *Access control is active for commands* が表示されたら、「TRUE」を選択して Enter をクリックします。
5. メッセージ「Default access for commands」が表示されたら、次を実行します。
 - すべてのユーザーにすべてのコマンドへのアクセスを許可する場合は、「ACCESS」を選択します。

特定のユーザーのコマンド発行をブロックするには、それらのユーザーを `command.ALL.disallow` ファイルまたは特定の `command.XXX.disallow` ファイルにリストしておく必要があります。ここでは:

xxx は、アクセス制御の対象となるコマンドです

- コマンドへのユーザーアクセスを拒否する場合は、「[NOACCESS]」を選択します。

特定のユーザーのコマンド発行を許可するには、それらのユーザーを `command.ALL.allow` ファイルまたは特定の `command.XXX.allow` ファイルにリストしておく必要があります。

注:

コマンドへのアクセスが拒否されたインスタンスをログに記録する場合は、そのプロンプトに応答して「TRUE」を入力します。

注:

コマンドアクセスを有効または無効にした場合は常に、変更を有効にするために ACSLS を再起動する必要があります。

クライアント ID のユーザー名への関連付け

???)の手順を参照してください。

どのユーザーがどのようなコマンドを使用できるかの定義

このプロセスは、コマンドアクセス制御を有効にしたときに選択したデフォルトの動作によって異なります。`$ACS_HOME/data/external/access_control` ディレクトリ内にポリシーファイルを作成する必要があります。

- 上で定義したデフォルトの動作が [NOACCESS] である場合は、すべての ACSLS コマンドへのアクセスを許可する各クライアントのユーザー ID を含む `command.ALL.allow` ファイルを作成する必要があります。各ユーザー ID をこのファイル内の個別の行にリストするようにしてください。

特定のユーザーに特定のコマンドのみを許可する場合は、そのユーザーが実行を許可されるコマンドごとに `command.XXX.allow` ファイルを作成する必要があります。たとえば、特定のユーザーにボリュームをライブラリに挿入する権限を許可するには、`command.ENTER.allow` という名前のファイルを作成し、認定された各「挿入」ユーザーの ID をファイル内の個別の行にリストします。

- 上で定義したデフォルトの動作が [ACCESS] である場合は、すべての ACSLS コマンドへのアクセスを許可しない各クライアントのユーザー ID を含む `command.ALL disallow` ファイルを作成する必要があります。各ユーザー ID をこのファイル内の個別の行にリストするようにしてください。

注:

同じコマンドまたは ALL に対する `command.XXX.allow` ファイルと `command.XXX.disallow` `command.XXX` ファイルの両方に同じ user_ID を含めることはできません。

コマンドアクセス制御の許可および禁止ファイルのコマンド名

`command.XXX.allow` および `command.XXX.disallow` ファイルには、下に示されているものとまったく同じように指定された名前 (コマンド名は大文字) を持つコマンドコンポーネントが含まれている必要があります。コマンドのその他のバリエーション (`QUERY_VOLUME` など) へのアクセスの制御はサポートされていません。

```
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

注:

ENTER (1) - ポリシーは仮想挿入および手動挿入に適用されますが、自動挿入には適用されません。
 MOUNT (2) - ポリシーは、*mount scratch* および *mount readonly* にも適用されます。

次の表は、コマンドアクセスがいつ許可されるかを判定するためのクイックリファレンスとして使用してください。

表7.3 コマンドアクセスが有効

コマンドに対するデフォルトアクセスは NOACCESS	アクセスが許可される	アクセスが拒否される
リクエストは <i>cmd_proc</i> から入力された	X	
<i>user_ID</i> は <i>command.COMMAND.allow</i> にリストされている	X	
<i>user_ID</i> は <i>command.ALL.allow</i> にリストされている	X	
-- その他のすべての状態 --		X

表7.4 コマンドアクセスが有効

コマンドに対するデフォルトアクセスは ACCESS	アクセスが許可される	アクセスが拒否される
リクエストは <i>cmd_proc</i> から入力された	X	
<i>user_ID</i> は <i>command.COMMAND.disallow</i> にリストされている		X
<i>user_ID</i> は <i>command.ALL.disallow</i> にリストされている		X
-- その他のすべての状態 --	X	

- 定義するポリシーへの更新をすべて保存します。
 - *acsss_config* を実行します

- 「Option 6 - "Rebuild Access Control Information"」を選択します。

ACSLs は、この変更を動的に認識します。

アクセス制御メッセージのロギング

ユーザーがアクセスを拒否されたために失敗したすべてのトランザクションをログに記録するためのポリシーを設定できます。メッセージには、ユーザー名と試行されたコマンドが表示されます。

アクセス制御のロギングを有効にするには:

1. `acsss_config` を実行し、「Option 4 - "Set Access Control Variables"」を選択します
2. 「Messages will be logged when access to commands or volumes is denied」のプロンプトで、[FALSE] を [TRUE] に変更します。
3. 「Option 6 - "Rebuild Access Control Information"」を選択します。

ACSLs はこの変更を認識し、コマンドのリクエストが拒否されるたびにロギングを開始します。

第8章 ライブラリ管理

ライブラリの管理には次のタスクが含まれる場合があります。

- 「ACS 番号の指定」
- 「ライブラリの監査」
- 「ACSLS への SCSI またはファイバ接続ライブラリの追加」
- 「拡張ストア機能の使用」
- 「混合メディアライブラリの管理」
- 「デュアル TCP/IP 接続の管理」
- 「デュアル LMU 構成の管理」
- 「デュアル LAN クライアント構成の管理」
- 「プライマリおよびセカンダリ LAN の IP アドレスを登録する」
- 「TCP/IP 接続のタイムアウト間隔を設定する」
- 「イベント通知への登録」
- 「所定の状態への自動回復」
- 「ライブラリが一時的に使用不可になっている場合のマウントおよびマウント解除のキューへの追加と再試行」
- 「テープドライブの移動、追加、または取り外し時の ACSLS の再構成」
- 「メディア検証」

ACS 番号の指定

ACSLS では、すべての ACS 番号を順番に割り当てることなく、ライブラリの構成または再構成を行うことができます。

例:

9310 ライブラリから SL8500 に移行するとします。9310 ACS は現在 ACS 0 で、SL8500 ACS は ACS 1 です。

ACS 番号のスキップにより、SL8500 ACS の番号を変更しなくても、SL8500 ACS にすべてのカートリッジとドライブを移行し、9310 ACS を削除できます。

SL8500 ACS の番号を変更するとしたら、すべてのボリュームが削除され、以降の監査で再度追加されることになるでしょう。また、すべての `drive_id` も変更されることになります。

既存の ACS の番号を変更するには、「[acs_renumber.sh](#)」を参照してください。

テープドライブの移動、追加、または取り外し時の ACSLS の再構成

テープドライブの場所を入れ替えたときや、ライブラリ内のテープドライブを別のテープドライブに交換したときは常に、ACSLS データベース内のドライブのシリアル番号およびドライブのタイプを更新するために ACSLS を再構成する必要があります。ライブラリに対してテープドライブの挿入または取り外しを行なった場合は、再構成してテープドライブを追加または削除してください。テープドライブは、メディア検証プールに追加されると ACSLS のアクセスから削除され、メディア検証プールから削除されると ACSLS に追加されます。

テープドライブを使用して既存のドライブを交換した場合、ACSLS がライブラリからテープドライブのステータスを読み取るまで、ドライブのタイプおよびシリアル番号は更新されません。これは、次の場合に発生します。

- ACSLS の起動
- ACS または LSM が準備未完了状態になったあとで準備完了状態になったため、ACSLS がライブラリを回復するとき
- 顧客が ACS、LSM、またはテープドライブをオフラインに変更してからオンラインに戻したとき
- 顧客がテープドライブ、LSM、または ACS を再構成したとき。

テープドライブの追加または取り外しを行なった場合は、ACSLS 構成を更新して、ACSLS データベース内のドライブを追加または削除する必要があります。

ACSLS のテープドライブ構成を更新することで、マウントのエラーを回避し、間違ったテープドライブにカートリッジをマウントすることを防ぎます。

ACSLS に構成されているテープドライブを更新する

ライブラリ内のテープドライブが移動または交換された場合は、動的構成を使用してドライブのタイプおよびシリアル番号を更新してください。1 つのテープドライブを同じ場所で別のものに交換する場合、更新に顧客確認は必要ありません。ライブラリに対してテープドライブの取り外しまたは挿入を行う場合は、構成の変更を顧客が確認する必要があります。

注:

- `config` リクエストを発行する前に、影響を受けるすべてのコンポーネントの準備ができていないことを確認してください。
- 動的構成を使用した ACSLS データベースの更新は、ACSLS が有効になっているときに行われます。動的構成は中断を伴わないため、構成の更新中も ACSLS でリクエストの処理を継続できます。
- `config lsm` または `config acs` を発行したあとは、影響を受けた LSM または ACS を監査することをお勧めします。

テープドライブ構成を更新するには、UNIX コマンドプロンプトから動的構成ユーティリティーのコマンドを使用します。`acsss` としてログインしている必要があります。

- `config drive <panel_id>`

単一のパネルまたは SL8500 レール上のテープドライブだけが変更の影響を受ける場合、パネル上のすべてのテープドライブに関してドライブ構成を更新するには、`config drive <panel_id>` を使用します。

- `config lsm <lsm_id>`

2つのドライブパネルを備えた SL3000 内のすべてのテープドライブに関してドライブ構成を更新するには、`config lsm <lsm_id>` を使用します。

注:

`config lsm <lsm_id>` では LSM 内の CAP およびストレージ容量も更新されるため、続いて LSM を監査するようにしてください。

- `config acs <acs_id>`

ACS (SL8500 ライブラリコンプレックスなど) 内のすべてのものに関して構成を更新するには、`config acs <acs_id>` を使用します。

注:

`config acs <acs_id>` では ACS 全体の CAP およびストレージ容量も更新されるため、続いて ACS を監査するようにしてください。

ライブラリの監査

監査は、ライブラリカートリッジの実際のインベントリに一致するように ACSLS データベースを更新します。

監査のしくみ

監査では、重複するボリュームおよび無効なボリュームが取り出されます。カートリッジに次のものが含まれています。

- すでにスキャンされたものと重複する外部ラベル。
- 見つからないまたは読み取れない外部ラベル、および仮想ラベルがない。
- 無効なメディアタイプ。
- 無効なボリューム ID。

ACSL S は監査から得られたすべてのデータベースの変更をイベントログに記録し、さらに監査中に `cmd_proc` メッセージを表示します。監査は LSM のストレージセルにのみ適用され、テープドライブまたは CAP には適用されません。監査の実行については、「[audit](#)」を参照してください。

監査を実行するタイミング

監査は次のために実行します。

- 新しく構成されたライブラリのボリューム情報をデータベース内に作成します。
- カートリッジが CAP から挿入されなかった場合にデータベースにボリュームを追加します。

例: LSM をライブラリに追加し、LSM のドアを開いて、LSM にカートリッジを手動で追加しました。

- ライブラリとデータベースの間の不一致を解決します。

例: LSM のドアを開き、カートリッジを CAP からではなく手動で取り出した場合は、監査を実行します。監査により、ボリュームは不在としてマークされるか、取り外されたボリュームがデータベースから削除されます。

ACSL S に対して構成または再構成したあとで、SL3000 を監査します。ライブラリは、ACSL S が監査するときに、ACSL S にアクセスできない可能性のあるセルの場所を報告します。アクセスできないセル位置には次のようなものがあります。

- CAP、ドライブ、およびオペレータパネルが取り付けられている場所。
- ロボットがアクセスできないセル位置。
- アクティブ化されていないセル位置。
- このパーティションに含まれていないセル位置。

監査間隔

監査間隔はいくつかの要因によって異なります。たとえば、ACSLS 構成、ライブラリがセルの内容をデータベースから報告するのか各セルを検査することによって報告するのか、ライブラリ構成、必要なデータベース変更の数、監査の範囲などの要因があります。次の表では、監査の範囲の違いが監査間隔にどのように影響を与えるかについて説明します。

表8.1 監査の範囲が監査間隔に与える影響

監査時間が短い	監査時間が長い
診断 ACS/LSM	オンラインの ACS/LSM
監査専用の ACS/LSM	ほかのリクエストを処理する ACS/LSM
いっぱいになっているパネル	(一部または完全に) 空のパネル
ドライブパネル	標準パネル
内部パネル	外部パネル

また、次の表に示すように、監査するコンポーネントの LSM タイプも考慮してください。4410、9310、および L5500 では、すべてのセルがいっぱいになっている場合に時間が最短になり、セルが空の場合に時間が最長になります。

表8.2 サポートされている LSM の平均監査時間

LSM タイプ	コンポーネント	平均監査時間 (最短および最長)
4410	LSM	3 - 8 時間
9310/L5500	LSM	1.2 - 6 時間
9360	LSM	5 分
97xx/L700/180	LSM	1 分
SL500	LSM	3 分
SL8500 または SL3000	LSM	5 分 (SL8500 の物理的監査が ACSLS 監査の 前に完了する場合)

注:

これらの時間は、ほかにアクティブなプロセスがないことを前提としています。4410 および 9310 では、監査されるセルが高い割合で占有されている場合は、監査時間が短くなる可能性があります。

ACSL S への SCSI またはファイバ接続ライブラリの追加

SCSI メディアチェンジャー (mchanger) は、ACSL S と任意の SCSI ライブラリ間で通信を行うデバイスドライバです。ACSL S に接続される SCSI またはファイバ接続ライブラリごとに、mchanger を作成する必要があります。

詳細は、「[SCSI mchanger デバイスドライバの追加](#)」を参照してください。

拡張ストア機能の使用

以降のセクションでは、拡張ストア機能の使用に関する情報を提供します。

注:

この機能は、パススルーポートのない単一の LSM には適用されません。

カートリッジがマウントされる時、その「ホーム位置」はマウント元のストレージセルです。通常は、カートリッジがマウント解除される時、ドライブとは異なる LSM にそのホーム位置がある場合、ACSL S はもっとも近い LSM (テープドライブから最短のパススルー距離) に新しいホーム位置を割り当てようとします。これは、新しいホーム位置へのカートリッジの「フロート」と呼ばれます。

拡張ストア機能のために使用可能になっている LSM 内にカートリッジのホーム位置がある場合、ACSL S は、マウント解除後にカートリッジをそのホーム位置に戻そうとします。

注:

マウント解除時にボリュームのホーム LSM がオフラインであったり、ラベルが一致していなかったりすると、ACSL S はカートリッジをそのホーム位置に戻せないことがあります。そのような場合、カートリッジはマウント解除された LSM からできるだけ近くに格納されます。

カートリッジのアーカイブに使用する拡張ストア LSM がある場合、その LSM のカートリッジは、この機能によって確実にその LSM 内のホーム位置に戻ります。たとえば、拡張ストア LSM 3 がこの機能のために使用可能になっている場合に、LSM 3 のカートリッジが LSM 1 に接続されたドライブにマウントされているとき、カートリッジのマウント解除後に、ACSL S はカートリッジを LSM 3 内のホーム位置に戻そうとします。LSM 3 が使用可能になっていない場合、ACSL S は LSM 1 内の新しいセルにカートリッジを格納しようとします。

この機能のために使用可能にできるのは LSM 全体のみです。パネルや個々のセルなど、LSM のサブコンポーネントを使用可能にすることはできません。ACS 全体をこの機能のために使用可能にするには、ACS 内の各 LSM を使用可能にする必要があります。

注:

拡張ストア機能を有効にすると、カートリッジのマウント解除時のバススルーアクティビティが増加します。これによってライブラリのパフォーマンスが大幅に低下する可能性があります。

LSM を拡張ストア機能のために使用可能にするには、*lsm_fixed_volume* ファイル (*\$ACS_HOME/data/external/fixed_volume* ディレクトリにある) を変更します。ファイルを変更したあとは、指定した LSM を使用可能にするために、ACSLIS を停止して再起動します。

サンプルファイルを変更するときは、次の規則を使用します。

- コメント行および空白行は、ファイル全体で使用できます。
- 各 LSM 識別子の行は、ACS 番号、コンマ、および LSM 番号で構成されます。ACS 番号と LSM 番号の間にスペースやタブがあってははいけません。
- LSM 識別子ごとに行を変える必要があります。
- LSM 識別子の行の明示的な順序付けは不要です。
- ファイルの終わりを示す明示的な区切り文字は不要です。

拡張ストア機能の制御ファイルの例:

```
# 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
```

次は、変更された制御ファイルの例です。この例では、太字で示されている行からコメント (#) 文字が削除されており、LSM 0,0 および 0,1 を使用可能にするように指定しています。

```
# 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
```

混合メディアライブラリの管理

混合メディアにより、ACSLs は同じライブラリ内でのテープドライブおよびメディア (カートリッジ) タイプの混在をサポートします。ACSLs の混合メディアサポートは、ロボットがテープドライブに互換性のないメディアタイプをマウントすることを防止します。たとえば、SL8500 内で、ロボットは T10000 テープドライブに LTO カートリッジをマウントしません。

ACSLs の混合メディアサポートには、カートリッジ上のメディア ID ラベルにメディア文字が記載されている必要があります。

注:

現在サポートされているドライブタイプ、メディアタイプ、およびドライブとメディアの互換性のリストについては、ACSLs の製品情報ガイドを参照してください。

カートリッジで使用されている記録形式について ACSLS の知識は限られています。ACSLs は、テープドライブへのデータパスにアクセスできないため、記録形式に互換性がないことを検出および防止することはできません。ただし、最新の T9840、T9940、および T10000 テープドライブでは、マウント解除時にメディアの記録形式が報告されます。この情報は ACSLS によってデータベースに保存され、次のコマンドで表示できます。

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

詳細は、「[display volume](#)」を参照してください。

注:

9310 およびその他の廃止されたライブラリの場合、`venter` コマンドを使用すると、ラベルのないカートリッジをライブラリに挿入できます。`venter` コマンドには、挿入するカートリッジのメディアタイプを指定するオプションはありません。混合メディア環境では、ACSL5 は仮想的に挿入されたカートリッジについてテープドライブ/メディアの非互換性を防止できません。ただし、SL8500 および SL3000 ライブラリは `volser` ラベルなしでメディアラベルを読み取ることができず、ACSL5 はこれらのライブラリにカートリッジを挿入するためにメディアラベルを必要とするため、`venter` はこれらのライブラリではサポートされていません。

ACSL5 の混合メディア設定を表示する

`drives_media.sh` ユーティリティは、ACSL5 で現在サポートされているドライブタイプ、メディアタイプ、およびドライブとメディアの互換性の設定を表示します。新しいドライブとメディアのサポートが追加されると、それらが表示されます。

ACSL5 の混合メディア設定を表示するには、次のように入力します。

```
drives_media.sh
```

情報は画面 (標準出力) に書き込まれます。

ACSL5 の混合メディア設定を `/tmp` ディレクトリ内のファイルに出力するには、次のように入力します。

```
drives_media.sh -f
```

情報は 3 つのファイルに書き込まれます。(ファイルがすでに存在する場合、それらは上書きされます。)

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

SCSI 接続 LSM の混合メディアの制限事項

SCSI 接続 LSM に関する次の制限事項に注意してください。

- 一部のテープドライブでは、制御パスを介した動的な書き込み保護設定がサポートされていないため、「読み取り専用」オプションを使用してカートリッジをマウントするときは注意が必要です。StorageTek のドライブはすべてこの機能をサポートしています。StorageTek 以外のドライブの場合、動的な書き込み保護がサポートされていることを確認することをお勧めします。特に、LTO ドライブおよび一部の早期の DLT ドライブは「読み取り専用」オプションをサポートしていません。

ドライブでこの機能がサポートされていない場合は、読み取り専用マウントの書き込み無効化保護を使用してマウントされると想定したカートリッジのデータを失うリスクがあります。

- ACSLS は、DLT テープドライブを備えた SCSI 接続ライブラリに対する *venter* コマンドをサポートしません。
- SCSI 接続ライブラリ内の DLT テープドライブに対して仮想マウントおよび仮想マウント解除を行うことはできません。したがって、ACSLS で Compac テープカートリッジを管理するには、カートリッジに外部ラベルが必要です。
- SCSI 接続ライブラリの自動クリーニング操作は、ACSLS ではなくライブラリのマイクロコードによって処理されます。ライブラリ制御パネルには、自動クリーニング操作のライブラリ制御を有効にするためのユーザー用メニューが用意されています。詳細については、ライブラリのドキュメントを参照してください。
- ライブラリを介した通常のロードまたは高速ロードオプションを選択できます。ただし、すべてのテープ管理システムが高速ロードオプションをサポートしているわけではありません。

スクラッチ優先順位の設定

scratch mount リクエストでは、使用するメディアタイプを明示的に指定するか、ACSLS にメディアタイプを選択させることができます。

ACSLS でメディアタイプを選択するには、ドライブタイプごとに、互換性のあるメディアタイプに優先順位を付けたリストを事前に定義する必要があります。このリストは「スクラッチ優先順位」と呼ばれます。

- サーバー全体に 1 セットのプリファレンスがあります。プリファレンスはクライアントでは定義されません。
- ドライブに互換性のあるメディアタイプが表示されていない場合、メディアは選択されません。

以降のセクションでは、ACSLS でスクラッチ優先順位の決定に使用される、ユーザー定義およびシステム定義のファイルについて説明します。

ユーザー定義の混合メディアファイル

次のファイルは、`$ACS_HOME/data/external/mixed_media/` にあるユーザー定義の混合メディアファイルです。

- *scratch_preferences.dat*

ユーザー定義のプリファレンスファイル。プリファレンス定義のプライマリソースです。

- `scratch_preferences.SAMPLE`

サンプルプリファレンスファイルをコピーして `scratch_preferences.dat` ファイルを作成できます。

システム定義の混合メディアファイル

`drives_media.sh -f` を使用して、ACSL S の混合メディア設定を表示します

次の場所に保存されているメディア互換性設定を確認します。

- `/tmp/media_compatibility.txt`

これは、システム定義の互換性設定ファイルです。これはユーザー定義のプリファレンスファイルが存在しないか、ファイルにドライブタイプが見つからない場合にのみ使用されます

- `/tmp/drive_types.txt`

システムで定義されている、サポートされているドライブタイプのリスト。

- `/tmp/media_types.txt`

システムで定義されている、サポートされているメディアタイプのリスト。

ACSL S が混合メディアファイルを使用するしくみ

次の表では、ACSL S がスクラッチマウントリクエストのメディアタイプを選択するために、混合メディアファイルをどのように使用するかについて説明します。

表8.3 ACSL S が混合メディアファイルを使用するしくみ

<code>scratch_preferences.dat</code> ファイル	ACSL S の実行内容 の状況
存在しません。	システムの <code>media_compatibility</code> ファイルに含まれている定義を使用します
1つのドライブに対して複数のメディアタイプが示されています。	リストに示された順序でメディアタイプを選択します。
特定のドライブに対してメディアタイプが示されていません。	システムの <code>media_compatibility</code> ファイルに含まれているデータを使用します

scratch_preferences.dat ファイル ACSLS の実行内容 の状況

特定のドライブタイプが示されていま システムの *media_compatibility* ファイルに含まれ
せん。 ているデータを使用します

スクラッチ優先順位ファイルを定義する

scratch_preferences.dat ファイルを定義するには、次の手順を使用します。このファイルには、特定のドライブタイプに対して選択すべきスクラッチカートリッジタイプを順序付けしたリストが含まれます。ACSLS は、*mount * コマンド* にメディアタイプが明示的に指定されていない場合にこのファイルを使用します。

次の例は、*scratch_preferences.SAMPLE* の内容を示しています。

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

スクラッチ優先順位ファイルを定義するには、次の手順を実行します。

1. *acsss* としてログインします。
2. 外部混合メディアのディレクトリに移動します。

```
cd $ACS_HOME/data/external/mixed_media
```

3. サンプルのスクラッチ優先順位ファイルをコピーして、ユーザー定義ファイルを作成します。

```
cp scratch_preferences.SAMPLE scratch_preferences.dat
```

4. *vi* などのテキストエディタを使用して、*scratch_preferences.dat* ファイル内のプリファレンスのリストを変更します。
 - ファイルの上部にあるコメントの説明に従います。
 - 1つのドライブタイプに対して複数のメディアタイプを使用する場合は、各メディアタイプを別々の行に入力します。上から下の順に優先されます。
5. ファイルを保存します。
6. *cmd_proc* から ACSLS を再起動します。

```
start
```

デュアル TCP/IP 接続の管理

デュアル TCP/IP は、SL8500 および SL3000 に購入できるオプションです。ライブラリに 2 つの TCP/IP 接続を提供します。ただし、2 つの接続の 1 つのみを動作させてライブラリを使用することもできます。

デュアル TCP/IP の目的は、障害の発生した通信パスを自動的に認識し、回避することです。これは自動的に行われるため、動作不能な接続から手動で切り替える必要はありません。詳細は、「[デュアル TCP/IP サポート](#)」を参照してください。

デュアル LMU 構成の管理

ACSLS は、次のものから成るデュアル LMU 構成をサポートします。

- 1 台のアクティブ LMU。ACS を管理します
- 1 台のスタンバイ LMU。アクティブ LMU に障害が発生した場合は、ACS を管理するアクティブの役割に自動的に切り替わります

どちらの LMU も、LSM に接続された LAN に接続されています。最初は、先に電源が投入された LMU がアクティブで、2 番目に電源が投入された LMU がスタンバイになります。LMU は互いのステータスを定期的を確認するため、アクティブ LMU に障害が発生した場合にスタンバイ LMU がアクティブの役割を引き継ぐことができます。

注:

ACSLS でデュアル LMU 構成がサポートされるのは、ホスト/LMU マイクロコード互換性レベル 12 (またはそれ以上) がロードされている 9330 および L5530 LMU のみです。同じマイクロコードレベルが両方の LMU にロードされている必要があります。ACSLS は、シリアル接続または TCP/IP によってこれらの LMU と通信します。Ethernet 接続は、TCP/IP 接続された LMU ごとに 1 つだけ使用できます。

制限: ACSLS は LMU の切り替えを自動的に開始しません。ACSLS とアクティブ LMU との通信が失われた場合でも、ACSLS は LMU の切り替えを開始しません。ACSLS は、既存のアクティブ LMU との通信を続けようとします。冗長電子装置を備えた SL8500 または SL3000 ライブラリを含む ACSLS-HA 構成を除き、ACSLS は LMU の切り替えを自動的に開始しません。

ACSLS のデュアル LMU サポートには、次のものが含まれます。

- ACSLS はアクティブとスタンバイの両方の LMU に接続します。ACSLS は両方の LMU への接続を常にモニターします。通信が失われた場合、ACSLS はこの状態を報告します。

- アクティブ LMU に障害が発生した場合は、スタンバイ LMU が新しいアクティブ LMU として自動的に引き継ぎます。これが発生した場合、ACSL S は自動切り替えを認識し、新しいアクティブ LMU にリクエストを送信します。ACSL S は、処理中のトランザクションの回復も行います (監査を除く)。
- `switch lmu` コマンドを発行して、ACS の管理をアクティブ LMU からスタンバイ LMU に手動で切り替えることができます。ACSL S は、スタンバイ LMU に「アクティブへのスイッチオーバーの強制実行」伝達を送信します。スタンバイは新しいアクティブ LMU として引き継ぎます。手動スイッチオーバーのあと、ACSL S は未処理のトランザクションを回復しません (監査を除く)。

シリアルデュアル LMU 構成の冗長性をさらに高めるために、デュアルシリアルケーブルを使用して ACSLS サーバーを各 LMU に接続することをお勧めします。

また、ACSL S のデュアル LMU サポートにより、ACSL S サーバーを停止することなく LMU で IPL を実行できるため、シングル LMU 構成も強化されます。

`query lmu` コマンドは、シングル LMU およびデュアル LMU の両 ACS 構成について、LMU およびポートのステータスを表示します。詳細は、「[query lmu](#)」を参照してください。

`switch lmu` コマンドを使用して、ACS の管理を ACS のアクティブ LMU からスタンバイ LMU に手動で切り替えることもできます。詳細は、「[switch lmu](#)」を参照してください。手動スイッチオーバーは、デュアル LMU を接続する LAN ケーブルの交換などのハードウェア保守に使用できます。

たとえば、LMU A がアクティブの役割で、LMU B がスタンバイの役割であると仮定します。LAN ケーブルを交換する必要がある場合は、次のように実行できます。

1. LMU B に切り替えます。
2. LMU A へのポートをオフラインに変更 (`vary`) します。
3. LAN ケーブルを交換します。
4. LMU A へのポートをふたたびオンラインに変更 (`vary`) します。
5. ふたたび LMU A に切り替えます。

LMU のスイッチオーバー (自動または手動) のあと、ACSL S は未処理の (アクティブおよび保留中の) リクエストをすべて回復します。スイッチオーバー時に未処理の各リクエストを完了する時間は、LMU 間の切り替え、および前の未処理のリクエストを回復するために必要な時間によって増加します。したがって、リクエストの回復には 3 - 5 分かかることがあります。

デュアル LAN クライアント構成の管理

ACSL5 5.2 以上では、プライマリ LAN とセカンダリ (バックアップ) LAN から成るデュアル LAN クライアント構成がサポートされています。プライマリ LAN に障害が発生した場合、クライアントは、セカンダリ LAN に切り替えます。ACSL5 は、そのクライアントに対する未処理のメッセージをすべて削除し、セカンダリ LAN を使用して通信を開始します。セカンダリ LAN にスイッチオーバーする前に ACSL5 は未処理のメッセージをすべて削除するため、リクエストが正常に完了したとしてもクライアントは成功メッセージを受け取りません。

たとえば、クライアントが ACSL5 に 10 個のカートリッジの取り出しをリクエストします。ACSL5 がカートリッジの取り出しを開始したあとで、ACSL5 とプライマリクライアント LAN の間の通信に障害が発生します。ACSL5 は、そのクライアントに対する未処理のメッセージをすべて削除し、セカンダリ LAN を介して通信を開始します。ACSL5 は 10 個のカートリッジすべての取り出しを正常に完了しますが、クライアントに成功メッセージを送信しません。クライアントは、リクエストが正常に完了したことを確認する必要があります。この例で、クライアントは取り出し済みボリュームの ID に対して `query volume` リクエストを発行し、ACSL5 は「`volume not found`」というエラーメッセージを返します。これにより、ACSL5 がカートリッジを取り出したことが確認されます。

LAN 通信のスイッチオーバーが発生した場合、ACSL5 では一時的なリクエストがすべて正常に完了することが保証されます。ただし、スイッチオーバーのあと、クライアントは最初にプライマリ LAN 上で送信した永続的な未処理のリクエスト (CAP 操作など) をすべて取り消し、セカンダリ LAN からの通信を介してこれらのリクエストを再送信する必要があります。プライマリ LAN 通信を介して割り当てられたリソース (CAP、ロック、ドライブなど) は、セカンダリ LAN へのスイッチオーバー後も割り当てられたままになります。

以降のセクションでは、次の手順を実行して ACSL5 をデュアル LAN クライアント操作用に構成する方法について説明します。

- `csc_ip_switch.dat` ファイルを作成することによって、プライマリおよびセカンダリ LAN の IP アドレスを登録します
- マルチホーム ACSL5 サーバー用に 2 つ目の Ethernet ポートをインストールします
- バックアップ LAN へのシステムのスイッチオーバー時間を短縮するために、TCP/IP 接続のタイムアウト間隔を設定します。

プライマリおよびセカンダリ LAN の IP アドレスを登録する

プライマリおよびセカンダリ LAN の IP アドレスを ACSLS に登録するには、`$ACSSS_HOME/data/internal/client_config/` ディレクトリ内に `csc_ip_switch.dat` ファイルを作成します。`csc_ip_switch.dat` ファイルの例を次に示します。

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

上記の例で示すように、コメントの前には # 記号が付きます。エントリは、クライアントシステムのプライマリ LAN の IP アドレス (左側の列)、1 つ以上の空白、およびクライアントシステムのセカンダリ LAN の IP アドレス (右側の列) で構成されます。この例で、System Zed のプライマリ LAN の IP アドレスは `129.80.30.40`、セカンダリ LAN の IP アドレスは `129.80.30.50` です。

`csc_ip_switch.dat` ファイルを作成または更新したあと、ACSLS が実行されている場合は、ACSLS を停止して再起動する必要があります。

ACSLS は、再起動時に `csc_ip_switch.dat` ファイルを正常に読み取ると、「`2010 I DUAL PATH OPTION ACTIVATED`」という成功メッセージを ACSLS イベントログに記録します。それ以外の場合、デュアル LAN のサポートはアクティブ化されません。

マルチホーム ACSLS サーバー用に 2 つ目の Ethernet ポートをインストールする

このセクションでは、マルチホーム ACSLS サーバー用に 2 つ目の Ethernet ポートをインストールする手順について説明します。2 つ目の Ethernet ポートは、バックアップ LAN を制御する 2 つ目の制御パスアダプタへの接続を提供します。この手順には SBUS Buffered Ethernet カードが必要です。この Ethernet カードは、Oracle からパーツ番号 X1053A で注文できます。

このインストール手順には次の内容が含まれます。

- ハードウェアの取り付けおよびカーネルの再構築
- 新しい Ethernet ポートのホスト名の定義
- `/etc/notrouter` ファイルの作成

2 つ目の Ethernet ポートをインストールするには、次の手順を実行します。

1. 製造元の指示に従って SBus Buffered Ethernet カードを取り付けます。

2. 新しいデバイスのためにシステムカーネルを再構成します。
 - a. システムの電源を投入します。ブートが開始したら、`[[STOP]]-[[A]]` を押して PROM モニターに入ります。
 - b. `ok` プロンプトで、サーバーをブートします。

```
boot -r
```

3. ブートが完了したら、`root` としてログインします。
4. 2つ目の Ethernet ポートのホスト名を作成します。

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

ここで、`2nd_host_name` は 2つ目の Ethernet ポートのホスト名です。

5. 次のコマンドを入力します。

```
touch /etc/notrouter
```

これにより、`/etc/notrouter` ファイルが作成されます。

6. サーバーをリブートします。

```
reboot
```

これで手順は完了します。

TCP/IP 接続のタイムアウト間隔を設定する

UNIX システム変数 `tcp_ip_abort_cinterval` は、クライアントと ACSLS サーバー間の TCP/IP 接続のタイムアウト間隔を設定します。この変数のデフォルト値 (180 秒) を変更すると、バックアップ LAN へのスイッチオーバー時間が短縮される可能性があります。ただし、実際のスイッチオーバー時間は、構成および障害のタイプによって異なります。

たとえば、ACSLS サーバーのポートを介して複数のクライアントが通信している場合に、このポートに障害が発生すると、ACSLS は各クライアントとの通信を順に回復します。障害の発生したポートの複数のクライアントを回復するには、障害の発生したポートと通信している単一のクライアントを回復するよりも長い時間がかかります。

バックアップ LAN へのスイッチオーバー時間を短縮するには、次の手順を実行します。

1. ACSLS サーバーに `root` としてログインします。
2. プロンプトから、次のコマンドを入力します。

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

このコマンドは、TCP/IP 接続のタイムアウト間隔を 15 秒 (デフォルトは 180 秒) に変更します。

ヒント: このコマンドを (サーバーのリブート後も) 永続させるには、`/etc/rc2.d/S69inet` ファイルの「Set configurable parameters」セクションにこのコマンドを追加します。

イベント通知への登録

イベント通知を使用すると、ACSAPI クライアントはテープライブラリで発生するイベントを追跡できます。この機能は CSC ツールキットによって提供されます。具体的には、イベント通知を使用して ACSAPI クライアントは次のことを実行できます。

- ライブラリリソースイベントまたはボリュームイベント、あるいはその両方に登録します
- これらのイベントへの登録を解除します
- ライブラリイベントへの登録ステータスを確認し、イベント発生時に通知されるようにします

クライアントの登録リクエストおよびイベント通知メッセージは、登録が解除されるまで、クライアントに配信されます。イベント通知では、クライアントが生存していることを確認するために、クライアントの登録ステータスが定期的にチェックされます。これにより、アクティブでなくなっているクライアントに応答を送信することを避け、ネットワークリソースの無駄な使用を防止します。

次のタイプのイベントを追跡できます。

- ボリュームの追加と削除、およびクリーニングカートリッジの最大使用制限の超過が ACSLS データベース内で発生したとき。
- ライブラリコンポーネントのステータスの変更 (LSM またはドライブがオンラインからオフライン、診断、または回復に変化するなど)、または CAP の開閉。
- ハードウェア障害 (ロボットハンドの動作不能など)。

所定の状態への自動回復

ACSL SLS では、テープライブラリおよびドライブがユーザーの望む可用性に復元されるようになりました。このために、ACS、ポート、LSM、およびテープドライブについて現在の状態と所定の状態の両方が追跡されます。ライブラリまたはドライブが ACSLS からアクセスできなくなった場合や動作不能になった場合、ACSL SLS はその現在の状態をオフラインに変更しま

す。ライブラリまたはドライブがふたたびアクセス可能または動作可能になると、ACSL S はそれを自動的に回復し、所定の状態がオンラインである場合はそれをオンラインに戻します。

現在の状態および所定の状態

- 所定の状態は、ライブラリおよびテープドライブの可用性を管理します。所定の状態とは、ACS、ポート接続、LSM、またはテープドライブに対してユーザーが望む可用性のことです。ライブラリのすべてのコンポーネントで、最初の所定の状態はオンラインになっています。所定の状態は、明示的な vary コマンドを使用して設定できます。(これは、`cmd_proc`、ACSL S GUI、または ACSAPI クライアントからの vary です。ライブラリのステータス変更によって ACSLS が内部的に生成する vary ではありません。)ライブラリコンポーネントの所定の状態は、`query lmu` および `display` コマンドを使用して表示できます。
- ACS、ポート、LSM、ドライブの現在の状態 (「state」として指定される) は、そのコンポーネントの現在の可用性であり、所定の状態によって制限されます。現在の状態は、ライブラリコンポーネントが準備を完了して通信しているかどうかを示し、そのコンポーネントおよび上位コンポーネントの所定の状態によって制限されます。

たとえば、LSM の所定の状態がオンラインであっても、LSM の準備ができていない場合、その現在の状態はオフラインです。LSM の準備がふたたび完了すると、ライブラリは ACSLS にメッセージを送信し、ACSL S は LSM を自動的に回復し、その現在の状態をオンラインに戻します。

ただし、LSM の所定の状態がオフラインである場合、ACSL S は現在の状態をオフラインに設定します。その LSM が準備未完了状態になったあとで準備完了ステータスに戻った場合、ACSL S は LSM の現在の状態をオフラインのままにします。

現在の状態は下位へカスケードされ、所定の状態はそうならない

ライブラリコンポーネントの現在の状態は、下位コンポーネントにカスケードされます。

- 具体的には、ACS がアクセス不可になった場合、ACS の現在の状態はオフラインになります。すべての LSM およびドライブもアクセス不可になるため、それらの現在の状態もオフラインに設定されます。
- ACS の所定の状態が vary コマンドによってオフラインに変更されると、ACS の現在の状態はオフラインに設定され、その ACS 内にあるすべての LSM およびドライブの現在の状態がオフラインに設定されます。
- 同様に、LSM の現在の状態がオフラインに変化すると、その LSM 内にあるすべてのドライブの現在の状態がオフラインになります。ドライブの所定の状態は変更されません。

ライブラリコンポーネントの所定の状態を変更しても、下位コンポーネントの所定の状態は影響を受けません。

- ACS の所定の状態を変更しても、その ACS 内にある LSM およびドライブの所定の状態は影響を受けません。
- LSM の所定の状態を変更しても、その LSM 内にあるドライブの所定の状態は影響を受けません。
- これにより、LSM 内で選択したドライブをオフラインにして、アクセス不可にすることができます。そのあとで、保守のために LSM をオフラインに変更 (*vary*) できます。LSM をふたたびオンラインに変更 (*vary*) すると、選択したドライブはオフラインのままになり、その他のドライブはオンラインに戻ります。

物理ライブラリおよびドライブの可用性は、論理ライブラリおよびドライブに影響を与えます。

- 論理ライブラリおよび論理ライブラリ内のテープドライブの可用性は、基となる物理ライブラリと論理ライブラリの両方に設定した目的の状態によっても左右されます。
- 物理ライブラリと論理ライブラリの両方の目的の状態がオンラインの場合、論理ライブラリと論理テープドライブの現在の状態は、基となる物理ライブラリとドライブの現在の情報を反映します。
- ただし、物理 ACS またはドライブの所定の状態がオンラインであっても、論理ライブラリまたはドライブの所定の状態がオフラインである場合、論理ライブラリまたはドライブはオフラインのままになり、使用できません。

ライブラリが一時的に使用不可になっている場合のマウントおよびマウント解除のキューへの追加と再試行

ACSL S は、ライブラリが一時的に停止していることを検出すると、マウントおよびマウント解除リクエストを照会します。すべてのソースからのマウントおよびマウント解除リクエストは、必要なすべてのライブラリコンポーネントの所定の状態がオンラインである場合、自動的にキューに入れられ、再試行されます。したがって、次のソースからのマウントおよびマウント解除リクエストは、すべて自動的にキューに入れられ、再試行されます。

- ACSAPI クライアント
- *cmd_proc*
- 論理ライブラリ内のテープドライブのファイバ接続クライアント

ライブラリハードウェアの所定の状態がオンラインで、現在の状態がオフラインである場合、一時的な停止が発生します。一時的な停止の例は、LSM ドアが開かれたとき、ACSL S がラ

ライブラリとの通信を失ったとき、LC 切り替え操作中などです。ライブラリまたはテープドライブが一時的に停止している間、マウントおよびマウント解除はキューに入れられ、ライブラリが使用可能になったときに再試行されます。

ライブラリハードウェアの所定の状態がオフラインである場合、ACSLS はマウントまたはマウント解除リクエストを処理できず、適切なエラーステータスを示します。

ACSLS 7.3.1 では、マウントおよびマウント解除のキューへの追加と再試行を管理するために、2 つの動的変数 `MOUNT_RETRY_DELAY` および `MOUNT_RETRY_TIME_LIMIT` が導入されました。ここで、

- `MOUNT_RETRY_DELAY` は、キューに入れられたマウントおよびマウント解除を再試行する頻度、または、ライブラリおよびドライブの可用性を再チェックする頻度を制御します。
- `MOUNT_RETRY_TIME_LIMIT` は、マウントおよびマウント解除をキューに入れて再試行する時間制限です。この後、リクエストは失敗します。

テープドライブの移動、追加、または取り外し時の ACSLS の再構成

テープドライブの場所を入れ替えたときや、ライブラリ内のテープドライブを別のテープドライブに交換したときは常に、ACSLS データベース内のドライブのシリアル番号およびドライブのタイプを更新するために ACSLS を再構成する必要があります。ライブラリに対してテープドライブの挿入または取り外しを行なった場合は、ACSLS を再構成してテープドライブを追加または削除する必要があります。テープドライブは、メディア検証プールに追加されると ACSLS のアクセスから削除され、メディア検証プールから削除されると ACSLS でアクセスできるようになります。これらの場合、ACSLS の再構成も行う必要があります。

テープドライブを使用して既存のドライブを交換した場合、ACSLS がライブラリからテープドライブのステータスを読み取るまで、ドライブのタイプおよびシリアル番号は更新されません。これは、次の場合に発生します。

- ACSLS の起動。
- ACS または LSM が準備未完了状態になったあとで準備完了状態になったため、ACSLS がライブラリを回復するとき。
- ACS、LSM、またはテープドライブをオフラインに変更してからオンラインに戻したとき。
- テープドライブ、ACS、または LSM を再構成したとき。

テープドライブの追加または取り外しを行なった場合は、ACSLS 構成を更新して、ACSLS データベース内のドライブを追加または削除する必要があります。ACSLS のテープドライブ

構成を更新することで、マウントのエラーを回避し、間違ったテープドライブにカートリッジをマウントすることを防ぎます。

ACSLS に構成されているテープドライブを更新する

ライブラリ内のテープドライブが移動または交換された場合は、動的構成を使用してドライブのタイプおよびシリアル番号を更新してください。1 つのテープドライブを同じ場所で別のものに交換する場合、更新に顧客確認は必要ありません。ライブラリに対してテープドライブの取り外しまたは挿入を行う場合は、構成の変更を顧客が確認する必要があります。

テープドライブを更新する場合:

- `config` リクエストを発行する前に、影響を受けるすべてのコンポーネントの準備ができていることを確認してください。
- 動的構成を使用した ACSLS データベースの更新は、ACSLS が有効になっているときに行われます。動的構成は中断を伴わないため、構成の更新中も ACSLS でリクエストの処理を継続できます。
- `config lsm` または `config acs` を発行したあとは、影響を受けた LSM または ACS を監査することをお勧めします。

テープドライブ構成を更新するには、UNIX コマンドプロンプトから次の動的構成ユーティリティのコマンドを発行します。`acsss` としてログインしている必要があります。

- `config drive <panel_id>`

単一のパネルまたは SL8500 レール上のテープドライブだけが変更の影響を受ける場合、パネル上のすべてのテープドライブに関してドライブ構成を更新するには、`config drive <panel_id>` を使用します。

- `config lsm <lsm_id>`

2 つのドライブパネルを備えた SL3000 内のすべてのテープドライブに関してドライブ構成を更新するには、`config lsm <lsm_id>` を使用します。

注:

`config lsm <lsm_id>` では LSM 内の CAP およびストレージ容量も更新されるため、続いて LSM を監査するようにしてください。

- `config acs <acs_id>`

ACS (SL8500 ライブラリコンプレックスなど) 内のすべてのものに関して構成を更新するには、`config acs <acs_id>` を使用します。

注:

`config acs <acs_id>` では ACS 全体の CAP およびストレージ容量も更新されるため、続いて ACS を監査するようにしてください。

メディア検証

メディア検証では、SLConsole または StorageTek Tape Analytics (STA) を使用して、すべての T10000 テープカートリッジタイプを検証できます。T10000C および T10000D ドライブの専用「メディア検証プール」が使用されます。メディア検証プール内のドライブは ACSLS には使用できません。ドライブが以前に ACSLS に構成されていた場合は、ACSLS がそれらにアクセスしようとする、ライブラリは「インストールされていないドライブ」として報告します。

メディア検証プールにドライブを追加する

ドライブが ACSLS の制御から削除され、メディア検証ドライブプールに追加された場合、そのドライブを含んでいる LSM (SL8500 レールまたは SL3000 ライブラリ) は、ACSLS に対してまず準備未完了状態になり、次に準備完了状態になります。ACSLS はライブラリから「構成が変更された」というメッセージも受信します。

ACSLS ホストは自動的に、ドライブをオフライン状態に更新します。ドライブがメディア検証プールに残る場合は、`config drives <panel_id>` ユーティリティを使用してドライブを ACSLS 構成から削除します。

注:

`config drives` ドライブは、ライブラリがオンラインで、ほかのドライブに対するマウントおよびマウント解除が発生しているときに実行できます。

メディア検証プールからドライブを削除する

メディア検証プールからドライブを削除したあと:

- パーティション分割されていないライブラリの場合、ドライブスロットはすぐに ACSLS で使用可能になります。
- パーティション分割されているライブラリの場合、検証プールから削除されたドライブスロットはパーティションに割り当てられません。ドライブスロットをパーティションに割り当てるには、SL コンソールを使用します。

ドライブがメディア検証ドライブプールから削除され、ホストで使用可能になると、ライブラリは ACSLS に「構成が変更された」というメッセージを送信します。

ドライブが ACSLS 構成に含まれている場合は、ドライブをオンラインに変更してください。ドライブが ACSLS 構成に含まれていない場合は、`config drives <panel_id>` ユーティリティーを使用して追加してください。

注:

`config drives` ドライブは、ライブラリがオンラインで、ほかのドライブに対するマウントおよびマウント解除が発生しているときに実行できます。

第9章 カートリッジの管理

ACSLS は高度なカートリッジ管理機能を提供します。これらの機能はいくつかの方法で提供されます。

- 自動的 (失われたカートリッジの回復など)。
- デフォルトで有効 (不在カートリッジと取り出し済みカートリッジに関する情報の保持など)。
- 顧客による定義 (カートリッジが監査によってデータベースに追加される場合、または CAP を介して挿入される場合の、ボリューム属性の割り当てなど)。

適切なカートリッジ管理機能を使用することで、ACSLS が提供するものの価値が高まります。

カートリッジ管理は次で構成されます。

- 「LSM の設定」
- 「CAP の使用」
- 「カートリッジの挿入」
- 「カートリッジの取り出し」
- 「CAP の回復」
- 「新しい再アクティブ化されたカートリッジへのポリシーの自動適用」
- 「クリーニングカートリッジ」
- 「スクラッチカートリッジの管理」
- 「不在カートリッジと取り出し済みカートリッジのサポートの使用」
- 「不在、取り出し済み、および見つからないカートリッジ」
- 「カートリッジの回復」
- 「見つからないカートリッジ」
- 「不在カートリッジと取り出し済みカートリッジ」
- 「手動ボリューム削除ユーティリティーの使用」
- 「期限切れカートリッジの特定」

- 「アクティブな LSM からの、もっとも長い間アクセスされていないカートリッジの移動」
- 「無効になっている LSM のドライブへのカートリッジの手動ロード」

LSM の設定

カートリッジは、ライブラリがオフラインのときにセルに手動で配置するか、CAP を介してライブラリに挿入できます。

ライブラリと ACSLS を適切に機能させるための重要な要件は、マウント解除、パススルー、および取り出し操作に対応させるために、各 LSM のいくつかの空きセルが使用可能であることです。各 LSM に取り付けられているテープドライブごとに、少なくとも 1 つの空きセルを予約する必要があります。

LSM の空きセル数を確認するには、次のコマンドを発行します。

```
query lsm lsm_id
```

SL8500 では、各レーンが LSM として定義されます。

CAP の使用

次のセクションでは、CAP のタイプ、状態、モード、および優先順位を説明します。

CAP のタイプ

CAP の各タイプには、カートリッジをロードするための標準の容量と方法があります。1 つの LSM に複数のタイプの CAP がある場合があります。次の表に、サポートされている CAP のタイプ、識別子と容量、およびロードの方法を示します。

表9.1 CAP のタイプ

CAP のタイプ	識別子と容量	ロードの方法
StorageTek VTL	CAP 0。20 個のカートリッジを保持します	仮想ボリュームは Audit を使用して検出されます。「 VTL の動作 」を参照してください。
SL3000	CAP 6 およびオプションで CAP 1-5、CAP 7-10。それぞれが 26 個のカートリッジを保持します。	CAP 内にロードされている 2 つの取り外し可能なマガジンのそれぞれに、13 個のカートリッジが配置されます。
SL8500 回転式	CAP 0 およびオプションで CAP 1。それぞれが 39 個のカートリッジを保持します。	CAP 内にロードされている 3 つの取り外し可能なマガジンのそれぞれに、13 個のカートリッジが配置されます。

CAP のタイプ	識別子と容量	ロードの方法
SL8500 バルク	CAP 0 および CAP 1。それぞれが 33 個または 36 個のカートリッジを保持します	CAP 内にロードされている 3 つの取り外し可能なマガジンのそれぞれに、11 個または 12 個のカートリッジが配置されます。「一括 CAP」を参照してください。
SL500	CAP 0。5 個から 25 個までのカートリッジを保持します	CAP 内にロードされている取り外し可能なマガジンに、5 個のカートリッジが配置されます。基本モジュールに 1 つのマガジン。CAP が含まれている拡張モジュールに 2 つのマガジン。
L180	CAP 0、10 個のカートリッジを保持します。	CAP 内にロードされている 2 つの取り外し可能なマガジンのそれぞれに、5 個のカートリッジが配置されます。
L700	CAP 0 およびオプションで CAP 1。それぞれが 20 個のカートリッジを保持します。	CAP 内にロードされている 4 つの取り外し可能なマガジンのそれぞれに、5 個のカートリッジが配置されます。
拡張 (4410 および 9310)	CAP 0 および CAP 1。それぞれが 40 個のカートリッジを保持します。	CAP 内にロードされている取り外し可能なマガジンにカートリッジが配置されます。
9360	CAP 0 は 20 個のカートリッジを保持します。オプションの CAP 1 は 30 個のカートリッジを保持します。	CAP 内にロードされている取り外し可能なマガジンにカートリッジが配置されます。
優先 (PCAP)	CAP 2。1 つのカートリッジを保持します。	カートリッジは一度に 1 つずつ、CAP に直接挿入します。
9710 または 9740 CAP	CAP 0、14 個のカートリッジ、または 10 個のカートリッジを格納できるマガジンを保持します。	カートリッジは、CAP セルに直接ロードするか、CAP 内にロードされている取り外し可能なマガジンに配置します。
9714、9730、または 9738 CAP	CAP 0、1 つのカートリッジを保持します	カートリッジは単一セル CAP に直接ロードされます。

CAP のタイプ	識別子と容量	ロードの方法
レガシー 4400	CAP。21 個のカートリッジを保持します。	カートリッジは CAP のセルに直接ロードされます。

CAP の状態

CAP の状態は、カートリッジの挿入と取り出しが可能かどうかを決定します。次の表では、有効な CAP の状態について説明します。CAP の状態を決定する手順については、「[CAP 情報の表示](#)」を参照してください。デバイス状態の変更の詳細については、コマンド、「[query pool](#)」を参照してください。

注:

SL8500 ライブラリに関する詳細については、「[SL8500 内部アドレスと ACSLS アドレスについて](#)」を参照してください。SL500 ライブラリに関する詳細については、「[SL500 CAP の動作](#)」を参照してください。

表9.2 CAP の状態

状態	説明	リクエストを処理する方法
<i>online</i>	正常な動作状態です。	すべてのリクエストが受け入れられて、処理されます。
<i>offline</i>	CAP は論理的に無効になっています。	すべてのリクエストが拒否されます。
<i>offline-pending</i>	移行状態。CAP がオンラインからオフラインになるときに発生します。	すべての新しいリクエストが拒否されます。 現行および保留中のリクエストは、完了するまで処理されます。
<i>diagnostic</i>	CAP はクライアントアプリケーションからの干渉なしで診断アクティビティに使用できます。	クライアントアプリケーションからのリクエストは拒否されます。 <i>md_proc</i> からのリクエストは処理されます。
<i>recovery</i>	移行状態。CAP がオフラインからオンラインになるときに発生します。	新しいリクエストは拒否されます。

CAP のモード

CAP のモードは、カートリッジの挿入と取り出しのための CAP の使用方法を制御します。次の表では、有効な CAP モードについて説明します。CAP のモードを決定する手順について

は、「[CAP 情報の表示](#)」を参照してください。CAP のモードの変更の詳細については、コマンド、「[query cap](#)」を参照してください。

ヒント: CAP のモードは、CAP の使用中は変更できません。つまり、手動または自動のいずれかの挿入操作中にドアが開いている場合は、挿入操作を完了するまでそのモードは変更できません。

表9.3 CAP のモード

モード	説明	挿入/取り出しへの影響
手動	使用中ではないときは CAP はロックされています。これはすべての複数カートリッジ CAP の初期モードです。	<p>コマンドを明示的に発行したあとでのみ、カートリッジを挿入したり取り出したりできます。コマンドに <code>cap_id</code> を指定するか、以前に定義した CAP の優先順位に基づいた CAP の自動選択を ACSLS に許可します。</p> <p>一部のクライアントアプリケーションでは、CAP を手動モードにする必要があります。テープ管理システムのドキュメントを参照してください。</p>
自動	<p>使用中ではないときは CAP はロック解除されています。これはすべての優先順位の CAP の初期モードです。</p> <p>パーティション分割されたライブラリには、CAP のモードを自動に設定することはできません。この例外は専用の CAP (1 つのパーティションにのみ割り当てられている) であり、SL3000 では自動モードに設定できます。</p> <p>SL8500 のアクセスドアが開いたり閉じたりするとき、SL8500 は CAP をロックされた状態にします。CAP がロックされている場合、自動モードの挿入には使用できません</p>	<p><code>enter</code> コマンドを明示的に発行せずに、カートリッジを挿入できます。挿入は、CAP のドアを開いたとき、カートリッジを内部に配置したとき、および CAP を閉じたときに開始されます。</p> <p><code>cancel</code> コマンドを使用して、進行中の自動挿入操作をキャンセル (<code>cancel</code>) することはできません。進行中の自動挿入を終了するには:</p> <p>CAP のドアが開いている場合は、すべてのカートリッジを取り外して、ドアを閉じます。</p> <p>CAP のドアが閉じていて、カートリッジがライブラリに移動中の場合は、残りのカートリッジをライブラリに挿入できるようにする必要があります。そのあと <code>enter</code> が終了します。</p> <p>カートリッジを取り出すには、<code>eject</code> コマンドを明示的に発行する必要があります。コマンドに <code>cap_id</code> を指定するか、以前に定義した CAP の優先順位に基づいた CAP の自動選択を ACSLS に許可できます。</p>

モード	説明	挿入/取り出しへの影響
		ACSLs に自動モードの CAP が表示されていても、ロックされていて開かないため自動挿入に使用できない場合: ACSLs と SL8500 を同期してから、CAP を自動挿入に戻します。
		<code>set cap mode manual cap_id</code>
		<code>set cap mode automatic cap_id</code>

CAP の優先順位

CAP の優先順位は、CAP リクエストが CAP ID にアスタリスクを指定したときに、ACSLs が CAP を自動的に選択する方法を指定します。次の表では、CAP の優先順位とそれらの結果について説明します。CAP の優先順位を決定する手順については、「[CAP 情報の表示](#)」を参照してください。CAP の優先順位の変更の詳細については、「[query cap](#)」を参照してください。

表9.4 CAP の優先順位

優先順位	結果
16 (もっとも高い)	最初に使用される
15 (次に高い)	次に使用される
-	
1 (もっとも低い)	最後に使用される
0	自動的に選択されることはない (すべての CAP の初期の優先順位)

CAP の優先順位と自動 CAP 選択は、次のコマンドに適用されます。

- `audit`
- `eject`
- `enter`
- `venter`

これらのコマンドを、`cap_id` のすべてまたは一部にアスタリスク (*) を含めて入力すると、ACSLs は自動的に、リクエストで指定された各 ACS または LSM に対してもっとも高いゼロ以外の優先順位を持つ使用可能な CAP を選択します。

例:

- `audit * server`

ACSLs は各 ACS のもっとも高いゼロ以外の優先順位の CAP を選択します。

- `enter 0, 1, *`

ACSLs は LSM 0,1 のもっとも高いゼロ以外の優先順位の CAP を選択します。

CAP 情報の表示

次に、`query cap` コマンドを使用して現在の CAP 情報を表示するためのいくつかのガイドラインを示します。

- 選択された CAP についての情報を表示するには、次を入力します。

```
query cap cap_id cap_id ...
```

- ライブラリ内のすべての CAP の情報を表示するには、次を入力します。

```
query cap all
```

カートリッジの挿入

カートリッジの挿入を手動にするか自動にするかを選択できます。

- カートリッジを手動で挿入するには、`enter` コマンドを発行する必要があります。これにより CAP のロックが解除されるため、カートリッジを挿入できるようになります。
- 自動挿入は、自動モードになっている CAP を開くことによって開始されます。CAP が自動モードのときは、挿入コマンドを発行する必要はありません。

次の手順で挿入プロセスについて説明します。

1. 挿入を開始すると、CAP のロックが解除されて予約されます。これは別のホストでは使用できません。
2. CAP を開いたあと、CAP にカートリッジを配置して、CAP を閉じます。これで、CAP はロックされました。

ACSLs ライブラリロボットが CAP 内のカートリッジを検査/監査します。挿入するすべてのカートリッジに、この ACSLS サーバーによってすでに管理されているほかの `vol_ids` と重複しない、有効な外部ラベルが付いている必要があります。

注:

仮想挿入では、一部のライブラリ内にラベルのないカートリッジを挿入できます。

3. ACSLS が、有効なカートリッジにライブラリ内のホームセルを割り当て、それらを割り当てられたホームセルの位置に移動します。

重複するカートリッジと外部ラベルなしのカートリッジは CAP 内に残り、削除される必要があります。

4. 完了すると、CAP のロックが解除されるため、追加のカートリッジを挿入できます。
 - CAP が自動モードの場合は、自動挿入が完了し、CAP の予約が解除されて使用可能になります。
 - これが手動挿入の場合、CAP は手動挿入用に予約されたままになります。自動挿入を終了するには、`cancel` コマンドを使用するか、挿入が開始された `cmd_proc` で `Ctrl + c` によってキャンセルします。

`enter` コマンドの追加情報については、「[enter](#)」を参照してください。

注:

カートリッジの追跡が有効になっている場合は、イベントログにすべてのカートリッジの挿入が記録されます。

表9.5 カートリッジの挿入コマンド

タスク	コマンド
自動モードでのカートリッジの挿入	<code>set cap mode automatic cap_id</code>
手動モードでのカートリッジの挿入	<code>enter cap_id</code>
仮想ラベルが付いたカートリッジの挿入 (<code>venter</code>)	<code>venter cap_id vol_id vol_id</code>

ラベルがない、または読み取れないカートリッジは、ACSLS が管理できないため、LSM のドアを開いてこれらのカートリッジをストレージセル内に配置しないでください。監査中に ACSLS は、ストレージセル内に配置されたラベルがない、または読み取れないカートリッジを取り出します。

挿入リクエストの終了

これらの手順を使用して、現行または保留中の手動挿入または仮想挿入を、終了またはキャンセルします。

`cancel` コマンドを使用して、進行中の自動挿入操作をキャンセルすることはできません。進行中の自動挿入を終了するには:

- CAP のドアが開いている場合は、すべてのカートリッジを取り外して、ドアを閉じます
- CAP のドアが閉じていて、カートリッジがライブラリに移動中の場合は、残りのカートリッジをライブラリに挿入できるようにする必要があります。そのあと挿入が終了します。

手動挿入をキャンセルするには:

1. 現行および保留中のすべてのライブラリのアクティビティを表示します。

```
query request all
```

2. キャンセルしたい enter/venter リクエストの *request_id* をメモします。
3. *cmd_proc* から、次を入力します。

```
cancel request_id
```

ここで、*request_id* はキャンセルしたいリクエストの識別子です。

4. CAP のロックが解除されるのを待ってから、CAP を開き、すべてのカートリッジを取り外します。

cmd_proc は、キャンセルリクエストが受信される前にライブラリに挿入されたカートリッジの数を示すメッセージを表示します。これらのカートリッジは引き続き ACSLS によって制御されます。

「enter」を参照してください。

カートリッジの取り出し

ライブラリからカートリッジを取り出すには、*eject* コマンドを発行する必要があります。

次の手順で取り出しプロセスについて説明します。

1. 取り出しを開始したあと、CAP はロックされます。これは別のホストでは使用できません。
2. ロボットが指定されたカートリッジを指定された CAP に配置し、次に ACSLS が、カートリッジが格納されていたセル位置をほかのカートリッジが使用できるようにします。
3. CAP を開き、CAP からすべてのカートリッジを取り外し、CAP を閉じます。そのあと、ACSLS が CAP を調べて空であることを確認します。これで、CAP が挿入または監査などの別の操作に使用できるようになります。

eject コマンドで CAP いっぱいのカートリッジよりも多く指定する場合は、いっぱいになったときに CAP を空にして CAP を閉じると、すべてのカートリッジが取り出されるまで ACSLS が取り出しプロセスを続行します。

`eject` コマンドの追加情報については、「[eject](#)」を参照してください。また、「[ejecting.sh](#)」も参照してください。

ボリューム統計の収集が有効になっている場合は、`acsss_stats.log` にすべてのカートリッジの取り出しが記録されます。「[一般的な製品動作変数の設定](#)」を参照してください。

CAP の回復

このセクションでは、CAP の回復について説明します。

一般的な CAP の回復手順

一般的な CAP の回復手順を次に示します。

CAP の回復を行う前に挿入と取り出しを完了する

可能な場合は、キャンセルして CAP を回復しようとするのではなく、挿入または取り出しを完了させてください。これにより、複雑さが減り、CAP のハングのリスクが減ります。

- CAP いっぱいのカートリッジの挿入を完了してから、キャンセルすることで手動挿入を終了します。(自動モードでは、CAP は一度に 1 つの CAP いっぱいのカートリッジのみを挿入します。)
- 可能な場合は、取り出しコマンドに指定されたすべてのカートリッジを取り出してください。そうでない場合は、取り出しをキャンセルしようとする前に、いっぱいになった CAP のカートリッジを ACSLS に取り出させて CAP を空にしてください。

ハングした CAP を強制的にオフラインにしてからオンラインに変更することで回復する

CAP を回復するには、強制的にオフラインに変更する必要があります。CAP を強制的にオフラインに変更してからオンラインに戻すことで CAP が回復し、通常は、CAP を使用しているハングした `enter` または `eject` も終了します。

1. CAP を強制的にオフラインに変更します。

```
vary cap cap_id offline force
```

現在のロボットのリクエストのみが完了し、そのあとすぐに CAP がオフラインになります。保留中のリクエストは破棄され、新しいリクエストは拒否されます。

ハングした手動 `enter` または `eject` は、通常はキャンセルされます。

2. まだアクティブの場合は、*enter* または *eject* リクエストをキャンセルします。

enter または *eject* リクエストがまだアクティブかどうかを確認するには:

```
query request all
```

enter または *eject* がまだアクティブな場合は、次のコマンドを入力してキャンセルします。

```
cancel request_id
```

3. CAP をオンラインに戻します。

```
vary cap cap_id online
```

これにより CAP が回復し、ほかのリクエストに使用できるようになります。

アクセスドアを開いたあとの CAP の回復

ACSL S は、SL8500 または SL3000 のアクセスドアを開いてから閉じたり、SL8500 または SL3000 を再初期化したりしたあとに、自動挿入モードの CAP のロックを解除するようになりました。

SL8500 または SL3000 ライブラリの再初期化のあとに CAP のロックが解除されて、それを回復する必要がある場合は、次の適切な手順に従って CAP を回復してください。

自動挿入に使用する CAP のロックが解除されない

自動挿入のためにロックが解除されない CAP を回復するには、ACSL S とライブラリ両方の間で CAP の状態を同期する必要があります。

1. CAP モードを手動に設定して、自動挿入モードを終了します。

```
set cap mode manual cap_id
```

2. CAP を自動モードに設定し直します。

```
set cap mode automatic cap_id
```

手動挿入に使用する CAP のロックが解除されない

手動挿入のためにロックが解除されない CAP を回復するには、ACSL S とライブラリ両方の間で CAP の状態を同期する必要があります。

1. CAP を強制的にオフラインに変更します。

```
vary cap cap_id offline force
```

2. CAP をオンラインに戻します。

```
vary cap cap_id online
```

3. 手動挿入を再開します。

```
enter cap_id
```

取り出しに使用する CAP のロックが解除されない

取り出しを行っていた CAP を回復するには、ロックされた CAP 内に残っているカートリッジを取り外し、ACSL5 とライブラリ両方の間で CAP の状態を同期する必要があります。

1. CAP 内のすべてのカートリッジを取り外します。
 - a. CAP を強制的にオフラインに変更 (*vary*) します。

```
vary cap cap_id offline force
```

- b. CAP をオンラインに変更 (*vary*) します。

```
vary cap cap_id online
```

2. 次のいずれかを選択します。

CAP が自動モードになっている場合:

- a. CAP モードを手動に設定して、自動挿入モードを終了します。

```
set cap mode manual cap_id
```

- b. CAP を自動モードに設定します。これにより CAP のロックが解除されます。

```
set cap mode automatic cap_id
```

- c. CAP を開き、CAP 内に残っているすべてのカートリッジを取り外します。

CAP が自動モードでない場合:

- a. 手動 *enter* を開始します。

```
enter cap_id
```

- b. CAP 内に残っているすべてのカートリッジを取り外します。

- c. 挿入をキャンセルします。

挿入を待機している `cmd_proc` で `Ctrl + c` を使用するか、`enter` リクエスト ID をキャンセルします。

3. 取り出しを再開します。

```
enter cap_id vol_id | volrange...
```

L1400、L700、L700e、または L180 ライブラリ内の CAP のロックを解除するための回復手順

L1400、L700、L700e、または L180 ライブラリで挿入または取り出しに使用する CAP のロックが解除されない場合は、ライブラリで IPL を実行して CAP を回復できます。CAP を回復するには、次の適切な手順に従ってください。

手動挿入に使用する CAP のロックが解除されない

手動挿入のためにロックが解除されない CAP を回復するには:

1. `enter` をキャンセル (`cancel`) します。

挿入が完了するのを待機している `cmd_proc` で `Ctrl + c` を使用するか、`enter` リクエスト ID をキャンセル (`cancel`) します。

2. オペレータパネルの「RESET」ボタンを押して、ライブラリを再 IPL します。
3. ライブラリの初期化が終了したら、別の挿入を開始します。

自動挿入に使用する CAP のロックが解除されない

自動挿入のためにロックが解除されない CAP を回復するには:

1. CAP モードを手動に設定し直して、自動挿入モードを終了します。

```
set cap mode manual cap_id
```

2. オペレータパネルの「RESET」ボタンを押して、ライブラリを再 IPL します。
3. ライブラリの初期化が完了したら、CAP を自動モードに設定し直します。

```
set cap mode automatic cap_id
```

取り出しに使用する CAP のロックが解除されないため、空にできない

取り出しのためにロックが解除されない CAP を回復するには (CAP がいっぱいになったあと、またはすべてのボリュームが取り出されたあと):

1. ライブラリへのアクセスドアを開き、すべてのカートリッジを CAP から取り外して、アクセスドアを閉じます。
2. オペレータパネルの「RESET」ボタンを押して、ライブラリを再 IPL します。

ライブラリの再 IPL により、ACSLs が取り出しを「ライブラリ障害」で終了します。

3. オプションで、ライブラリを監査します。

ライブラリの初期化が完了したあとに監査を実行することは良いアイデアですが、必須ではありません。

4. すべてのカートリッジが取り出されていない場合は、別の取り出しを開始します。

新しい再アクティブ化されたカートリッジへのポリシーの自動適用

このセクションでは、新しい再アクティブ化されたカートリッジにポリシーを自動的に適用する方法について説明します。

クリーニングカートリッジ属性の自動割り当て

最新のクリーニングカートリッジには、クリーニングカートリッジ専用予約されているメディアタイプのラベルが付けられています。たとえば、T10000 下位互換のクリーニングカートリッジにはメディアドメインとタイプ「CL」のラベルが付けられ、LTO ユニバーサルクリーニングカートリッジには「CU」のラベルが付けられています。

ACSLs は、これらのメディアドメインとタイプを持つカートリッジがクリーニングカートリッジ専用であると認識するため、監査、挿入、またはカートリッジ回復によってこれらのカートリッジが追加されると、クリーニングカートリッジ属性を自動的に設定します。これには、それらをクリーニングカートリッジとして識別することと、それらの最大クリーニング使用回数を設定することが含まれます。

watch_vols ポリシー

`watch_vols` ユーティリティでは、データベースに追加されたり監査によって再アクティブ化されたりしたカートリッジに、それらが挿入または再挿入されたときにも、自動的に属性を割り当てることができます。ポリシーは `vol_attr.dat` ファイルに指定され、`vol_id` または `vol_range` によって選択されます。このユーティリティは、自動的に次を実行できます。

- `vol_id` の範囲、または `vol_attr.dat` ポリシーテーブルにリストされている特定のボリュームに基づいて、ボリュームの所有権を割り当てます。
- カートリッジをスクラッチプールに割り当てます。
- 新しい再アクティブ化されたカートリッジを、特定の LSM に移動します。

- カートリッジを論理ライブラリに割り当てます。

詳細については、「[watch_vols](#)」を参照してください。

クリーニングカートリッジ

テープドライブは、読み取り/書き込み記録ヘッドから染み汚れと付着物を取り除くため、定期的にクリーニングする必要があります。ドライブ制御ユニットは、テープが各ドライブを通過する回数を追跡し、ドライブでクリーニングが必要になったときに ACSLS にメッセージを送信します。

クリーニングカートリッジの詳細については、次を参照してください。

- 「[ACSLS による自動クリーニング](#)」
- 「[クリーニングカートリッジの最大使用回数](#)」
- 「[クリーニングカートリッジの挿入](#)」
- 「[使用済みのクリーニングカートリッジの取り出し](#)」
- 「[ドライブの手動クリーニング](#)」
- 「[ファイバ接続ライブラリ上のカートリッジのクリーニング](#)」

ACSLS による自動クリーニング

ACSLS は、TCP/IP またはシリアル (HLI) 接続のライブラリ (SL8500、SL3000、および 9310) には自動クリーニングを実行できますが、ファイバまたは SCSI 接続のライブラリ (SL150、SL500、および L700) には実行できません。

自動クリーニングが有効になっている場合、ACSLS は必要なときにテープドライブにクリーニングカートリッジを自動的にマウントします。`AUTO_CLEAN` 動的変数が `TRUE` (デフォルト) に設定されている場合、自動クリーニングは有効になっています。

最新のテープドライブは、必要に応じてクリーニングをリクエストします。ドライブがライブラリに通知し、それにより ACSLS にメッセージが転送されます。ACSLS は、ドライブをクリーニングする必要があることを記録します。ACSLS がドライブに対する次のマウントリクエストを処理すると、そのあとにマウントとクリーニング操作が行われます。これには、互換性のあるクリーニングカートリッジを選択し、クリーニングカートリッジをマウントし、クリーニングカートリッジをマウント解除してから、元のマウントリクエストに指定されているデータカートリッジのマウントに進むことが含まれます。

ACSLS は、クリーニング操作中に使用済み (使い果たされた) クリーニングカートリッジのマウントなどの回復可能な問題が発生した場合は、別のクリーニングカートリッジを選択して

クリーニング操作を再試行します。`AUTO_CLEAN_RETRY_LIMIT` 動的変数は再試行の回数 (デフォルトの 1 回の再試行と 0-5 回の再試行) を制御します。この変数を表示および変更するには、`acsss_config` を使用して「General Product Behavior Variables」を選択します。

`UNIFORM_CLEAN_USE` 動的変数は、クリーニングカートリッジの選択に使用する方法を定義します。オプションは次のとおりです。

- `VOLID_SORT` – `vol_id` の順。次を使用する前に 1 つのクリーナーを使い果たします。
- `LEAST_USED` – 使用回数の順。使用回数を均等に分散します。
- `MOST_CAPACITY` – 残りの使用回数の順。すべてのクリーナーを同時に使い果たします。

デフォルトは `VOLID_SORT` です。この変数を表示および変更するには、`acsss_config` を使用して「General Product Behavior Variables」を選択します。

ACSLS による自動クリーニングの詳細については、次を参照してください。

- [「クリーニングカートリッジの最大使用回数」](#)
- [「クリーニングカートリッジの挿入」](#)
- [「CSI チューニング変数の設定」](#)の「`AUTO_CLEAN`」
- [「CSI チューニング変数の設定」](#)の「`AUTO_CLEAN_RETRY_LIMIT`」。

クリーニングカートリッジの最大使用回数

さまざまなクリーニングカートリッジタイプのそれぞれには、使い果たされた (期限切れまたは使用済みである) ことがドライブから報告されるまでの最大使用回数があります。この最大使用回数は、クリーニングカートリッジのタイプに応じて異なります。ACSLS がクリーニングカートリッジを追加すると、カートリッジの最大使用回数が ACSLS データベースに記録されます。ACSLS は、クリーニングの `access_count` (カートリッジのマウントなどが行われた回数) が `max` 使用回数より少ない場合のみ、自動クリーニング用のクリーニングカートリッジを選択します。テープドライブからクリーニングカートリッジが使い果たされた (使用済みである) ことが報告されると、ACSLS は `max` 使用回数より多いアクセス数を設定します。

ACSLS がクリーニングカートリッジに自動的に設定する最大使用回数は、カートリッジがサポートする実際のクリーニング使用回数より大きい数です。これは、一部のアプリケーションが、ドライブがクリーニングをリクエストしていなくても、クリーニングカートリッジのマウントをスケジュールするためです。ドライブでクリーニングの準備ができていない場合は、ヘッドが早く摩耗してしまわないように「擬似クリーニング」を行うことがあります。これは、ドライブのアクセス数が、実際にはクリーニングカートリッジを使用しなくても増分されていることを意味します。`max` 使用回数の値をより大きくすると、ドライブから使用済みと報告されるまで、これらのカートリッジを使用できます。

クリーニングカートリッジの手動の定義

`set clean` コマンドを使用して、クリーニングカートリッジを定義し、その最大使用回数を設定できます。

```
set clean max_usage vol_id | volrange
```

ここでは:

- `max_usage` は、ACSL5 がカートリッジをクリーニングするカートリッジの選択を停止するまでにクリーニングカートリッジが使用される回数です。
- `vol_id | volrange` は、クリーニングカートリッジまたはカートリッジの範囲を指定します。

`set clean` を使用して次を実行します。

- クリーニングカートリッジの最大使用回数を変更します。

たとえば、クリーニングの必要がなかったドライブにクリーニングカートリッジが手動でマウントされ、`access_count` が増分しましたが、行われたのは「擬似クリーニング」だけでした。クリーニングカートリッジから最大の使用回数を得るには、より大きい `max_usage` を設定してください。

```
set clean max_usage vol_id|volrange
```

- カートリッジのクリーニングカートリッジ属性をオフに設定します。たとえば、誤ってデータカートリッジをクリーニングカートリッジとして定義した場合は、カートリッジのクリーニングカートリッジ属性をオフに設定して、カートリッジをデータカートリッジとして再定義します。

```
set clean off vol_id|volrange
```

クリーニングカートリッジのモニタリング

使用済みのクリーニングカートリッジを取り出して、ライブラリ内のクリーニングカートリッジをモニタリングする必要があります。必要に応じて新しいクリーニングカートリッジを挿入します。

- すべてのクリーニングカートリッジを表示するには:

```
query clean all
```

- ACS 内の 1 つの `media_type` のすべてのクリーニングカートリッジを表示するには、`display` コマンドを使用します。

```
display volume * -home acs,*,*,* -media media_type
```

- カートリッジの最大クリーニング使用回数と現在の使用回数を表示するには:

```
display volume * -home acs,*,*,* -media media_type -f vol_id acs lsm
media max_use access_count
```

- ACS 内のすべてのクリーニングカートリッジを、その最大クリーニング使用回数および現在の使用回数とともに表示するには:

```
display volume CLN* -home acs,*,*,* -f acs lsm type media max_use
access_count
```

- 使用済みのすべてのクリーニングカートリッジを表示するには (これらのカートリッジは、取り出して新しいクリーニングカートリッジと交換する必要があります):

```
display volume * -spent_clean
```

関連項目:

- [「クリーニングカートリッジの挿入」](#)
- [「使用済みのクリーニングカートリッジの取り出し」](#)

クリーニングカートリッジの挿入

クリーニングカートリッジを挿入するときは、必ず次の手順を完了してください。

- メディアタイプがライブラリ内のドライブタイプと互換性のあるクリーニングカートリッジを使用してください。ACSLS が自動的に、クリーニング操作ごとに正しいタイプのカートリッジを選択します。

ドライブタイプと互換性のあるクリーニングカートリッジを確認するには、「ACSLS 製品情報」マニュアルのドライブとメディアの互換性の表を参照するか、`drive_media.sh` ユーティリティを使用してください。

- ライブラリ内のドライブタイプごとに、少なくとも 2、3 のクリーニングカートリッジを定義してください。ほとんどの場所では、少なくとも 4 つのドライブごとに 1 つのクリーニングカートリッジが適切です。

ACSLS にクリーニングカートリッジを定義するには:

1. CAP の挿入を準備します。

詳細については、「[カートリッジの挿入](#)」を参照してください。

2. クリーニングカートリッジを挿入します。

`cmd_proc` は、挿入したカートリッジのカートリッジ ID を含むメッセージを表示します。

「[クリーニングカートリッジ属性の自動割り当て](#)」で説明しているように、ACSL S はクリーニングカートリッジを、それらが監査、挿入、またはカートリッジ回復によって挿入または追加されたときに自動的に定義します。これにはそれらの最大使用回数が含まれます。

使用済みのクリーニングカートリッジの取り出し

ACSL S は、クリーニングカートリッジがその最大使用回数に達したとき、またはドライブからクリーニングカートリッジが使用済みであると報告されたときに、イベントログにメッセージを記録します。ACSL S はカートリッジをライブラリ内に残しますが、それをクリーニング用には選択しなくなります。使用済みのクリーニングカートリッジを取り出して、交換品を挿入します。

使用済みのクリーニングカートリッジを取り出すには:

1. `query clean` および `display volume` を使用して、最大使用回数を超えた、または使用済みのクリーニングカートリッジを識別します。

```
query clean all
```

```
display volume * -spent_clean
```

2. クリーニングカートリッジを取り出します。

```
eject cap_id vol_id | volrange
```

ここでは:

`cap_id` は、クリーニングカートリッジを取り出すために使用される CAP を指定します。

`vol_id | volrange` は、取り出すクリーニングカートリッジの ID を指定します。

3. 使用済みのクリーニングカートリッジを取り外します。

「[クリーニングカートリッジのモニタリング](#)」を参照してください

ドライブの手動クリーニング

自動クリーニングが無効になっているとき、または動作していないときにドライブをクリーニングするには、この手順を使用します。

ドライブを手動でクリーニングするには:

1. クリーニングするドライブと互換性のあるクリーニングカートリッジタイプを確認します。

各ドライブタイプのクリーニングカートリッジのリストについては、「製品情報ガイド」を参照し、「ドライブとメディアの互換性」の表を調べてください。

2. 使用可能なクリーニングカートリッジを表示します。

```
query clean all
```

ドライブと同じ ACS 内の互換性のあるすべてのクリーニングカートリッジを表示するには、*display* コマンドを使用します。

```
display volume * -home acs,*,*,*,* -media media_type
```

カートリッジの最大クリーニング使用回数と現在の使用回数を表示するには:

```
display volume * -home acs,*,*,*,* -media media_type -f vol_id acs  
lsm media max_use access_count
```

ACS 内のすべてのクリーニングカートリッジを、それらの最大クリーニング使用回数および現在の使用回数とともに表示するには:

```
display volume CLN* -home acs,*,*,*,* -f acs lsm type media max_use  
access_count
```

3. それらのリストにあるものから互換性のあるクリーニングカートリッジを選択し、ドライブにマウントします。

```
mount vol_id drive_id
```

4. ドライブがクリーニングされ、クリーニングカートリッジがアンロードされたあと、クリーニングカートリッジをマウント解除します。

```
dismount vol_id drive_id
```

ファイバ接続ライブラリ上のカートリッジのクリーニング

ACSL S 自動クリーニングは、ファイバ接続ライブラリ内のドライブではサポートされていません。これらのドライブは、クリーニングカートリッジを手動でマウントすることによってのみ、ACSL S を使用してクリーニングできます。ただし、ファイバ接続ライブラリでは、ライブラリ GUI を使用して自動クリーニングを有効にできます。詳細については、ライブラリのドキュメントを参照してください。

ドライブがクリーニングされていないときの対処

次に、ドライブがクリーニングされていないときに試すいくつかのトラブルシューティングのヒントを示します。

自動クリーニングが有効になっていることを確認する

自動クリーニングが無効になっている場合は、ドライブにクリーニングが必要になると ACSLS がイベントログにメッセージを記録して、`cmd_proc` でクリーニングメッセージを表示します。クリーニングカートリッジを手動でマウントする必要があります。

自動クリーニングを有効または無効にするには、`acsss_config` を使用します。さらに、`acsss_config` を使用すると、選択または照会のためのクリーニングカートリッジの順序付けの方法を指定できます。

`AUTO_CLEAN` 動的変数がデフォルト設定である **TRUE** (オン) に設定されている場合、自動クリーニングは有効になっています。`AUTO-CLEAN` を確認するには、次を入力します。

```
dv_config -e AUTO_CLEAN
```

ファイバ接続ライブラリの場合、ACSLs は自動クリーニングを実行しません。

ドライブ用のクリーニングカートリッジが存在していることを確認する

すべてのクリーニングカートリッジが期限切れになっている (`max_usage` 値を超えている)、またはドライブから使用済みと報告されている場合、ACSLs はドライブをクリーニングせずに元のマウントリクエストを実行します。そのマウントと、クリーニングされていないドライブへの以降の各マウントについて、ACSLs がイベントログにメッセージ 376 N 「*Drive drive_id: No Cleaning cartridge available*」をポストします。「[クリーニングカートリッジの手動の定義](#)」の説明に従って、ドライブタイプと互換性のあるクリーニングカートリッジをさらに追加します。

ドライブがクリーニングされていない場合は、ライブラリ内にそのドライブ用のクリーニングカートリッジがあることと使用回数がまだ残っていることを確認してください。

`cmd_proc` から、`display` コマンドを使用して次を表示できます。

- すべてのクリーニングカートリッジとそれらの使用回数:

```
display volume * -clean -f media access_count max_use
```

- 特定のメディアタイプのすべてのボリューム。

たとえば、すべての LTO クリーニングカートリッジを表示するには:

```
display volume * -media LTO-CLNU -f access_count max_use
```

- 使い果たされた (使用済みの) すべてのクリーニングカートリッジとそれらの使用回数:

```
display volume * -spent_clean -f media access_count max_use
```

SL8500 または SL3000 ライブラリの SL コンソールを使用して自動クリーニングが無効になっていることを確認する

SL8500 または SL3000 の自動クリーニングが動作しない問題が発生したことがある場合は、SL コンソールを使用して自動クリーニングがライブラリに対して有効になっていないことを確認してください。

ACSLs を使用して自動クリーニングを有効にすると、マウント解除後にライブラリから「drive needs cleaning」というメッセージを受信したときに、次のマウントの前にクリーニングカートリッジが自動的にマウントされます。

SL コンソールを使用してライブラリレベルで自動クリーニングを有効にしている場合は、ライブラリによって自動クリーニングが行われます。ライブラリ自動クリーニングが有効になっている場合、ライブラリは ACSLS に drive needs cleaning メッセージを送信しません。ACSLs はドライブをクリーニングする必要があることを認識できません。そのあと、ACSLs がマウント解除応答を送信する前にドライブをクリーニングするために、ライブラリがそのシステムセルのいずれかからクリーニングカートリッジをマウントしようとします。

結果として、ライブラリが自動クリーニングを実行しようとしても、システムセルにはクリーニングカートリッジがないという混乱が生じることがあります。ACSLs が通常のストレージセルのクリーニングカートリッジを管理することはありませんが、ACSLs は「drive needs cleaning」メッセージを受信しません。その結果、ドライブはクリーニングされません。

これを解決するには:

- ACSLS で自動化されたクリーニングが有効になっていても、ドライブがクリーニングされていない場合は、ライブラリでも自動クリーニングが有効になっているかどうかを確認してください。
- ライブラリで自動クリーニングが有効になっている場合は、SL コンソールを使用して無効にしてください。

SL コンソールまたはライブラリオペレータパネルを使用します。

- a. 「System Detail」タブを選択します。

- b. 「Library」を選択します。
- c. 「Auto Clean」タブを選択します。
- d. 「Configure」タブを選択します。
- e. 自動クリーニングがこのパーティション (または「Partition 1 or None」) に対して有効になっているかどうかチェックします。
- f. 自動クリーニングが有効になっている場合は、無効にします。

クリーニングカートリッジに **questionable** とマークされているかどうかをチェックする

自動クリーニングは、障害のあるクリーニングカートリッジが繰り返し選択されないよう、問題のあるカートリッジを選択しません。カートリッジに読み取り不能なラベルがあることがライブラリから報告されると、カートリッジに **questionable** とマークされます。

`display` コマンドを使用して、**questionable** とマークされたクリーニングカートリッジを特定できます。これにより、クリーニングカートリッジの ACS、LSM、タイプ、`max_use`、および `access_count` も表示されます。

```
display volume CLN* -f media_status acs lsm media_status type max_use access_count
```

questionable のステータスをクリアするには:

- カートリッジを取り出して検査し、問題がない場合はライブラリに再度挿入します。
- カートリッジを挿入すると、**questionable** のステータスがクリアされます。

スクラッチカートリッジの管理

スクラッチカートリッジには、データが含まれていないか、上書きできるデータが含まれています。ユーザーまたはアプリケーションがスクラッチカートリッジをマウントして、そのカートリッジに新しいデータを書き込みます。

スクラッチステータスを割り当てるには:

- カートリッジは、スクラッチカートリッジとして定義して、`set scratch` コマンドによってスクラッチプールに割り当てることができます。
- `watch_vols` ユーティリティは、カートリッジの `vol_id` または `volrange` に基づいて、カートリッジを自動的にスクラッチプールに割り当てることができます。「[watch_vols](#)」を参照してください。

ボリュームスクラッチステータスのクリア:

- カートリッジのスクラッチステータスは、(`mount scratch` または `normal mount` リクエストのいずれかによって) カートリッジが正常にマウントされたときにクリアされます。

注:

`set scratch` コマンドを使用して、スクラッチステータスをクリアできます。ボリュームのスクラッチステータスはボリュームがマウントされたときにクリアされますが、`pool id` はクリアされません。その結果、データボリュームがプールに割り当てられます。

`set scratch` コマンドは、次を使用してスクラッチプールにデータボリュームを割り当てるためにも使用できます。

```
set scratch off pool_id vol_id | volrange
```

ライブラリにスクラッチマウントリクエストを満たすために使用できる十分なスクラッチカートリッジがあることを確認する必要があります。詳細については、次を参照してください。

- [「ライブラリへのスクラッチカートリッジの追加」](#)
- [「スクラッチプールのリバランス」](#)

次のセクションでは、スクラッチカートリッジとスクラッチプールの管理についての追加情報を提供します。

- [「スクラッチプールおよびスクラッチカートリッジ情報の表示」](#)
- [「スクラッチプールの削除」](#)
- [「スクラッチカートリッジのマウント」](#)
- [「カートリッジのスクラッチ解除」](#)

スクラッチプールおよびスクラッチカートリッジ情報の表示

スクラッチプール情報を表示するには、次の ACSLS 関数を使用します。

- `query pool`

スクラッチプールの属性を表示します。[「query pool」](#)を参照してください。

- `query scratch`

スクラッチカートリッジの情報を表示します。[「query scratch」](#)を参照してください。

- `query mount *`

指定したスクラッチプール (および、オプションでプール内の特定のカートリッジメディアタイプ) のメディア互換カートリッジのステータスを表示します。「[query mount *](#)」を参照してください。

- カスタマイズされたボリュームレポート

選択されたスクラッチボリューム情報をレポートします。「[ロギングボリューム統計レポートの作成](#)」を参照してください。

ライブラリへのスクラッチカートリッジの追加

スクラッチカートリッジをライブラリに追加するには、この手順を使用します。

ライブラリにスクラッチカートリッジを追加するには:

1. 必要に応じて、新しいスクラッチプールを作成します。

詳細については、「[query scratch](#)」を参照してください。

2. スクラッチカートリッジをライブラリに挿入します。

詳細については、「[カートリッジの挿入](#)」を参照してください。

3. 手順 2 で挿入したカートリッジをスクラッチカートリッジとして定義し、それらをスクラッチプールに割り当てます。

これは、`watch_vols` ユーティリティの `vol_attr.dat` に定義されているポリシーを使用して、または `set scratch` を使用して実行できます。

スクラッチプールのリバランス

スクラッチカートリッジをプール間で移動することによってスクラッチプールをリバランスするには、この手順を使用します。

スクラッチプールをリバランスするには:

1. すべてのスクラッチプールの属性を表示します。

```
query pool all
```

詳細については、「[query pool](#)」を参照してください

2. リバランスするプール内のスクラッチカートリッジの ID を表示するには、`query scratch` コマンドを使用します。

詳細については、「[query scratch](#)」を参照してください

3. スクラッチカートリッジをプール間で移動するには、`set scratch` コマンドを使用します。

たとえば、カートリッジ YUMA20 から YUMA80 (現時点ではプール 5 からプール 10 に存在します) を再割り当てするには、次を入力します。

```
set scratch 10 YUMA20-YUMA80
```

詳細については、「[set scratch](#)」を参照してください。

スクラッチプールの削除

スクラッチプールを管理するために、スクラッチカートリッジがもう含まれていないすべてのスクラッチプールを削除する場合があります。共通プール (プール 0) を削除することはできません。削除できるのは空のスクラッチプールのみであることに注意してください。データまたはスクラッチカートリッジのいずれかが含まれている場合、スクラッチプールは削除できません。ただし、「[すべての空のプールの削除](#)」を使用して、すべての空のプールを削除できます (ACSLS はスクラッチまたはデータカートリッジが含まれているプールを削除しません)。

スクラッチプールを空にする

スクラッチプールを削除する前に空にするには、この手順を使用します。

スクラッチプールを空にするには:

1. データカートリッジをプールの外に移動するには、次を入力します。

```
set scratch off 0 vol_id volrange ...
```

ここで、`vol_id` または `volrange` は、共通プール (プール 0) に移動するデータカートリッジを指定します。詳細については、「[set scratch](#)」を参照してください。

2. スクラッチカートリッジをプールの外に移動するには、次のいずれかを実行します。
 - カートリッジを別のプールに移動します。
 - 「[カートリッジの取り出し](#)」を参照してください。ただし、スクラッチカートリッジを取り出すと、ACSLS はこれらのカートリッジを管理しなくなります。あとでこれらのカートリッジを使用する場合は、それらを再挿入し、スクラッチプールに割り当てる必要があります。

単一のプールの削除

単一のプールを削除するには:

```
delete pool pool_id
```

すべての空のプールの削除

`delete pool all` コマンドは、スクラッチまたはデータカートリッジを含むプールではなく、空のスクラッチプールのみを削除します。

すべての空のプールを削除するには:

```
delete pool all
```

スクラッチカートリッジのマウント

`mount scratch (cmd_proc` を使用した `mount *`) コマンドは、指定されたドライブと互換性があり、できるだけ近くにあるスクラッチカートリッジを選択し、それをドライブにマウントします。プールが指定されている場合、スクラッチカートリッジはそのプールに割り当てる必要があります。

ホームセル内または別の回復エラーでスクラッチカートリッジが見つからないために、スクラッチカートリッジのマウントが失敗した場合、ACSL S は自動的に別のスクラッチカートリッジを選択しようとしてマウントを再試行します。

スクラッチカートリッジのマウントごとのボリュームアクセス制御ポリシーを設定でき、それをマウントした ACSAPI ユーザーによって自動的に所有されます。[「ボリュームの所有権の確立」](#)を参照してください。

単一メディア環境と混合メディア環境でスクラッチカートリッジをマウントするには、次の手順を使用します。

単一メディア環境

- 指定されたプールからカートリッジをマウントするには:

```
mount * drive_id pool_id
```

指定されたプールから使用できるカートリッジがなく、プールが「overflow」に設定されていた場合、ACSL S は共通プール (プール 0) からカートリッジを選択します。

- 共通プールからカートリッジをマウントするには:

```
mount * drive_id
```

混合メディア環境

- 指定したプールから指定したメディアタイプのスクラッチカートリッジをマウントするには:

```
mount * drive_id pool_id media media_type
```

指定されたプールから使用できるカートリッジがなく、プールが *overflow* に設定されていた場合、ACSLs は共通プール (プール 0) から指定されたメディアタイプのカートリッジを選択します。

- スクラッチ優先順位で決定されたメディアタイプを持つ指定されたプールからスクラッチカートリッジをマウントするには:

```
mount * drive_id pool_id media *
```

指定されたプールから使用できるカートリッジがなく、プールが *overflow* に設定されていた場合、ACSLs は、定義されているスクラッチ優先順位に従って、共通プール (プール 0) からカートリッジを選択します。

- 指定されたメディアタイプの共通プールからカートリッジをマウントするには:

```
mount * drive_id media media_type
```

- スクラッチ優先順位で決定されたメディアタイプを持つ共通プールからカートリッジをマウントするには:

```
mount * drive_id media *
```

カートリッジのスクラッチ解除

スクラッチカートリッジは、マウント時のデータカートリッジステータスに自動的に再割り当てされます。

エラーでスクラッチされたカートリッジを「スクラッチ解除」する (それらをデータカートリッジステータスに戻す) には、この手順を使用します。

カートリッジをスクラッチ解除するには:

1. `query pool` および `query scratch` コマンドを使用して、スクラッチ解除するカートリッジの ID およびプール ID を表示します。

詳細については、「[query pool](#)」および「[query scratch](#)」を参照してください。

2. 選択されたカートリッジをスクラッチ解除するには、次を入力します。

```
set scratch off 0 vol_id volrange ...
```

ここで、*vol_id* または *volrange* は、スクラッチモードから変更して共通プール (プール 0) に移動するカートリッジを指定します。詳細については、「[set scratch](#)」を参照してください。

不在カートリッジと取り出し済みカートリッジのサポートの使用

ACSL S での不在カートリッジのサポートは、ライブラリ内で見つからないカートリッジを削除するのではなく、それらに不在とマークします。これらのカートリッジがあとでライブラリ内で見つかる、ACSL S は、それらをデータベースに再度追加するのではなく、それらをアクティブステータスに変更します。再アクティブ化により、アクセス数と、プール、ボリュームアクセス制御所有権、ロックなどの設定が保持されます。

同様に、取り出し済みカートリッジのサポートは、カートリッジが取り出されたときのカートリッジ情報を保持します。カートリッジは再挿入されたときに再アクティブ化されます。

不在ボリュームと取り出し済みボリュームのサポートは、*ABSENT_VOLUME_RETENTION_PERIOD* がゼロ以外の日数に設定されているときに有効になっています。デフォルト値は 5 日です。

不在カートリッジと取り出し済みカートリッジのサポートのその他の面として、次が含まれます。

- 手動ボリューム削除 (*del_vol*) ユーティリティは、*-d* オプションが指定されていない限り、ボリュームを不在として保持します。このオプションが指定されている場合、ボリュームは、不在または取り出し済みというステータスの期限切れを待たずに削除されます。
- ACSLS は、SL3000 および SL8500 ライブラリで失われたカートリッジの場所を照会します。
- ACSLS は、ライブラリ内の予期された場所で見つからないボリュームを検索することで、ボリュームの回復を向上させます。ACSL S は、ボリュームを自動的に削除する代わりに、記録されたすべての場所を検索します。
- クライアントは、*ENABLE_STATUS_VOLUME_ABSENT* および *ENABLE_STATUS_VOLUME_MISSING* 構成設定を使用して、それらが不在、取り出し済み、および見つからないというステータスとして ACSAPI を介して報告されるかを指定できます。
- *-i* オプションを指定した *volrpt* ユーティリティは、不在または取り出し済みというステータスのボリュームレコードを報告します。デフォルトでは、*volrpt* は不在または取り出し済みボリュームを報告しません。

不在、取り出し済み、および見つからないカートリッジ

ACSLs は 3 つのカートリッジ (ボリューム) のステータスを報告します。

- *missing*

LSM がオフラインになっているかドライブが通信中ではないため、カートリッジがライブラリ内で見つからず、カートリッジの記録された少なくとも 1 つの場所を検索できません。カートリッジに関する情報は保持されています。

- *absent*

カートリッジがライブラリ内で見つかりません。カートリッジの記録されたすべての場所は検索済みで、それらのどこにもカートリッジがありません。カートリッジに関する情報は保持されています。(保持期間が期限切れになる前に) カートリッジが見つかった、またはライブラリに再挿入された場合、再アクティブ化されます。

- *ejected*

カートリッジは取り出し済みです。カートリッジに関する情報は保持されていて、(保持期間が期限切れになる前に) 見つかった、または再挿入された場合、カートリッジは再アクティブ化されます。

カートリッジ (ボリューム) のステータスレポート

ACSLs は、ACSLs コマンドに回答してステータスが「missing」、「absent」、または「ejected」のカートリッジ (ボリューム) を報告し、これは ACSAPI リクエストに回答した場合とは異なります。

ACSLs コマンドに回答して表示される情報は、カートリッジを「missing」、「absent」、または「ejected」として識別します。

ただし、ACSLs が ACSAPI リクエストに回答して表示するカートリッジのステータス情報は、次の ACSLS 動的変数によって制御されます。

1. *missing*

- ACSLS 動的変数 `ENABLE_STATUS_VOLUME_MISSING` が TRUE の場合、ACSLs は `STATUS_VOLUME_MISSING` を報告します。
- ACSLS 動的変数 `ENABLE_STATUS_VOLUME_MISSING` が FALSE の場合、ACSLs は `STATUS_VOLUME_IN_TRANSIT` を報告します。

2. *absent*

- ACSLS 動的変数 `ENABLE_STATUS_VOLUME_ABSENT` が TRUE の場合、ACSLS は `STATUS_VOLUME_ABSENT` を報告します
- ACSLS 動的変数 `ENABLE_STATUS_VOLUME_ABSENT` が FALSE の場合、ACSLS はボリュームを ACSLS データベースから削除されているかのように扱い、`STATUS_VOLUME_NOT_IN_LIBRARY` を報告します。

3. *ejected*

- ACSLS 動的変数 `ENABLE_STATUS_VOLUME_EJECTED` が TRUE の場合、ACSLS は `STATUS_VOLUME_EJECTED` を報告します
- ACSLS 動的変数 `ENABLE_STATUS_VOLUME_EJECTED` が FALSE の場合、ACSLS はボリュームを ACSLS データベースから削除されているかのように扱い、`STATUS_VOLUME_NOT_IN_LIBRARY` を報告します。

”`ABSENT_VOLUME_RETENTION_PERIOD` 動的変数”

`ABSENT_VOLUME_RETENTION_PERIOD` 動的変数は、不在ボリュームと取り出し済みボリュームが ACSLS データベースに保持される時間を制御し、これらのボリュームが保持される日数を指定します。2 つの特殊な値があります。

- 値 0 (ゼロ) 日は、ボリュームが削除され、不在または取り出し済みとマークされないことを指定します。(これは ACSLS 6.1 より前の ACSLS リリースの動作です。)
- 値 999 日は、不在ボリュームと取り出し済みボリュームがデータベースにいつまでも保持されることを指定します。

カートリッジの回復

カートリッジ回復 (`acscr`) は ACSLS 内部プロセスで、ストレージセルまたはテープドライブの実際の内容が ACSLS データベースに保存されている情報と一致しない場合は常に、不一致を解決するために呼び出されます。これは次によって行われます。

- ライブラリで、ボリュームのホームセルと、場合によってはドライブを調べられるようにします。そのあと、その結果で ACSLS データベースを更新します。
- ACSLS (SL3000 および SL8500 ライブラリを含む) で、カートリッジが置かれているライブラリを確認してから、ライブラリの応答を使用して ACSLS データベースを更新することで、カートリッジを回復できるようにします。

カートリッジ回復で、別の場所に記録されているカートリッジなどの不一致が見つかった場合は、別の回復リクエストが作成され、それがリクエストキューに追加されます。(これは「カスケード」と呼ばれます。)

その他のプロセスは、ACSLS データベースとライブラリの実際の内容の間に不一致が発生したときに、カートリッジ回復に回復リクエストを渡します。そのため、カートリッジ回復は、カートリッジに見つからないとマークされたり、不在に変更されたり、再アクティブ化されたりする、中心となる場所です。したがって、ほかの多くの ACSLS コマンドおよびユーティリティーの動作のように見える動作は、実際には、ライブラリによって報告された情報に一致するようにデータベースを更新するときにカートリッジ回復によって行われます。

ほかのプロセスがカートリッジ回復に回復リクエストを渡すときは、次のことができます。

1. 続行し、非同期でカートリッジ回復を続行させます (カートリッジ回復は独立して進行します)。
2. 失われた特定のカートリッジが必要な場合は、カートリッジ回復がこの回復リクエストの処理を終了するのを待ってから、見つかったものを報告してください。

見つからないカートリッジ

次の場合、カートリッジは見つからないとしてマークされます。

- カートリッジ回復でライブラリ内のカートリッジが見つかりません。
- カートリッジのすべての記録された位置 (カートリッジに 1 つのドライブ位置が記録されている場合は、ホームセルとドライブ) を調べることができません。

たとえば、カートリッジ回復がオフラインの LSM またはオフラインのドライブのホームセルを調べることができない場合と、ほかの場所にあるカートリッジを探さない場合は、カートリッジに見つからないとしてマークされます。

カートリッジ回復は、カートリッジのホームセルを調べて、そこで別のカートリッジが見つからないかぎり、カートリッジのホーム位置を維持します。この場合、カートリッジには「homeless」とマークされ、`home_lsm` フィールドにマイナス 1 (-1) が入ります。

カートリッジ回復が見つからなかったカートリッジを見つけると、見つからなかったカートリッジが見つかった場所に依じて、そのカートリッジのステータスがデータベース内で「home」または「in drive」に変更されます。

- カートリッジが記録されたホームセル以外のセル内で見つかった場合、カートリッジ回復はカートリッジのホームセルをチェックして、重複したカートリッジが見つかったかどうかを確認します。
- カートリッジが記録されたホームセル内にはない場合、カートリッジ回復は、それが見つかったセルをその新しいホームセルとして記録します。

- 新しいカートリッジが重複している場合、カートリッジ回復はイベントログにこれを報告します。重複するカートリッジは取り出されません。
- カートリッジ回復は、ドライブで「homeless」カートリッジを見つけた場合は、新しいホームセルを割り当てません。カートリッジがマウント解除されるときに、マウント解除プロセスで新しいホームセルが割り当てられます。

不在カートリッジと取り出し済みカートリッジ

このセクションでは、不在カートリッジと取り出し済みカートリッジについて説明します。

カートリッジが見つからない

カートリッジ回復が記録されたすべての場所を調べることができ、カートリッジを見つけることができない場合:

1. *ABSENT_CARTRIDGE_RETENTION_PERIOD* が 0 の場合、カートリッジ回復は、
 - カートリッジのレコードをデータベースから削除します。
 - カートリッジのホームセルだったセルのデータベース内のセルレコードに、「empty」とマークします。
2. *ABSENT_CARTRIDGE_RETENTION_PERIOD* が 0 より大きい場合、カートリッジ回復は、
 - まだカートリッジに不在か取り出し済みとマークされていない場合は、データベース内のカートリッジのレコードのステータスを「absent」に変更します。
 - カートリッジを「homeless」として記録します (*home_1sm* フィールドにマイナス 1 (-1) が入ります)。
 - カートリッジの以前のホームセルのデータベース内のセルレコードに「empty」とマークします。

カートリッジが見つかった

カートリッジ回復は、取り出し済みカートリッジまたは不在カートリッジを見つけた場合、カートリッジを再アクティブ化します。

取り出し済みカートリッジまたは不在カートリッジがストレージセル内で見つかった場合は、これがその新しいホームセルになり、カートリッジ回復がデータベース内のカートリッジのステータスを「home」に変更します。

ドライブでカートリッジが見つかった場合は、ACSL5 がカートリッジのマウント解除時に新しいホームセルを割り当てます。

手動ボリューム削除ユーティリティーの使用

手動ボリューム削除ユーティリティー、`del_vol` により、オフラインになっていて使用できない LSM 内にあるボリュームにアクセスできます。カートリッジを手動で LSM から取り出して、別の LSM に再挿入しようとする、ACSLS は `duplicate volume` メッセージを発行し、カートリッジを挿入しません。`del_vol` ユーティリティーを使用すると、最初にデータベースからボリュームを削除してから、オフラインの LSM から手動で取り外して、オンラインの LSM 内に正常に再挿入できます。

`del_vol` ユーティリティーは、ボリュームを削除するためのオプションを使用して、ボリュームを不在として保持するようになりました。ボリュームは、不在または取り出し済みというステータスの期限切れを待たずに削除できます。

注:

オンライン LSM からカートリッジを取り外すには、カートリッジに対して `eject` コマンドを発行します。カートリッジが実際には LSM 内にはない場合は、`-f` (force オプション) を指定して `del_vol` を実行できます。このユーティリティーを使用するには、ACSLS とデータベースが稼働している必要があります。システムの回復中は `del_vol` を実行しないでください。予期しない結果が発生する可能性があります。

このユーティリティーの詳細については、「[del_vol](#)」を参照してください。

`del_vol` ユーティリティーを使用してカートリッジを削除するには:

1. `acsss` としてログインします。
2. カートリッジを削除します。

```
del_vol vol_id
```

期限切れカートリッジの特定

テープカートリッジがその設計寿命を超えると、メディアが擦り切れて、カートリッジのゲートのような機械部品が摩耗することがあります。カートリッジが設計された寿命の終わりに達したときは、それらのデータを新しいカートリッジに移行して、古いカートリッジを廃棄することを検討してください。これにより、カートリッジの機械の部品が故障したりデータが読み取り不可になったりするわずかなリスクを回避できます。

カートリッジの時系列経過時間と使用時間は異なります。いくつかの 9840 カートリッジが 10 年間使用され続けていても、それらの使用パターンは異なります。毎日使用されていたものもあれば、規模の大きいアーカイブに使用され、まれにしかアクセスされないものもあります。設計寿命を越えたカートリッジを特定することは、きわめて重要です。

廃棄する必要があるカートリッジを特定するには、それらの使用状況を確認する必要があります。カートリッジの使用状況はカートリッジのディレクトリに記録され、カートリッジがマウント解除される前に、ドライブがディレクトリを更新します。

ACSLS によって制御されるライブラリ内のカートリッジの場合:

- ACSLS が管理する一部のライブラリの場合、カートリッジの使用状況は「保証期限」と「寿命」のパーセンテージとして表示されます。
- ACSLS の以前のリリースとライブラリの場合は、*display* コマンドと *volrpt* ユーティリティを使用して ACSLS *access_count* を表示できます。

カートリッジの寿命のパーセンテージ

最新のファームウェアを実行中の最新のライブラリと、最新のファームウェアを実行中の StorageTek ドライブでは、カートリッジがマウント解除される時に、テープドライブからライブラリにカートリッジの「保証期限」と「寿命のパーセンテージ」が報告されます。そのあと、ライブラリがこれを ACSLS に報告します。ACSLS はこの情報をそのデータベースに保存するため、ACSLS の *display volume* コマンドを実行することでそれを表示できます。[「display コマンドオプションの使用」](#)を参照してください。

例: すべての T9840 カートリッジを *end_of_life* でソートして、ACS、LSM、メディア、および *end_of_life* 情報とともに表示するには:

```
display volume * -media STK1R -f acs lsm media end_of_life warranty
_life -s end_of_life
```

具体的には、これらのライブラリとドライブについて、この情報が ACSLS に報告されます。

ライブラリ:

- SL3000
- SL8500 (4.10 ファームウェア)

テープドライブ:

- すべての T10000 テープドライブ - 1.38 ファームウェア
- T9840A、T9840C、および T9840D (T9840B を除くすべての T9840 テープドライブ。) - 1.42 ファームウェア
- T9940A および T9940B テープドライブ - 1.42 ファームウェア

アクセス数

多くの場合、カートリッジ寿命のレポートは使用可能ではありません。このような場合は、ACSLS `access_count` が使用可能な最良の情報です。ACSLS データベースは、ボリュームが選択またはアクセスされた回数を記録します。これは、カートリッジが接続されたライブラリのグループ (ACS 内) に置かれたままだった場合に、それらがマウントされた回数を推定するために使用できます。

この情報はライブラリのタイプに関係なく収集されるため、9310、4410、および 9360、さらに SL8500 と SL3000 についても維持されます。ACSLS はこの情報を 10 年間保存しているため、まだ下位レベルのリリースを使用している場合でも、この情報は引き続き存在しています。ただし、このデータには制限があります。もっとも大きいのは、カートリッジがライブラリに挿入されたときにカウントがゼロ (0) に設定されることです。

設定されている保持期間のボリュームについての情報が保持されるため、カートリッジが ACS から取り出され、保持期間内に同じまたは別の ACS に再挿入されたときに、カウントが保存されます。デフォルトの保持期間は 5 日です。ただし、ボリュームがライブラリから取り出され、ボリューム情報の保持期間よりも長くオフサイトに置かれたままの場合、ボリュームに関する情報は ACSLS データベースから削除されます。

単一のライブラリ内に残されていたカートリッジの場合、これらの ACSLS アクセス数は非常に役立ちます。T9840 カートリッジの場合は、ACSLS `access_count` が 11,000 を超えている場合、問題のカートリッジがまだ寿命を過ぎていない場合は寿命に近づいています。T10000 カートリッジの寿命の値は 16,000 マウントです。

ACSLS は新しいリリースの ACSLS がインストールされたときにデータベース情報を保持および移行できるツールを提供しているため、この情報は 10 年以上遡って存在できます。カートリッジからのデータが存在しない場合は、これが唯一のオプションです。

ACSLS のカートリッジマウント回数の詳細

この ACSLS フィールドは `access_count` と呼ばれます。これは次をカウントします。

- マウント (マウント解除はカウントされません)
- 挿入と取り出し (多くの場合、挿入と取り出しはまれにしかありません)
- 移動 (ただし、`cmd_proc` を使用した `move` コマンドはまれにしか使用されず、ACSAPI クライアントでは使用できません)

`access_count` は、主にカートリッジがマウントされた回数のカウントです。ACSLS は取り出し済みボリュームを `ABSENT_VOLUME_RETENTION_PERIOD` (デフォルトは 5 日) の間記

憶します。カートリッジが ACS 間で移動し、オフサイトに送られてからオンサイトに戻るときに、ACSLs が `access_count` を記憶できます。

次の両方を使用して、ACSLs `access_count` を参照できます。

- ACSLS `display` コマンド。

アクセス数でソートされたすべての 9840 データカートリッジを表示し、メディアタイプ、ACS、および LSM も表示するには:

```
display volume * -media STK1R -s access_count -f media access_count
acs lsm
```

- `volrpt` ユーティリティ。

`volrpt` は使用回数 (`access_count`) でソートでき、選択されたフィールドのみを含めることができます。たとえば、`vol_id`、`media type`、`access_count`、および `location` が含まれているカスタムの `volrpt` は、スクリプトによってさらに処理できるようにプラットフォームに出力できます。

カートリッジの保証期限と寿命のしきい値

保証期限と寿命のしきい値を次の表に示します。

表9.6 寿命のしきい値

しきい値	マウント回数
9x40 (T9840 および T9940) の保証	10,000
9x40 の寿命	11,000
T10000 の保証	15,000
T10000 の寿命	16,000

アクティブな LSM からの、もっとも長い間アクセスされていないカートリッジの移動

ACSLs はテープドライブからカートリッジをマウント解除するときに、ほかの LSM からテープドライブと同じ LSM 内の新しいホームセルに移動したカートリッジを「フロート」することで、パススルーを回避しようとします。

たとえば、カートリッジが SL8500 ライブラリ 3、レール 2 (LSM 9) からライブラリ 1、レール 4 (LSM 3) のドライブにマウントされる場合、これには 2 回の水平方向のパススルーと 1 回の

エレベータパススルーが必要です。ACSLs は、カートリッジをマウント解除するときに、LSM 4 内の新しいホームセルを見つけてマウント解除時のパススルーを回避しようとします。

問題:

LSM に空き (未割り当て) ストレージセルがない場合は、LSM にカートリッジをフロートできません。テープドライブの LSM に空きセルがない場合、ACSLs は引き続きカートリッジをドライブにもっとも近い LSM にマウント解除しようとしますが、これには少なくとも 1 回のパススルーが必要です。

解決方法:

長時間にわたってアクセスされていないカートリッジを特定し、それらをいっぱいになった LSM から出して、カートリッジがマウント解除時にフロートできる空きセルを用意します。

カートリッジがマウントまたはマウント解除、挿入、または移動されるたびに、ACSLs がカートリッジに関して記録する情報の中の *access_date* が更新されます。*access_date* はそれらのアクティブではないカートリッジを特定するために使用できます。

このプロセス全体は、同じ ACS 内にボリュームを移動するだけで、カートリッジの取り出しやそれらのステータスの変更など、これらのカートリッジの将来のマウントを妨げる可能性のあることを何も実行しない場合は安全です。

もっとも長い間使用されていないカートリッジを特定して移動するには、次の手順に従います。

- 空きセルが少なすぎる LSM と空のセルを含む LSM を特定します。
- もっとも早いアクセス日で LSM 内のカートリッジを選択します。
- いっぱいになった LSM から空のセルを含む LSM にカートリッジを移動します。

空きセルが少なすぎる LSM と空のセルを含む LSM を特定する

cmd_proc の使用:

```
query lsm all
```

「Free Cell Count」列を使用すると、空きセルがほとんどないかまったくない LSM と、非アクティブなカートリッジを移動できる空のセルがある LSM の両方を特定できます。

例:

```
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

アクセス日で LSM 内のカートリッジを確認する

ここで、ACS 内のほかの LSM に移動できる非アクティブなカートリッジを特定する必要があります。

簡単にソートできるアクセス日が報告されることを確認する

アクセス日がソートの助けになる方法で報告されていることを確認します。日付が報告される形式は、`TIME_FORMAT` 動的変数によって制御されます。

- デフォルト形式 `TIME_FORMAT=%Y-%m-%d %H:%M:%S` を使用して、カートリッジをアクセス日で簡単にソートできるようにします。UNIX コマンドプロンプトで、次を入力します。

```
dv_config -p TIME_FORMAT
```

変数プロンプトで Enter ? と入力して、ヘルプを表示します。

- 変更を行なった場合は、共有メモリー内の動的変数を更新します。

```
dv_config -u
```

LSM 内のカートリッジの最終アクセス日の分布を調べる

十分な空きセルがない LSM ごとに、最終アクセス日でソートされたカートリッジをリストします。`VOLID` と `access_date` のみを選択するカスタムの `volrpt` が必要です。

詳細については、次でコメントのヘッダーを参照してください。

```
$ACS_HOME/data/external/volrpt/owner_id.volrpt
```

一列になったフィールドは、`field_name`、`field_length`、および `delimiter_length` (このフィールドのあとは空白) です。

次の例では、2 行のアクティブな行があります。おそらく 6 文字の `VOLUME_ID` が表示されます。`ACCESS_DATE` については日付部分のみが必要で、時間は必要ありません。

```
VOLUME_ID      6      2
ACCESS_DATE    10     2
```

レポートを作成するには:

1. 次のように入力します: `$cd ACS_HOME/data/external/volrpt.`
2. `owner_id.volrpt` をコピーして、`access_date.volrpt` などのファイルに保存します

[「ロギングボリューム統計レポートの作成」](#)を参照してください。

3. テキストエディタを使用して、`ACCESS_DATE` を編集します。
4. LSM のカートリッジのソートされたリストを作成します。

```
volrpt -l <lsm_id> -d -f access_date.volrpt | sort -k 2,2 -0 vols  
_sorted_lsm_##
```

ここで、`access_date.volrpt` はカスタムレポートの名前で、`##` は LSM 番号です。

`vols_sorted_lsm_##` ファイルを調べて、各 LSM の最終アクセス日の分布を確認します。

いっぱいになった LSM から空のセルを含む LSM にカートリッジを移動する

現在は、移動するカートリッジのリストを作成して、空き容量のある LSM に移動する必要があります。

移動するカートリッジのリストを作成する

1. アクセス日でソートされたカートリッジのリストが含まれるファイルを取得し、アクセス日を削除してカートリッジのリストだけが含まれるようにします。

```
cat vols_sorted_LSM_## | cut -d" " -f1 > vols_LSM_##_tmp
```

2. 各 `vols_LSM_##` ファイルを取得し、移動するカートリッジの最初の 100 (または指定する数) を選択します。

```
head -100 vols_LSM_##_tmp > vols_LSM_##
```

前述の操作の両方を組み合わせることができます。

```
cat vols_sorted_LSM_## | cut -d" " -f1 | head -100 > vols_LSM_##
```

空き領域のある LSM にカートリッジを移動する

カートリッジの移動元の LSM ごとに、カートリッジ用の空き容量のある移動先 LSM を選択します。

1. `moving.sh` ユーティリティーを使用して、
`-t <lsm_id>` (例: `-t 0,8`) で指定した新しい LSM にカートリッジを移動します。

```
moving.sh -f vols_LSM_## -t <lsm_id>
```

2. LSM ごとに別々の `moving.sh` を実行します。

ライブラリがビジー状態の場合は、一度に 1 つまたは 2 つの `moving.sh` ユーティリティーしか実行できない場合があります。

無効になっている LSM のドライブへのカートリッジの手動ロード

LSM が機能せず、オフラインにした場合、データパスが引き続き動作している場合は、ライブラリドライブにカートリッジを手動でロードできます。

無効になっている LSM 内のドライブにカートリッジを手動でロードするには:

1. LSM ドアを開きます。
2. すでにドライブ内にあるカートリッジのカートリッジラベルをメモして、それらを取り外します。この手順の最後にこれらのカートリッジを交換する必要があります。
3. 読み取りまたは書き込みするカートリッジのあるドライブをロードします。

LSM が修復されるまで必要なだけこの手順を繰り返してから、手順 4 に進みます。

注意:

この手順では、カートリッジをライブラリセルから取り外して、これらのカートリッジをドライブにロードできます。これらのカートリッジのセル位置をメモし、手順 4 でカートリッジを確実にこれらの位置に戻してください。

4. LSM が修復されたあと、すべてのカートリッジをドライブから取り外して、手順 2 でメモした元のカートリッジと交換します。
5. LSM ドアを閉じて、LSM をオンラインに `vary` し、通常の操作を再開します。

第10章 データベース管理

データベースには、ライブラリの構成とすべてのライブラリカートリッジの場所に関するすべての情報が格納されます。

バックアップおよび回復される ACSLS 制御ファイルには、\$ *ACS_home* の下の *data/external* にあるユーザーが構成可能なファイルと *data/internal/client_config* にあるいくつかのファイルが含まれます。

この章では、データベースのインポートとエクスポート、インポートされたデータベースとライブラリ構成の検証、データベースのバックアップ、およびデータベースの復元と回復について説明します。

- データベースのエクスポートとインポートには次が含まれます。
 - データベースのディスクファイルまたはローカルテープデバイスへのエクスポート
 - ディスクファイルまたはローカルテープデバイスからのデータベースのインポート
 - ACSLS 制御構成ファイルのインポート
 - カスタマイズされた動的変数のマージ
 - インポートされたデータベースとライブラリ構成の検証
- データベースと ACSLS 制御ファイルのバックアップには次が含まれます。
 - 自動データベースバックアップ
 - ローカルテープデバイスまたはディスクへの手動のバックアップの実行
 - UNIX ファイルへのバックアップ
 - 別のサーバーに復元できるバックアップの作成
- データベースと ACSLS 制御ファイルの回復と復元には次が含まれます。
 - データベースの最新のバックアップへの復元
 - 特定のファイルからの回復
 - ACSLS 制御ファイルの復元
 - 別のサーバーで作成されたバックアップの復元
 - データベースの再起動

使用されるユーティリティ

次のユーティリティを使用します。

- 次にバックアップするための *bdb.acsss* ユーティリティ:
 - 指定された UNIX ファイル
 - テープデバイス
 - デフォルトのファイルと場所
- 次のための *rdb.acsss* ユーティリティ:
 - データベースの破損からの回復
 - 意図しない結果を生成する変更から
 - サーバー障害から
- ACSLS のバージョン間の移行のための *db_export.sh* および *db_import.sh* ユーティリティ。これには、以降のリリースへの移行または以前のリリースへの移行が含まれません。

ACSLs をインストールするときに、自動的にデータベース管理ソフトウェアもインストールします。ACSLs のインストール後、次の実行時に、ACSLs データベースが初期化されます。

- *acsss_config* を使用して、ライブラリハードウェアを構成します
- *db_import.sh* を使用して、以前にエクスポートされたデータベースをインポートします
- *rdb.acsss.sh* を使用して、別のサーバーに作成されたデータベースバックアップを回復します

データベースのエクスポート

このセクションでは、ACSLs データベースとその関連付けられた ACSLS 制御ファイルを、ACSLs の以前のバージョン、ACSLs の同じリリースレベルから移行するか、または以前のリリースに戻す方法について説明します。

注:

ACSLs の実行中は *db_export.sh* を実行できません。これにより、データベースの整合性のあるコピーが確実に取得されます。

db_export.sh ユーティリティは、テープまたは指定されたファイル上のデータベースの ASCII 表現をディスクに作成します。これは ACSLS 制御ファイルの収集も担当します。このユーティリティは 2 つの方法で使用できます。

- これをオプションなしで実行した場合、エクスポートされたファイルはデフォルトのテープデバイス `/dev/rmt/rmt/0n` にコピーされます。

```
db_export.sh
```

- 別のテープデバイスを使用する場合は、`-f` オプションのあとに目的のテープデバイスを続けて使用します。

```
db_export.sh -f /dev/rmt/3n
```

- 同じマシン上のローカルファイルにエクスポートする場合は、`-f` オプションを使用して、ファイルのパス名を指定します。

```
db_export.sh -f /export/save/acsls_export.03_Dec_2014
```

ファイルに保存する場合、結果は2つの個別のファイルになります。データベーステーブルはユーザーが指定したファイル名で保存されます。その他の制御ファイルは `.misc` 拡張子を持つ同一のパス名になります。

`db_export.sh` によって生成されるファイルは、アップグレードまたは回復時に `db_import.sh` ユーティリティーへの入力として使用されます。

注:

これはすべての以前の ACSLS バージョンを最新バージョンに移行するための望ましい方法です。

`-f` オプションを付けても付けなくても、`db_export.sh` ユーティリティーを実行すると、エクスポートする ACSLS のバージョンを選択するように求められます。

```
$ db_export.sh
```

データベースの `/dev/tape` へのエクスポート

次のオプションから選択することによってエクスポートするリリースを選択します。

テープにエクスポートする場合は、巻き戻しなし (`no-rewind`) デバイスが必要です。

- 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

以前のリリースへのエクスポートの前のサポートされていない テープライブラリ、ドライブ、およびカートリッジの取り外し

テープライブラリ、テープドライブ、またはカートリッジメディアタイプの一部をサポートしていない ACSLS の以前のリリースにデータベースをエクスポートする場合は、データベースをエクスポートする前に、サポートされていないテープライブラリを構成から削除し、テープドライブとカートリッジをライブラリから取り外します。そうしないと、次が発生する可能性があります。

- ライブラリをサポートしていない以前の ACSLS リリースを選択する場合、データベースをエクスポートする前に構成からライブラリを削除するように求められます。
- テープドライブを、それらをサポートしていない以前の ACSLS リリースにエクスポートした場合、ドライブは「unknown」として報告され、使用できません。
- メディアタイプをサポートしていない以前の ACSLS リリースにカートリッジをエクスポートすると、カートリッジが不在としてマークされるため、それらを手動でライブラリから取り外す必要があります。

Linux にエクスポートする前の論理ライブラリの削除

Linux 上で実行される ACSLS は、ファイバターゲットを使用してアクセスされる論理ライブラリをサポートしていません。Linux 上で実行される ACSLS にデータベースをエクスポートする場合は、論理ライブラリをすべて削除します。そうしないと、Linux で実行されるすべての論理ライブラリを使用できなくなります。

ディスクファイルへのエクスポート

次の手順に示すように、ACSL S データベースと ACSLS 制御ファイルをディスクファイルにエクスポートできます。

1. `acs`ss としてログインします。
2. ACSLS を無効にします。

`acs`ss *disable* (UNIX コマンドプロンプトから)

3. `db_export.sh` ユーティリティを起動します。

```
db_export.sh -f /path/db_file
```

4. 移行先のバージョンの目的のオプションを選択します。
 - これを実行すると、ユーティリティに、テーブルデータが正常にエクスポートされたことを示す出力が表示されます。
 - エクスポートが完了すると、エクスポートが成功したことを示すメッセージが表示されます。
 - `db_export.sh` ユーティリティは、`-f` オプションで指定された場所に 2 つのファイル `db_file` と `db_file.misc` を作成します。
5. これらのファイルは削除されないセキュアな場所に配置するか、移動してください。

次のディレクトリは ACSLS 保守がインストールされるときに削除される可能性があるため、これらのファイルをこのディレクトリ内または下に配置しないでください。

- `$ACS_HOME`
(ACSSS ホームディレクトリ)
- `$ACSDB_BACKUP_DIR` (`/export/backup` など)
(ACSLs バックアップが格納されているディレクトリ)
- `/tmp`

注:

新しいリリースのオペレーティングシステムをインストールする予定がある場合は、ACSLs サーバーにエクスポートされたファイルを保存しないでください。

6. ACSLS とデータベースを起動するには、次のコマンドを入力します。

```
acsss enable
```

テープへのエクスポート

次の手順に示すように、ACSLs データベースと ACSLS 制御ファイルをテープにエクスポートできます。

データベースと ACSLS 制御ファイルをテープにエクスポートするには:

1. `acsss` としてログインします。
2. ACSLS を無効にします。

```
acsss disable (UNIX コマンドプロンプトから)
```

3. デフォルトのテープデバイスに空のテープを挿入します。

4. `db_export.sh` ユーティリティーを起動します。

```
db_export.sh -f tape_device
```

例: `dbexport.sh -f /dev/rmt/0mn`

5. エクスポートする目的のオプションを選択します。

これを実行すると、ユーティリティーに、テーブルデータが正常にエクスポートされ、ACSL5 ファイルが正常にバックアップされたことを示す出力が表示されます。エクスポートが完了すると、メッセージが表示されます。

6. プログラムが完了し、プロンプトが再表示されたときにのみ、ドライブからカートリッジを取り外してください。

注意:

プログラムがエクスポートを完了する前に、カートリッジを取り外すと、ファイルが失われます。「カートリッジを保護」と書き、エクスポートされたデータベースとして内容を識別するためにはっきりとマークを付けます。

ライブラリにカートリッジを残しておかないでください。

7. ACSLS とデータベースを起動するには、次のコマンドを入力します。

```
acsss enable
```

データベースのインポート

`db_import.sh` ユーティリティーを使用すると、次の属性が新しいデータベースにインポートされます。

- ボリューム: これらのデータベーステーブルには、次のようなライブラリ内の各ボリュームに関連付けられたすべての情報が含まれます。
 - ボリュームが存在する場所
 - カートリッジのタイプ (データ、スクラッチ、クリーニングなど)
 - 最後に関連付けられたスクラッチプール
 - カートリッジの現在のステータス (ホーム、マウント済みなど)
 - 挿入日と最後にアクセスされた日付
 - 挿入日からのマウントの数
 - 最大使用回数 (クリーニングカートリッジの場合)
 - 関連付けられたロック ID およびユーザー ID (カートリッジがロックされている場合)

- ACS とライブラリ: データベーステーブルには、LSM、ドライブ、パネル、セルなどの ACS およびライブラリコンポーネントが含まれます。
- ACSLS 制御ファイルには、次を含む初期インストール以降のすべての構成の更新が含まれます。
 - アクセス制御情報
 - 固定ボリュームの設定
 - スクラッチメディアの設定
 - カスタム `volrpt` テンプレート
- 動的および静的変数: 以前のリリースでカスタマイズされた動的変数をインポートできません

このセクションでは、`db_import.sh` ユーティリティを使用して次を実行する方法について説明します。

- ACSLS データベースを再作成します
- 重要な ACSLS 制御ファイルを回復します
- `db_export.sh` ユーティリティを使用して、エクスポートされたデータからカスタマイズされた動的変数を回復します。

ディスクファイルからのインポート

次の手順に示すように、ディスクファイルから ACSLS データベースと ACSLS 制御ファイルをインポートできます。

ディスクファイルから ACSLS データベース、ACSLS 制御ファイル、またはカスタマイズされた動的変数をインポートするには:

1. `acsss` としてログインします。
2. ACSLS を無効にします。

```
acsss disable (UNIX コマンドプロンプトから)
```

3. `db_import.sh` ユーティリティを起動します。

```
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:

- オプション 1 - 異なるリリースまたはプラットフォームバージョンからのデータ、制御ファイル、および動的変数のインポート。

このオプションは、別のリリースに移動するかまたは ACSLS のアップグレード時に、データベースファイル、制御ファイル、および動的変数をインポートするために使用します。

注意:

既存のデータベースと制御テーブルと動的変数の設定は破棄され、再構築されて、エクスポートから提供されたデータが取り込まれます。結果は最終で、データベースを再構築しなければ回復されません。既存のテーブル内の情報を保持するには、*db_export.sh* を使用して、テーブルデータをエクスポートしていないかぎり続行しないでください。

このオプションは以前の環境からカスタマイズされた動的変数も回復します。これは、以前にカスタマイズされた動的変数を記録する必要がなく、ACSLs のバージョンをアップグレードする場合に役立ちます。ディレクトリ *acs.home* 内の *data/external* 下のアクセス制御ファイルを含むすべてのファイルが回復されます。アクセス制御が構成されている場合、*data/internal/client_config* も回復されます。

- オプション 2 - 同じリリースまたはプラットフォームバージョンからのデータ、制御ファイル、および動的変数のインポート

このオプションは、データベースと制御ファイルの両方を含む ACSLS 環境を再作成するために使用します。これは次の場合に使用されます。

- ハードウェアの障害からの回復、またはハードウェアのアップグレード時。
- データのエクスポート元の ACSLS サーバーと同じにするため、ACSLs サーバーを再構築する必要があります。
- オプション 3 - 任意の ACSLS リリースレベルからのデータベーステーブルのみのインポート

このオプションは、任意の ACSLS リリースレベルからデータベースファイルのみをインポートする場合に使用します。

このオプションは、既存のデータベーステーブルおよび制御ファイルを破棄し、それらを再構築し、エクスポートされたデータベースから提供されたデータをそれらに取り込みます。既存のテーブル内の情報を保持するには、`db_export.sh` を使用して、データをエクスポートしていないかぎり続行しないでください。

- オプション 4 - 任意の ACSLS リリースレベルからの ACSLS 制御ファイルのインポート

このオプションは、任意のバージョンの ACSLS から ACSLS 制御ファイルのみをインポートする場合に使用します。これは、ディレクトリ `acs.home` 内の `data/external` 下のアクセス制御ファイルを含むすべてのファイルをインポートします。アクセス制御が構成されている場合、`data/internal/client_config` もインポートします。

このオプションは同じバージョンから ACSLS データベースファイル、制御ファイル、動的変数を回復します。これは、ディレクトリ `acs.home` 内の `data/external` 下の `access` 制御ファイルを含むすべてのファイルを回復します。

このオプションは以前の環境からカスタマイズされた動的変数を回復します。これは、以前にカスタマイズされた動的変数を記録する必要がなく、ACSLs のバージョンをアップグレードする場合にきわめて役立つオプションです。

このオプションを選択すると、データベースのエクスポートから設定が収集され、新しい変数の設定で共有メモリーが再構成されます。

- オプション 5 - カスタマイズされた動的変数のみのマージ

これは、以前にカスタマイズされた動的変数を記録する必要がなく、ACSLs のバージョンをアップグレードする場合にきわめて役立つオプションです。このオプションを選

択すると、データベースのエクスポートから設定が収集され、新しい変数の設定で共有メモリーが再構成されます。

警告:

このオプションを実行する前に **ACSL S 7.2.0** からインポートし、**ACSL S** を起動すると、特定のデータが失われる可能性があります。**ACSL S** を以前のバージョンからアップグレードし、カスタマイズされた動的変数があった場合、**ACSL S** を起動する前に、カスタマイズされた変数をインポートする必要があります。

4. 「[インポートされたデータベースとライブラリ構成の検証](#)」の説明に従って、インストールを検証します。
5. **ACSL S** を起動するには、次のコマンドを入力します。

```
acsss enable
```

テープからのインポート

次の手順を使用して、テープから **ACSL S** データベースをインポートし、**ACSL S** 制御ファイルを回復して、カスタマイズされた動的変数を再構築します。

1. **acsss** としてログインします。
2. **ACSL S** を無効にします。

```
acsss disable (UNIX コマンドプロンプトから)
```

3. **db_export.sh** コマンドでエクスポートしたデータベースのテープをテープドライブに挿入します。
4. UNIX コマンドプロンプトで、次を入力して、データベースインポートユーティリティーを実行します。

```
db_import.sh
```

db_import.sh ユーティリティーは、「[ディスクファイルからのインポート](#)」に示すように、そのメインメニューを表示します。詳細情報も提供します。

注:

別の端末からテープの巻き戻しを行なっているときに、端末から **db_import** ユーティリティーを実行すると、「**unsuccessful**」メッセージを受け取ります。

5. メニューオプションについては、[手順3](#)を参照してください。
6. 「[インポートされたデータベースとライブラリ構成の検証](#)」の説明に従って、インストールを検証します。

7. デフォルトのテープデバイス以外のテープ (巻き戻しなし (no rewind)) からインポートします。
8. ACSLS を起動するには、次のコマンドを入力します。

```
acsss enable
```

ファイバ用の mchanger の新しいプラットフォームへの移行

SCSI メディアチェンジャー (mchanger) は ACSLS とファイバ接続ライブラリ間で通信するファイバ接続ライブラリデバイスドライバです。mchanger は ACSLS に接続されているファイバ接続ライブラリごとに作成する必要があります。

`/dev/mchanger#` デバイスドライバリンクの番号は、別のプラットフォームやリリースに ACSLS をインポートすると変更されることがあり、これによって問題が発生する可能性があります。たとえば、古い ACSLS サーバーで `/dev/mchanger3` 経由で接続されていた SL500 または SL150 ライブラリが、新しい ACSLS サーバーでは `/dev/mchanger4` 経由で接続されることがあります。

Linux では mchanger 名の形式が異なるため、ACSL S Linux サーバーから新しい Linux サーバーに移動するときはこれは問題になりません。Linux サーバーでは、番号の代わりに mchanger 名にライブラリシリアル番号が含まれます。

次の手順は、ファイバ接続ライブラリ用にメディアチェンジャードライバを構成していて、新しい ACSLS リリースまたはサーバープラットフォームに移行する場合の問題を回避します。

1. Solaris または AIX ACSLS サーバーで、古い ACSLS サーバー上の各ファイバ接続ライブラリに関連付けられた mchanger 番号を記録します。「[古い ACSLS サーバーのファイバ接続ライブラリに関する詳細の記録](#)」を参照してください。
2. ライブラリの新しい mchanger 名で構成を更新します。「[ファイバ接続ライブラリの mchanger 名を変更するための ACSLS の再構成](#)」を参照してください。

古い ACSLS サーバーのファイバ接続ライブラリに関する詳細の記録

Solaris または AIX ACSLS サーバーで、古い ACSLS サーバーからデータベースをエクスポートする前に、古い ACSLS サーバー上の各ファイバ接続ライブラリに関連付けられた mchanger 番号を記録します。各ファイバ接続 ACS に関連付けられた mchanger およびライブラリのシリアル番号を示す `cmd_proc` および `showDevs.sh` ユーティリティからの出力を保存します。

```
cmd_proc:
```

- `query lmu all`

これは ACSLS とそれらのポート接続によって制御されるすべての ACS を表示します。Solaris および AIX システム上のファイバ接続ライブラリのポート名は `/dev/mchanger#` になります。ここで、`#` は番号です。

- `display lsm * -f type serial_num`

これは ACSLS によって管理されるすべての LSM のライブラリタイプとシリアル番号を表示します。SL500 または SL150 などのライブラリタイプを使用して、ファイバ接続ライブラリを識別します。シリアル番号を使用して、特定のライブラリを識別します。

ユーティリティ:

```
showDevs.sh -s
```

`-s` オプションを付けた `showDevs.sh` ユーティリティは、ファイバ接続ライブラリを識別する mchanger デバイスリンク、ライブラリのタイプ、ライブラリのシリアル番号、および詳細を表示します。

ファイバ接続ライブラリの mchanger 名を変更するための ACSLS の再構成

データベースをインポートしたあとに、Linux へ移行するか Linux から移行する場合、または同じ mchanger 番号が Solaris に構成されていない場合、これらのライブラリの新しい mchanger 名で構成を更新する必要があります。

`acsss_config` の使用:

1. `acsss` としてログインします。
2. `showDevs.sh` を使用して、すべてのファイバ接続ライブラリを表示します。

`-s` オプションを付けた `showDevs.sh` ユーティリティは、ファイバ接続ライブラリを識別する mchanger デバイスリンク、ライブラリのタイプ、ライブラリのシリアル番号、および詳細を表示します。

3. `acsss_config` からのプロンプトにコピー&ペーストできるように、`showDevs.sh` からの出力をファイルに保存します。
4. 1 つの端末ウィンドウに `showDevs.sh -s` からの出力を表示した状態で、2 つ目の端末ウィンドウを開き、`acsss` としてログインします。

5. 2つ目の端末ウィンドウで、*acsss_config* を実行します。
6. 「Option 8: Define or Change Library Configuration」を選択します。
7. 「Configure library communications? (y/n)」に **y** と応答します。
8. 「Library server data base exists and will be overwritten, continue (y or n)?」に **y** と応答します。
9. *query lmu all* からの保存された出力を参照して、古い ACSLS サーバー上に構成されたすべての ACS を再構成します。
 - a. 古い ACSLS サーバーにあったとおりに、すべての ACS を同じ順番、同じ ACS 番号で構成します。
 - b. ファイバ接続以外のライブラリを、パーティション分割するかまたはパーティション分割しないで、および古い ACSLS サーバー上と同じポート接続で構成します。
10. ファイバ接続ライブラリのポート接続を構成するときに、新しい ACSLS サーバーで使用される新しい *mchanger* リンク名を指定します。Linux 上の *mchanger* リンク名の場合、これを行う簡単な方法は、*showDevs.sh* 出力からそれらをコピーし、*acsss_config* のプロンプト後にそれらを貼り付ける方法です。
11. ACSLS ライブラリハードウェアの再構成を終了します。

インポートされたデータベースとライブラリ構成の検証

ACSLs を検証するには、次の手順を使用して、カートリッジをマウントまたはマウント解除します。

1. *acsss* としてログインしていることを確認します。
2. ACSLS が実行されていない場合は、次のコマンドを入力して起動します。

```
acsss enable
```

3. 次のコマンドを入力して、*cmd_proc* からサーバーをクエリーします。

```
query server
```

サーバーが回復モードであることを示すメッセージが表示された場合は、サーバーが実行中になったことを示すメッセージを待ちます。

4. 次がオンラインであることを確認します。そうでない場合は、*vary* コマンドでそれらをオンラインにします。

```
query port all
```

```
query acs all
```

```
query lsm all
```

```
query drive all
```

5. LSMに少なくとも1つのカートリッジがありますか。
 - はい - 手順を続けます。
 - いいえ - カートリッジをLSMに挿入します。
6. 次のコマンドを入力して、カートリッジをマウントします。

```
mount vol_id drive_id
```

`query drive` コマンドを使用して使用可能なドライブの ID を取得し、`query volume` コマンドを使用してライブラリカートリッジの ID を取得します。

7. マウント成功を示すメッセージが表示されましたか。

マウント成功メッセージ:

```
Mount: vol_id mounted on drive_id
```

- はい - 手順は終了です。
 - いいえ - エラーメッセージが表示された場合は、この検証手順を再度実行し、有効で使用可能なドライブおよびライブラリカートリッジを指定していることを確認します。まだマウントまたはマウント解除に失敗する場合は、サポートに連絡して支援を求めてください。
8. 次のコマンドを入力して、カートリッジをマウント解除します。

```
dismount vol_id drive_id force
```

ここで、`vol_id` はボリュームで、`drive_id` はステップ 6 で指定したドライブです。

自動データベースバックアップ

ACSL S は、24 時間おきの真夜中、または `acsss_config` のバックアップオプションに指定された時間と曜日に、データベースからディスクへのバックアップファイルを自動的に作成します。

テープへの手動バックアップの実行

ACSL S が作成する自動データベースバックアップに加えて、`bdb.acsss` ユーティリティを定期的に行って、遠隔地に保存でき、必要に応じて、データベースの障害回復に使用できるテープバックアップを手動で作成してください。

遠隔地のデバイスに転送される定期的なバックアップによって、ACSL S サーバーに障害が発生した場合に迅速な復元が可能になります。

次のあとに `bdb.acsss` を使用して、手動でデータベースをテープにバックアップします。

- `acsss_config` の実行。
- データベースのインポート。
- ライブラリ全体の監査。
- データベースの回復。

ACSL S サーバーに接続された、指定されたテープデバイスへのバックアップ

ACSL S サーバーに接続されている、指定されたテープデバイスに ACSL S データベースバックアップするには、次を実行します。

1. `acsss` としてログインします。
2. テープデバイスに空のテープを挿入します。
3. 端末ウィンドウから次のコマンドを入力します。

```
bdb.acsss -f tape_device
```

ここで、`tape_device` は、ACSL S サーバーに接続されているテープデバイスを指定します。

4. バックアップの進捗を報告するメッセージが表示されます。

次のメッセージが表示されるまで待ちます。

```
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. 次のメッセージが表示されるまで待ちます。

```
ACSL S database backup successfully completed.
```

例: ACSL S データベースをテープデバイス `/dev/rmt/0mn` にバックアップするには、次のコマンドを入力します。

```
bdb.acsss -f /dev/rmt/0mn
```

UNIX ファイルへのバックアップ

障害回復のために、ファイルがリモートディスク上にないかぎり、UNIX ファイルにバックアップすることはお勧めしません。「[bdb.acsss](#)」を参照してください。

ACSL S データベースを UNIX ファイルにバックアップするには、次を実行します。

1. *acsss* としてログインします。
2. 端末ウィンドウから次のコマンドを入力します。

```
bdb.acsss -f db_file
```

ここで、*db_file* は、ACSL S データベースを格納する UNIX ファイルを指定します。ファイルへの書き込みアクセス権限を持っている必要があります。

3. 次のメッセージが表示されるまで待ちます。

```
ACSL S database backup successfully completed.
```

回復と復元

このセクションでは、次の復元/回復手順について説明します。

- 破損または消失したデータベースの最新のバックアップへの復元
- 破損または消失したデータベースの指定された日時への復元
- ディスクの障害からの回復
- 障害が発生したサーバーの障害回復
- 特定のバックアップファイルからの回復
- データベース以外、ACSL S 制御ファイルの復元

これらの手順のほとんどは、*rdb.acsss* ユーティリティーを使用します。これは、最新のバックアップまたは指定された日時からのデータベースの復元、*bdb.acsss* によって作成されたバックアップを使用した障害回復、および *bdb.acsss* によって作成された ACSL S 制御ファイルの復元のためのオプションを提供します。これらのオプションの詳細については、「[rdb.acsss](#)」を参照してください。

注:

バックアップ後にカートリッジのホームセルが最後の場所から変更された場合、復元されたデータベースは最新になりません。マウント解除時のカートリッジの移動を防ぐには: 各 LSM がその ACS 内の唯一の LSM である必要がある (ほとんどの SCSI ライブラリに当てはまる) か、またはパススルーポート経由でほかの LSM に接続されているすべての LSM に対して拡張ストア機能が有効になっている必要があります。

詳細は、「[拡張ストア機能の使用](#)」を参照してください。すべての接続された LSM に対して拡張ストア機能が有効にされていないか、またはカートリッジが挿入または取り出し済みの場合、復元後にライブラリを監査して、データベースを最新にし、パススルーポート経由でほかの LSM に接続されているすべての LSM を有効にします。

注:

rdb.acsss ユーティリティの一般的なオプションとして、*-f* オプションを指定しないでください。データベースを外部ネットワークファイルまたは代替のテープデバイスにバックアップした場合、*rdb.acsss* を入力したあとにのみ *-f* オプションを使用してください。3 番目の回復オプションを選択します。プロンプトが表示されたら、*-f* と外部ネットワークファイルまたは別のテープデバイスへのパス名を入力します。詳細は、[6](#)を参照してください。

データベースの最新のバックアップへの復元

この手順では、自動バックアップによってローカルディスクに作成された最新のバックアップにデータベースを復元します。ACSL S 制御ファイルも復元されます。

破損または消失したデータベースを最新のバックアップに復元するには、次の手順を実行します。

1. *acsss* としてログインします。
2. ACSLS を無効にします。

acsss disable

3. 次のコマンドを入力します。

rdb.acsss

4. オプション 1 を選択します。

1.Restore from a current local disk backup

5. 手順については「[rdb.acsss](#)」を参照してください。
6. ACSLS を起動するには、次のコマンドを入力します。

acsss enable

障害が発生したサーバーからの回復

プライマリディスクとセカンダリディスクの両方が失われたか破損した場合に、障害回復のためにこの手順を使用します。

障害が発生したサーバーから回復するには、次の手順を実行します。

1. オペレーティングシステムをインストールします。
2. ACSLS をインストールします。

注意:

ディスクの障害の前に使用していた同じディレクトリに ACSLS をインストールする必要があります。

3. `acsss` としてログインします。
4. ACSLS を無効にします。

`acsss disable`

5. 次のコマンドを入力します。

`rdb.acsss`

6. オプション 2 を選択します。

`2.Restore from a previous tape or network file backup`

7. 手順については「[rdb.acsss](#)」を参照してください。
8. ACSLS を起動するには、次のコマンドを入力します。

`acsss enable`

9. デフォルトの設定を受け入れない場合は、`acsss_config` を実行して、自動バックアップの日時と保存期間を再指定する必要があります。

ACSLs 制御ファイルの復元

この手順では、ACSLs 制御ファイルを復元します。これらは、`access` 制御ファイル、固定ボリュームファイル、スクラッチ優先順位ファイル、カスタム `volrpt` ファイルなどの `data/external` ディレクトリ内のすべてのファイルを含むデータベース以外のファイルです。これらのファイルは、`bdb.acsss` のテープまたは外部ネットワークファイルへのバックアップから復元されます。

ACSLs 制御ファイルを復元するには、次の手順を実行します。

1. `acs`ss としてログインします。
2. ACSLS を無効にします。

```
acsss disable
```

3. 次のコマンドを入力します。

```
rdb.acss
```

4. オプション 4 を選択します。

```
Restore only ACSLS non-database control files
```

5. 手順については「[rdb.acss](#)」を参照してください。
6. ACSLS とデータベースを起動するには、次のコマンドを入力します。

```
acsss enable
```

第11章 レポートおよびロギング

この章では、次のことを行う方法について説明します。

- カスタムボリュームレポートを作成します
- ボリューム移動の統計情報のレポートを作成します

カスタムボリュームレポートの作成

`volrpt` ユーティリティを使用すると、ボリュームレポートを作成できます。詳細は、「`volrpt`」を参照してください。`$ACS_HOME/data/external/volrpt/owner_id.volrpt` は、カスタマイズしたボリュームレポートを作成するためのテンプレートとして実行または使用できるサンプル入力ファイルです。カスタマイズしたボリュームレポートを `$ACS_HOME/data/external/volrpt` ディレクトリに保存することもできます。この手順は、`volrpt` ユーティリティの入力ファイルを作成して、カスタムボリュームレポートを作成する場合に使用します。

カスタムボリュームレポートを作成するには、次の手順を実行します。

1. `acsss` としてログインします。
2. UNIX コマンドツールを開きます。
3. カスタムボリュームレポートディレクトリに変更します。

```
cd /home/ACSSS/data/external/volrpt
```

4. ボリュームレポートのサンプルファイルを、新しいカスタムファイルにコピーします。

```
cp owner_id.volrpt my.volrpt
```

`my.volrpt` は新しいファイルに割り当てる名前です。

5. `vi` などのテキストエディタを使用して `my.volrpt` ファイルを編集し、カスタムレポートに表示するフィールドおよび形式を指定します。
 - サンプルファイル内に一覧されているフィールドを指定できます。
 - 各エントリの形式は次のとおりです: `field_name field_length delimiter _length`

- フィールドの長さおよび区切りには任意のサイズを指定できます。指定したすべてのフィールドが、レポートを出力したときに 1 行に収まるようにしてください。
- 詳細な編集手順はサンプルファイルに示されています。

6. ファイルの編集が終了したら、保存します。

カスタムボリュームレポート

次の例に示すように、入力ファイルにフィールド、フィールドの長さ、および区切りの長さを指定することによって、カスタマイズされたレポートを作成します。

```
#####
#
# 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. ere 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
```

この入力ファイルから、表示されるカスタマイズされたレポートを生成します。ほかの使用方法としては、次の例に示すように、カスタマイズされたレポートを使用してボリュームの所有者を報告できます。

入力ファイルを使用した、カスタマイズされたボリュームレポートの例:

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
```

カスタムボリュームレポートの例

次の *volrpt* には、ACS 0 と 1、不在ボリュームと取り出し済みのボリュームが含まれており、ボリュームのステータスが示されています。

入力 *volrpt* オプション:

```
volrpt -f my.volrpt -a 0 1 -i
```

これらの制御文は、出力を選択およびフォーマットするために使用します。

カスタムボリュームレポート制御文の例:

```
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
```

不在ボリュームおよび取り出し済みのボリュームを示すカスタム *volrpt* の例:

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

```

ロギングボリューム統計レポートの作成

ボリュームログファイル (*acsss_stats.log*) を使用して、ボリューム移動の統計情報をログに記録できます。これらの統計は、ボリュームの位置が変更されたことを ACSLS が検出したときに記録されたエントリで構成されます。ACSL S は、挿入、取り出し、マウント、マウント解除が行われたとき、およびボリュームの位置が (通常は手動で移動することによって) 変更されたことを監査が検出したときにエントリをログに記録します。

acsss_config 構成プログラムを使用して、次を実行します。

- *LIB_VOL_STATS* 変数を使用して、ボリューム統計のロギングを有効または無効にします。
- ボリューム統計ログファイルの最大サイズを指定します。
- ボリューム統計ログファイルのロールオーバーファイルの数を指定します。

stats_report ユーティリティは、*acsss_stats.log* を使用してすべてのマウントおよびテープドライブの使用を報告します。

ボリューム統計ログファイルには、ボリューム統計のロギングが有効または無効のいずれであるかを通知する収集モードのエントリ、およびボリューム統計情報のエントリが含まれています。

ボリューム統計ログファイルのエントリの例:

```

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
```

前述の例では、収集モードのエントリは、統計収集が2014年6月30日午前8:53に開始され、同じ日の午前10:45に終了したことを示しています。これらの収集の開始時間および停止時間の間に、この収集期間のボリューム統計のエントリがあります。

ボリューム統計情報のエントリの形式は次のとおりです。

- *yyyy-mm-dd hh:mm:ss command*

vol_id home_loc function_loc client_host_ID

ここでは:

- *yyyy-mm-dd* はエントリの年、月、および日です。4桁の年形式がサポートされています。日付の形式は *acsss_config* を使用して指定します。
- *hh:mm:ss* は、エントリの時間、分、および秒です。
- *command* は、ボリュームを移動したか、(監査によって) ボリュームの移動が検出された ACSLS のコマンドまたはクライアントリクエストです。
 - *MOUNT* はマウントリクエストです。
 - *DISMOUNT* はマウント解除リクエストです。
 - *ENTER* は、手動または自動モードの *enter* リクエストです。
 - *EJECT* は取り出しリクエストです。
 - *監査* は監査リクエストです。
 - *ACSMV* は移動リクエストです。
 - *ACSCR* はボリューム回復アクティビティです。このアクティビティは ACSLS 処理によって自動的に生成されます。
- *vol_id* はボリューム識別子です。

- *home_loc* はボリュームのホーム (ストレージセル) 位置です。
- *function_loc* は、ボリュームを使用した次のリクエストのボリュームの位置です。
 - *mount* または *dismount* リクエスト

位置はトランスポート ID です。
 - *enter* または *eject* リクエスト

位置は CAP ID です。
- *audit* リクエストは、監査が次のいずれかのエラーを検出したことを指定します。
 - *STATUS_VOLUME_FOUND*

監査がデータベースに指定された位置と一致しない位置でボリュームを見つけました。
 - *STATUS_VOLUME_NOT_FOUND*

監査がデータベースに指定された位置でボリュームを見つけられませんでした。
 - ボリューム回復アクティビティ

位置はセル ID またはトランスポート ID であり、ボリューム回復によって次のいずれかの状況が検出されたことを示します。
 - › *STATUS_VOLUME_FOUND*

データベースに記録されていないボリュームが見つかり、追加されます。
 - › *STATUS_VOLUME_NOT_FOUND*

データベース内のボリュームが記録されている位置で見つからず、削除されます。
- *client_host_ID* は次のいずれかです。
 - クライアントアプリケーションのリクエストの場合は、ホストの IP アドレスです。
 - *cmd_proc* のコマンドの場合、*cmd_proc* が開始されたシェルの環境に環境変数 *LIBVOLSTATS_CMD_PROC_ID* が設定されている (ASCII 文字のみ) 場合、エントリはその環境変数の値の最初の 12 文字です。
 - *cmd_proc* コマンドの場合、環境変数 *LIBVOLSTATS_CMD_PROC_ID* が設定されていないか、ASCII 以外の文字が含まれている場合、エントリはローカルです。

第12章 ユーティリティーリファレンス

この章では、次の ACSLS ユーティリティーについて説明します。

- 「[acs_renumber.sh](#)」を使用すると、接続されているライブラリを再構成しなくても、ライブラリコンプレックス内の指定された ACS の識別子を変更できます。
- 「[acsss](#) マクロ」は、ACSLS の起動と停止、および保守とトラブルシューティングの制御とモニターを行います。
- 「[bdb.acsss](#)」は、ACSLS データベースおよび ACSLS 制御ファイルをバックアップします。
- 「動的構成 ([config](#)) ユーティリティー」は、ACSLS をオンラインで実行した状態のまま、ACSLS ライブラリ (およびコンポーネント) に構成変更を動的に実行します。それらの構成ユーティリティーを次に示します。
- 「[config acs](#)」は、動的に ACS を追加したり、既存の ACS およびそのコンポーネントを再構成したりします。
- 「[config drives](#)」 - 既存のドライブパネルで、ドライブの追加、ドライブタイプの変更、およびドライブの削除を動的に行います。
- 「[config lsm](#)」は、既存の LSM およびそのすべてのコンポーネントを動的に再構成します。これらのコンポーネントには、CAP、パネル、およびドライブが含まれます。
- 「[config ports](#)」は、ACS へのポート接続を動的に再構成します。
- 「[db_export.sh](#)」は、ACSLS のアップグレードインストールまたは再インストールを準備するために ACSLS データベースの情報および ACSLS 制御ファイルをエクスポートします。
- 「[db_import.sh](#)」は、[db_export.sh](#)ユーティリティーを使用したときにエクスポートされた ACSLS データベースの情報および ACSLS 制御ファイルをインポートします。
- 「[del_vol](#)」は、オフラインの LSM からボリュームを削除します。
- 「[drives_media.sh](#)」は、現在の ACSLS のリリースによってサポートされるすべてのドライブタイプ、メディアタイプ、およびドライブとメディアの互換性を表示します。
- 「[ejecting.sh](#)」は、大量の取り出し操作を迅速かつ効率的に行います。
- 「[free_cells.sh](#)」を使用すると、ACSLS によって制御されているライブラリ内の空きセルをモニターおよび管理できます。

- 「[getHba.sh](#)」は、ファイバチャネル HBA ポートを管理します。
- 「[get_license_info](#)」は、使用権ライセンスのソフトウェアでの実行が ACSLS で採用されなくなったため、ACSLs リリース 7.3.1 および 8.0.1 以降では削除されています。ACSLs によって制御されているライブラリの空きセル数を表示および管理するには、「[free_cells.sh](#)」を使用します。
- 「[greplog](#)」は、`acsss_event` ログをフィルタ処理して、特定のキーワードを含むメッセージを含めたり除外したりします。
- 「[install_scsi_Linux.sh](#)」は、ACSLs にライブラリを構成するときを使用できる `/dev/mchanger` リンクを作成します。
- 「[lib_type.sh](#)」は、指定された ACS ID に接続されている LSM の LSM タイプを返します。
- 「[licensekey.sh](#)」は、ライセンスキー検証が採用されなくなったため、ACSLs リリース 7.3.1 および 8.0.1 以降では削除されています。
- 「[moving.sh](#)」は、複数のカートリッジを 1 つ以上の LSM に移動します。
- 「[probeFibre.sh](#)」は、Emulex (LP10000) または QLogic (QLA2300) ファイバチャネル HBA の背後に接続されている各デバイスのモデル番号、リビジョンレベル、およびターゲット LUN アドレスを表示します。
- 「[rdb.acsss](#)」は、ACSLs データベースおよび ACSLS 制御ファイルを復元します。
- 「[showDevs.sh](#)」は、Solaris 上に構成されているすべての `mchanger` デバイスの詳細を表示します。
- 「[showDrives.sh](#)」は、ACSLs に接続されているすべての構成済みのドライブのリストを表示します。
- 「[stats_report](#)」は、ライブラリボリュームの統計情報を収集します。
- 「[userAdmin.sh](#)」は、ACSLs GUI ユーザーのパスワードを管理します。ユーザーの追加、ユーザーの削除、ユーザーの一覧表示、ユーザーパスワードの変更を行うことができます。
- 「[volrpt](#)」はボリュームレポートを作成します。
- 「[watch_vols](#)」は、CAP を使用して挿入されたボリュームに所有権およびプールの関連付けを自動的に割り当てます。

概要

ACSLs ユーティリティを使用する場合は、次の一般的なガイドラインに従ってください。

- 通常、この章で説明しているユーティリティは、ユーザー `acsss` が実行することが意図されています。これらを実行するために必要な権限および環境の依存関係を継承するには、ユーザー `acsss` としてログインしてください。

`su` を使用する場合は、`su - acsss` を使用してください。

- 次の操作を行なった場合は、`bdb.acsss` を使用してデータベースをテープに手動でバックアップすることをお勧めします。
 - ライブラリハードウェアの構成。
 - データベースのインポート。ACSL5 の新しいバージョンにアップグレードしたあとに、以前のバージョンで作成されたデータベースバックアップを使用しないでください。アップグレードしたらすぐに新しいバックアップを作成します。
 - データベースの回復。
- 正確で一貫性のあるデータベースが回復されるように、最新のデータベースバックアップを常に使用してください。

ユーティリティーが失敗した場合は、すべてのイベントログを保持します。これらのログは、問題の解決をサポートするために役立ちます。

レガシー起動/停止スクリプト

ACSL5 7.x で使用された起動スクリプトおよび停止スクリプトは、ACSL5 8.x ではサポートされません。

ACSL5 8.x は、Solaris Service Management Facility (SMF) に統合されたライブラリ管理アプリケーションを起動および停止するための新しいメカニズムを提供しています。これは、ACSL5 で使用されていた `rc.acsss` および `kill.acsss` に置き換わるものです。このメカニズムは、アプリケーションのステータスをモニターする機能も提供しています。

ACSL5 8.x を起動および停止するには `acsss` コマンドを使用します。単一のコマンド `acsss` によって、ACSL5 の起動、停止、およびモニタリングの機能が提供されます。このユーティリティーは `$ACS_HOME` ディレクトリにあり、すべてのユーザーがアクセスできます。

ユーティリティーコマンド

次のセクションでは、ACSL5 ユーティリティーについて説明します。

`acs_renumber.sh`

これは、接続されているライブラリを再構成しなくても、ライブラリコンプレックス内の指定された ACS の識別子を変更できる簡単なツールです。ライブラリ内のすべての LSM、CAP、ドライブ、およびボリュームは ACS との関係で識別されるため、割り当てる新しい ACS ID と各ライブラリリソースが一致するように、このユーティリティーはすべての各種データベーステーブルを更新します。

新しい論理ライブラリは現在アクティブなパターンを使用します。たとえば、ACS 0 を 1 に変更した場合、1001 および 1002 はそのままですが、ACS 1 内の新しい論理ライブラリは 2001 となります。ACS 6 を ACS 0 に変更した場合、7001 はそのままですが、ACS 0 内の新しい論理ライブラリ 1003 になります。新しく追加されるものは ACS に基づいて予測できますが、実際の対応関係はなくなります。

注:

このユーティリティーによって行われた変更は ACSLS サーバーにのみ適用され、これらのリソースを使用するクライアントアプリケーションには適用されません。このため、サーバーで ACS ID を変更したあとに、クライアントのデータベースを再構成する必要がある場合があります。

注:

このスクリプトを実行する前に、ACSLD を無効にする必要があります。

ACS に割り当てられている番号を変更するには、`acs_renumber.sh` を実行します。対話型セッションでは、変更するとクライアントアプリケーションに影響を与えることが最初に警告され、続行するかどうかを確認されます。

```
$ 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):
```

「y」と応答すると、変更が行われる前に、このルーチンによって既存のデータベースが自動的にバックアップされます。これにより、変更を元に戻す必要がある場合に、以前の構成に復元できます。(acs_renumber.sh ルーチンを繰り返して実行することによって変更を元に戻すこともできます。)

このルーチンは現在構成されている ACS のリストを表示し、それぞれについて、その ACS の番号を変更するかどうかを尋ねます。変更する場合は、割り当てる新しい値の入力を求められます。

```
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
```

入力を受け入れ (この例では、応答は 5 です)、ルーチンは保留中の変更の確認を求めます。

```
Change ACS-0 to ACS-5.
```

```
Correct? (y or n):
```

「y」と応答すると、ルーチンは関連するすべてのデータベーステーブルの更新を開始し、行なった変更のチェックポイントを作成するために、自動的にデータベースをバックアップします。

```
Updating tables:    Changing ACS-0 to ACS-5
```

```
acstable: 1 records
capttable: 4 records
celltable: 13424 records
drivetable: 128 records
handtable: 16 records
lmutable: 0 records
lsmtable: 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...
```

acsss マクロ

acsss マクロは、ACSL S に関連付けられているさまざまなサービスを起動および停止する主要な起動、停止、およびステータスのコマンドです。インストール内容によりませんが、ACSL S アプリケーションは Solaris または Linux システムにインストールされる最大 7 つのサービスで構成された集合体です。

- *acsdb* - ACSLS ライブラリデータベースを管理します。
- *acsls* - ライブラリ操作を実行するライブラリ制御ソフトウェア。
- *weblogic* - ACSLS GUI の Web サーバー。
- *surrogate* - java サービスと *acsls* の間の通信リンク。
- *rmi-registry* - 名前付き java オブジェクトおよびメソッドの検索サービス。
- *smce* - 論理ライブラリの SCSI メディアチェンジャーエミュレーション。
- *stmf* - 論理ライブラリのターゲットモードフレームワーク。

acsls サービスおよび *acsdb* サービスはすべてのインストールで共通しています。*weblogic*、*surrogate*、および *rmi-registry* サービスは、ACSL S GUI のサポートが構成されている場所に存在します。*smce* および *stmf* サービスは、(Solaris 上に) 論理ライブラリが構成されている場合にのみ適用されます。

すべてのサービスは、1 つのマクロ *acs* を使用して ACSLS ユーザーによって操作され、このマクロは異なるコンポーネント間の依存関係によって必要となる定義済みの順序でこれらのサービスを起動および停止します。このマクロは、Solaris の Service Management Facility (SMF) および Linux の *init.d* サービスユーティリティーにコマンドを発行して、実際の作業を行わせます。

形式

acs <command>

コマンドを指定せずに *acs* を入力すると、オプションのリストが表示されます。

オプション

コマンド	機能
<i>enable</i>	これは ACSLS に関連付けられているすべてのサービスを起動するためのデフォルトの方法です。有効にすると、各種サービスが有効な状態で維持され、システムのリブート後に自動的に再度有効になります。
<i>temp-enable</i>	<i>acs enable</i> と同じですが、システムのリブート後にサービスは再起動しません。
<i>maint-enable</i>	ACSL S データベースが関係しない一般的な保守作業での使用を目的としています。このオプションを使用すると、ACSL S が無効になっていても、GUI ユーザーは GUI インフラストラクチャーでログイン状態を維持できます。このメソッドは、マイナーなソフトウェアパッチのインストールを行う場合に使用されます。 <i>acsls</i> および <i>smce</i> サービスのいずれも有効になりません。
<i>db</i>	これは、 <i>db_export</i> 、 <i>db_import</i> 、および <i>acs config</i> を含むデータベース保守作業での使用が推奨されている制御モードです。ACSL S データベースエンジンを有効化し、ACSL S GUI を含むその他すべての ACSLS サービスを無効化します。
<i>disable</i>	ACSL S 処理の停止に使用するデフォルトメソッドです。これは完全な停止ではなく、 <i>acsls</i> および <i>smce</i> サービスが無効化されたあとも、データベースと GUI ログインセッションのアクティブの状態が維持されるため保守作業を実行できます。結果の状態は、 <i>acs maint-enable</i> と同じです。ACSL S とライブラリをアイドル状態にしてからサービスを無効化するため、サーバーを停止するのにもっとも安全なメソッドです。
<i>force-disable</i>	<i>acs disable</i> と同じですが、この操作では、アイドル状態になるのを待たずに <i>acsls</i> と <i>smce</i> を無効化します。
<i>shutdown</i>	これは、すべての ACSLS サービスを完全にシャットダウンします。ソフトウェアのインストールやアンインストールのほか、データベース (<i>acsdb</i>) または GUI インフラストラク

コマンド	機能
	チャー (rmi-registry および surrogate) のシャットダウンが必要になるその他の保守作業での使用を目的としています。
<i>status</i>	さまざまな ACSLS サービスの簡易的なステータスレポートを提供します。
<i>a-status</i>	<i>acsdb</i> サービスの動作ステータスを返します。
<i>d-status</i>	<i>acsls</i> サービスの動作ステータスを返します。
<i>g-status</i>	ACSLG GUI のステータスを表示します。
<i>l-status</i>	さまざまな ACSLS サービスの冗長ステータスサマリーを表示するほか、トラブルシューティング時の詳細分析に役立つログデータへのポインタも表示します。参照先のログはサービスの開始や停止に失敗したときに役立ちます。
<i>p-status</i>	<i>acsss status</i> と同様に、各サービス契約によってモニターされているさまざまなプロセス ID のリストが表示されます。
<i>w-status</i>	Weblogic サービスのステータスを表示します。
<i>timeout</i>	Solaris 上の <i>acsls</i> サービスの SMF 起動の <i>timeout</i> を報告します。
<i>legal</i>	ACSLG の法律上の注意点を英語またはフランス語で表示します。

ほとんどの場合、上位 3 つのコマンド (*enable*、*disable*、および *status*) のみを使用します。残りのコマンドは、ソフトウェアを保守する場合の利便性のためにあります。

bdb.acsss

bdb.acsss ユーティリティーは、ACSLG 環境を再構築するときに必要な ACSLG データベースの内容および ACSLG 制御ファイルをバックアップします。バックアップは、指定した tar ファイルまたはテープデバイス、またはデフォルトとして定義されているディレクトリ内に出力されます。

このユーティリティーは、ACSLG を停止することなく、ACSLG データベースのバックアップを実行します (ホットバックアップ)。

-f オプションを指定しない場合は、*/export/backup/<time_stamp>.tar* ファイルが作成されます。*time_stamp* は *bdb.acsss* コマンドが実行された時間です。*bdb.acsss* の実行後にテープの内容を確認するには、次の例を特定のテープデバイスに変更します。

Solaris の場合:

```
tar tvbf 2048 /dev/rmt/0mn
```

tar tvbf コマンドの実行後、テープは次のブロックに進みます。*tar tvbf* コマンドの実行後に *bdb.acsss* を実行する場合は、テープを巻き戻すか、再度位置設定してください。

形式

```
bdb.acsss [-f backup_file | tape_device]
```

オプションを指定せずに *bdb.acsss* を入力した場合は、このバックアップが実行された時点でデータベースを復元する機能を提供するデフォルトのバックアップが実行されます。

オプション

- *-f backup_file*

ACSL S データベースのバックアップを出力する UNIX ファイルを指定します。ファイルおよびディレクトリへの書き込み権限が必要です。

- *-f tape_device*

ACSL S サーバーに接続および構成されているテープデバイスを指定します。

使用方法

bdb.acsss ユーティリティは、必要に応じてデータベースの回復に使用できるバックアップを作成するために、ACSL S データベースをテープまたは外部ネットワークファイルにバックアップするために使用します。

次の操作を行なったあとに、*bdb.acsss* を使用してデータベースを手動でバックアップすることをお勧めします。

- *acsss_config* の実行。
- データベースのインポート。ACSL S の新しいバージョンにアップグレードしたあとに、以前のバージョンで作成されたデータベースバックアップを使用しないでください。
- ライブラリ全体の監査。
- データベースの回復。

例 1:

```
$ bdb.acsss -f /export/backup/my_backup
```

この例では、*my_backup* という名前のファイルが */export/backup* ディレクトリに作成されました。ファイルがある場所にファイルを保持するか、別のファイルシステム、別のサーバー、または書き込み可能な CD デバイスに移動するかを選択できます。

このファイルは、バックアップが実行されたときの状態にデータベースの復元するために使用できます。たとえば、バックアップが金曜日の午後 1:00 に実行され、復元が月曜日の午前 6:00 に実行された場合、データベースは金曜日の午後 1:00 の状態に戻ります。

この同じ `-f` オプションに、ファイル名の代わりにテープデバイスを指定でき、バックアップは指定されたテープデバイスに書き込まれます。

例 2:

```
$bdb.acsss -f /dev/rmt/0mn
```

この例では、テープデバイス `/dev/rmt/0mn` にテープアーカイブが作成されました。これは、あとで使用するためにオフサイトの場所に格納できます。

関連項目:

- [「rdb.acsss」](#)
- [10章「データベース管理」](#)

動的構成 (config) ユーティリティー

動的構成 (*config*) ユーティリティーを使用すると、ACSL S がオンラインで実行しているときに、ACSL S ライブラリ (およびコンポーネント) に構成の変更を実装できます。これらの構成の変更は、`acsss_config.log` ファイルに記録されます。

次の動的構成ユーティリティーがサポートされています。

- `config acs`
- `config drives`
- `config lsm`
- `config ports`

`config` ユーティリティーの使用には次の利点があります。

- ACSLS は実行を続けることができ、影響を受けないライブラリコンポーネントへのマウントリクエストを実行できます。
- ほかの構成情報はすべて変更せずに、指定したライブラリコンポーネントを再構成できます。たとえば、次を指定する場合:
 - ACS の場合、ほかの ACS の構成は影響を受けません。
 - LSM の場合、ほかの LSM の構成は影響を受けません。

- 。ドライブパネル、パネル上のドライブ、既存のすべてのドライブに対するマウントおよびマウント解除は影響を受けません。

次の点を理解することが重要です。

- 動的 *config* ユーティリティを使用するには、ACSL S が実行されている必要があります。
- *acsss_config* を使用して ACSLS の初期構成を作成する必要があります。6章「[ACSL S の動作を制御する変数の設定](#)」を参照してください。
- イベント通知はすべての動的構成変更を報告します。
- 動的構成を実行する前に、追加または再構成しているすべてのコンポーネントの準備ができていることを確認します。
- *acsss_config.log* ファイルは、表示されたメッセージに関する詳細を提供します。
- 構成変更の確認を行っていない場合は、`[[CTRL]]+C` を押して操作を取り消す (*cancel*) ことができます。
- 動的構成は、構成変更の前後に自動バックアップを実行します。
- 構成変更を確認したあとに取り消すことはできません。構成変更を元に戻す場合は、ACSL S を停止して、構成変更が行われる直前に取得されたバックアップを復元します。

構成変更の確認は 10 分後にタイムアウトします。

- ACS (または最後の ACS) のみを削除することはできません。
- ACS 内の最後の CAP、または ACSLS に定義されている最後のドライブを取り外さないでください。

動的構成の制限事項

動的構成ユーティリティには 2 つの重要な制限があります。

- ACS の削除、あるいはライブラリへのポート (接続) の削除または変更を行うことはできません。
- 既存の SCSI/ファイバで接続されたライブラリの場合は、*config acs* ユーティリティおよび *config lsm* ユーティリティでドライブ構成のみを更新できます。パネル構成または CAP 構成は更新されません。*config drives* および *config acs new* は、SCSI/ファイバで接続されたライブラリに対して制限なしに動作します。*config ports* ユーティリティは、SCSI/ファイバで接続されたライブラリではサポートされません。

解決方法:

動的構成でサポートされない構成変更の場合は、ACSL S を停止して *acsss_config* を使用する必要があります。

避けるべき操作

- ライブラリに対して大量の I/O が行われるため、動的構成を使用してライブラリおよびそのコンポーネントのステータス情報を表示しないでください

代わりに、*query* コマンドまたは *display* コマンドを使用します。

- 一度に複数の構成タスクを実行しようとしません。

一度に実行できる動的構成タスクは 1 つのみです。これにより、

- ACSL S と構成しているライブラリ間の I/O を原因とするパフォーマンスの問題が最小化されます
- 複数の構成タスク間の複雑なやり取りが回避されます。

config acs

config acs ユーティリティーを使用すると、次を行うことができます。

- ACS の追加、または既存の ACS とそのコンポーネントの再構成
- すべての ACS 番号を順番に割り当てることなくライブラリを構成または再構成できます。

たとえば、9310 ライブラリから SL8500 に移行してから、9310 を削除します。9310 には ACS 0、SL8500 には ACS 1 の番号が付けられます。*config acs* を使用すると、すべてのカートリッジとドライブを SL8500 に移行し、その後、SL8500 の番号を変更せずに 9310 を取り外すことができます。

- ACSL S を停止せずに、*config acs acs_id new* を使用して SL8500 ライブラリを追加します。
- 9310 から SL8500 にドライブを移動し、*config acs acs_id* を使用して両方の ACS のドライブ構成を更新します。
- 9310 からカートリッジを取り出して、SL8500 に挿入します。
- 最後に、ACSL S を停止する機能停止をスケジュールし、*acsss_config* を使用して構成から 9310 を削除します。9310 を構成しないでください。SL8500 が (デフォルトのゼロではなく) ACS 1 であることを忘れずに指定してください。
- パーティション分割された ACS の構成。
- LSM の追加または取り外し。*config lsm* ユーティリティーで行うことができるのは、既存の LSM の再構成のみであるためです。

各 ACS には少なくとも 1 つの CAP が必要です。これは、別のパーティションと共有される CAP にできます。ACSL S システム全体に対して少なくとも 1 つのドライブが構成されている必要があります。

たとえば、ACSL S が 4 つのライブラリをサポートしている場合、3 つのライブラリにはゼロ (0) 個のドライブを含めることができます。ただし、4 番目のライブラリには、少なくとも 1 つのドライブが含まれている必要があります。

形式

- 新しい ACS を追加するには、次のコマンドを入力します。

```
config acs new
```

- 既存の ACS を再構成するには、次のコマンドを入力します。

```
config acs acs_id
```

新しい ACS の追加

新しい ACS を追加するには:

1. 次のコマンドを入力します。

```
config acs new
```

2. 新しい ACS の ACS 番号を指定します。

ACSL S では、すべての ACS 番号を順番に割り当てることなく、ライブラリの構成または再構成を行うことができます。

すでに使用されている ACS 番号および最初の 5 つの使用可能な ACS 番号が表示されます。

新しい ACS の ACS 番号を入力します。

3. ACS をパーティション分割された SL8500 または SL3000 内に配置するかどうかを選択します。
 - 「y」を入力した場合は、ACS のパーティション ID を指定するように求められます。このパーティション ID は、SL コンソール上のパーティション ID と一致する必要があります。
 - これがパーティション分割されたライブラリではない場合、または SCSI/ファイバで接続されている場合は、「n」を入力します。

ACSLs は、パーティション分割された SCSI/ファイバ接続ライブラリ (SL500 など) をサポートしていません。また、パーティション分割された SCSI/ファイバ接続ライブラリには、パーティション ID はありません。

4. ACS への接続数に続けて、デバイスまたはホスト ID を入力します。

最大 15 接続を持つことができます。

注:

すべてのポートが同じ ACS に接続されることを確認します。

新しい ACS 構成が表示されます。

5. 新しい ACS の追加を確認します。

確認すると、構成情報が表示され、データベースが更新されます。

既存の ACS の再構成

ACS を再構成するときは、可能な場合、ACS をオンラインまたは診断モードにしてください。

ACS を再構成するには:

1. 次のコマンドを入力します。

```
config acs acs_id
```

古い構成と新しい構成が表示されます。

2. 新しい構成を確認します。

確認後、データベースが更新されます。

構成が変更されていない場合は、確認を求められずに構成が表示され、ユーティリティーが終了します。この例:

- ドライブタイプまたはシリアル番号あるいはその両方の変更のみ
- LSM シリアル番号の変更
- 4410 と 9310 の間の LSM タイプの変更のみ
- SL8500 ロボットなどのハンドの数の変更。

ただし、確認を必要とするほかの変更とともにこれらの変更が発生した場合は、新しい構成を確認します。その後、データベースが更新されます。

ACS およびそのコンポーネントがデータベースから削除されます。

config acs の制限

- SCSI 接続されたライブラリの場合、*config acs* はドライブの構成のみを更新します。追加、取り外し、または変更が行われたドライブを認識するには、SCSI 接続されたライブラリを IPL する必要があります。ライブラリを IPL するときには、すべてのドライブの準備ができていない必要があります。
- SCSI 接続されたライブラリの場合、*config acs* はパネルまたは CAP 構成を更新しません。それらを更新するには、ACSL を停止して *acsss_config* を使用する必要があります。

config drives

config drives ユーティリティを使用すると、既存のドライブパネル上のすべてのドライブを再構成できます。これには、ドライブの追加、既存ドライブのドライブタイプおよびシリアル番号の更新、データベースから削除されたドライブの削除が含まれます。

config drives ユーティリティは、ドライブ構成を動的に変更する (既存のドライブパネル上のドライブの取り付け、交換、または取り外しを含む) ために使用します。テープライブラリのハードウェア構成へのその他の変更 (ストレージセルの数または場所あるいはその両方の変更、CAP の数またはサイズの変更、ドライブパネルがあるストレージセルパネルの交換など) は、*config lsm* または *config acs* ユーティリティを使用して行う必要があります。

注:

- 変更されたドライブ構成を持つパネルを含む LSM は、オンラインまたは診断モードである必要があります。
- 再構成されているドライブパネルで、すべてのドライブの準備ができていない必要があります。
- 追加、取り外し、または変更が行われたドライブを認識するには、SCSI 接続されたライブラリを IPL する必要があります。ライブラリを IPL するときには、すべてのドライブの準備ができていない必要があります。
- 既存のドライブを新しいドライブに交換するときに、LSM、ドライブタイプをオンラインに変更した場合、または *config drives* を実行した場合、ドライブタイプおよびドライブのシリアル番号が自動的に更新されます。

形式

`config drive panel_id` または `config drives panel_id`

使用方法

既存のドライブパネル上のすべてのドライブを再構成するには:

1. 次のコマンドを入力します。

```
config drive panel_id または config drives panel_id
```

パネルの古いドライブ構成と新しいドライブ構成が表示されます。

2. 構成の変更を確認します。

確認後、データベースが更新されます。

- 構成を変更していない場合は、確認を求められずに構成が表示され、ユーティリティーが終了します。
- ドライブタイプまたはシリアル番号のみを変更した場合は、確認を求められずに ACSLS データベースが更新されます。

config lsm

このユーティリティーを使用すると、既存の LSM およびそのすべてのコンポーネントを再構成できます。これらのコンポーネントには、CAP、パネル、およびドライブが含まれます。

ACS に LSM を追加または削除する場合は、`config acs` ユーティリティーを使用する必要があります。

パネルを変更するときの手順:

- 取り外すか、変更しているパネルがあり、カートリッジが空にされている場合、LSM はオンラインのままにできます。
- 取り外すか、変更しているパネルがあり、カートリッジが含まれている場合は、LSM を再構成して影響を受けるパネルを監査するまで、影響を受ける LSM を診断に変更 (`vary`) することをお勧めします。この手順を実行しない場合、マウントおよびマウント解除が失敗することがあります。
- パネルを追加して、それらのパネルにカートリッジを手動で配置した場合は、監査を実行してデータベースを一致させます。

形式

```
config lsm lsm_id
```

使用方法

LSM を再構成するには:

1. 次のコマンドを入力します。

```
config lsm lsm_id
```

古い構成と新しい構成が表示されます。

パネルの横にある「y」は、パネルタイプが変更されたことを通知しています。詳細は、*acsss_config.log* ファイルを参照してください。

2. 新しい構成を確認します。

確認後、データベースが更新されます。

構成が変更されていない場合は、確認を求められずに構成が表示され、ユーティリティが終了します。

小さい変更は確認なしで自動的に行われます。例:

- ドライブタイプまたはシリアル番号あるいはその両方の変更のみ
- LSM シリアル番号の変更
- 4410 と 9310 の間の LSM タイプの変更のみ
- SL8500 ロボットなどのハンドの数の変更。

ただし、確認を必要とするほかの変更とともにこれらの変更が発生した場合は、新しい構成を確認します。その後、データベースが更新されます。

config lsm の制限

- SCSI 接続されたライブラリの場合、*config lsm* はドライブの構成のみを更新します。追加、取り外し、または変更が行われたドライブを認識するには、SCSI 接続されたライブラリを IPL する必要があります。ライブラリを IPL するときには、すべてのドライブの準備ができていない必要があります。
- パネル構成または CAP 構成は更新されません。それらを更新するには、ACSL5 を停止して *acsss_config* を使用します。

config ports

`config ports` ユーティリティーを使用すると、ACS にポート接続を動的に追加できます。

注:

すべての新しいポートは、既存のポートと同じ ACS に接続される必要があります。

`config acs acs_id` を実行してから、`config ports acs_id` を実行します。

1 つの ACS を別の ACS に置き換える場合、またはポート接続アドレスを変更する場合は、ACSLs を停止して `acsss_config` を使用します。

形式

`config ports acs_id` または `config port acs_id`

使用方法

ポートを追加するには:

1. 次のコマンドを入力します。

`config port acs_id` または `config ports acs_id`

指定した ACS の現在のポート接続が表示されます。

2. 指定した ACS のポート接続の数を入力します。

最大 15 接続を持つことができます。

3. デバイスまたはホスト ID を指定します。

注:

新しいポートが既存のポートと同じ ACS に接続されることを確認します。

古い構成と新しい構成が表示されます。

ポートの順序の変更は構成変更ではありません。接続は現在データベースに記録されている順序で表示されます。

4. 新しい構成を確認します。

確認後、データベースが更新されます。

構成を変更していない場合は、構成が表示されて、ユーティリティーが終了します。

config ports の制限

`config ports acs_id` ユーティリティは、SCSI/ファイバで接続されたライブラリではサポートされません。

`config ports` ユーティリティは、ライブラリへのポート (接続) を削除または変更しません。ACSL S を停止して `acsss_config` を使用する必要があります。

db_export.sh

`db_export.sh` ユーティリティは、ACSL S のアップグレードインストールまたは再インストールを準備するために、ACSL S データベーステーブルのデータおよび ACSLS 制御ファイルをエクスポートします。

注:

ACSL S が実行されている場合は、`db_export.sh` ユーティリティを実行できません。`db_export.sh` を実行する前に、`acsss disable` を実行します。

形式

```
db_export.sh -f [ db_file | tape_device ]
```

オプション

- `-f db_file`

ACSL S データベースのバックアップを出力する UNIX ファイルを指定します。ディレクトリおよびファイルの両方に対する書き込みアクセス権。

注記:

- データベースをファイルにエクスポートする場合、ファイルは非揮発性のディレクトリ内にある必要があります。ACSL S を再インストールする場合、再インストール作業によって `$ACS_HOME` または `$ACSD_BKUP (/export/backup など)` ディレクトリが破棄されます。エクスポートされたファイルはファイルシステム内の別の場所に配置してください。
- ファイル名をパスなしで指定した場合、`db_export` は現在の作業ディレクトリ内にそのファイル名でデータベースファイルを保存します。同じディレクトリの `<filename>.misc` という名前のファイルに ACSLS 制御ファイルが保存されます。
- 一部のテープライブラリ、テープドライブ、またはカートリッジのメディアタイプをサポートしていない ACSLS の以前のリリースにデータベースをエクスポートする場合は、データベー

スをエクスポートする前に、サポートされていないテープライブラリを構成から削除し、ライブラリからテープドライブとカートリッジを取り出します。

- `-f tape_device`

ACSL S サーバーに接続および構成されているテープデバイスを指定します。

オプションを指定しない場合、システムはテープデバイスをデフォルトで選択します。

使用方法

`db_export.sh` ユーティリティーは、ACSL S の再インストールまたは ACSLS のアップグレードインストールを準備するために使用します。

関連項目:

- [「db_import.sh」](#)
- [「rdb.acsss」](#)
- [「データベースのエクスポート」](#)
- [「ファイバ用の mchanger の新しいプラットフォームへの移行」](#)

db_import.sh

`db_import.sh` ユーティリティーは、`db_export.sh` ユーティリティーを使用したときにエクスポートされた ACSLS データベーステーブルのデータおよび ACSLS 制御ファイルをインポートします。ACSL S 制御ファイルは `$ACS_HOME/data/external` にあり、ユーザーが定義できる変数および ACSLS の構成が含まれています。これらは、アクセス制御設定、スクラッチ優先順位、拡張ストア LSM、カスタム `volrpt` 設定、ボリューム属性 (`watch_vols` ユーティリティー用)、などを指定します。また、`db_import.sh` ユーティリティーは、災害回復機能、および別のオペレーティングシステムに移動する場合または以前のリリースから移行する場合にカスタマイズされた動的変数を維持する機能を提供します。

形式

```
db_import.sh -f [ db_file | tape_device ]
```

オプション

- `-f db_file`

`db_export.sh` によって作成された UNIX ファイルを指定します。

- `-f tape_device`

ACSL S サーバーに接続および構成されているテープデバイスを指定します。

オプションを指定しない場合、システムはテープデバイスをデフォルトで選択します。

使用方法

`db_import.sh` ユーティリティは、`db_export.sh` ユーティリティを使用してエクスポートした ACSLS データベースをインポートするために使用します。

注:

ACSL S が実行されている場合、`db_import` ユーティリティは実行されません。`db_import.sh` を実行する前に、`acsss disable` を実行します。

関連項目:

- [「db_export.sh」](#)
- [「rdb.acsss」](#)
- [「データベースのエクスポート」](#)
- [「ファイバ用の mchanger の新しいプラットフォームへの移行」](#)

del_vol

`del_vol` ユーティリティは、ライブラリ内のボリュームを検査します。ボリュームを見つけることができない場合、`del_vol` は、リクエストに応じて、ボリュームが見つからないまたは不在としてマークするか、データベースから削除します。

ボリュームの参照されているすべての場所を検証できない (LSM がオフラインである場合、ドライブの準備ができていない場合など) 場合、`-n (no_confirm_flag)` がオンである場合を除き、ボリュームを不在としてマークするか、削除することの確認を求められます。ボリュームを取り出すと、ボリュームおよびそれに関連付けられている情報 (スクラッチプールのメンバーシップ、現在のロックと保留中のロックなど) がデータベースから削除されます。

`-q (quiet_flag)` が入力されている場合を除き、ボリューム関連の情報が表示されます。複数のオプションを使用する場合は、個別のオプションの形式にするか、連続した文字列の形式にすることができます。

`-n` オプションを指定した場合を除き、ボリュームを不在としてマークするか、削除する前に、`del_vol` によって確認を求められます。

- ボリュームがライブラリ内にある場合、データベースではアクティブなボリュームのままになります。

- ボリュームがライブラリ内で見つからない場合、ボリュームを削除することを指定した場合を除き、不在としてマークされます。
- ボリュームが配置されている可能性があるセルまたはドライブにアクセスできない (ライブラリまたはドライブがオフラインであるか、稼働していないため) 場合は、ボリュームを削除することが指定されている場合を除き、ボリュームが見つからないとしてマークされます。

注:

- このユーティリティーは、ライブラリ内で見つかったボリュームを削除しません。
- `del_vol` ユーティリティーを使用すると、不在ステータスまたは取り出し済みステータスの期限切れを待つことなく、ボリュームレコードを削除できます。
- `del_vol` ユーティリティーをサポートするには、ACSLs が実行されている (有効化されている) 必要があります。

形式

```
del_vol [-n] [-d] [-f] [-q] vol_id
```

オプション

- `-n`

確認なしモード。ユーザーに確認を求めずに、見つからないボリュームを不在としてマークするか、削除します。

- `-q`

非出力モード。データベースから抽出された情報の一部を出力しません。

- `-d`

不在ステータスまたは取り出し済みステータスの期限切れを待つことなく、指定されたボリュームを削除します。

- `-f`

テープドライブ内でスタックしたカートリッジを再挿入することを許可します。`-f` オプションを使用すると、ボリュームがライブラリ内に実際に存在するかどうかを確認せずに、ボリュームを削除するか、不在としてマークできます。これにより、障害のあるテープドライブにあったボリュームをデータベースから削除できます。このオプションを使用すると、ドライブの修復中に、ドライブからボリュームを取り出し、それをデータベースから削除してから、通常にライブラリで使用するためにボリュームを再挿入できます。

- `vol_id`

削除するボリュームシリアル番号。

注:

ボリュームシリアル番号にドル (\$) 記号が含まれている場合は、単一引用符で囲みます。例: `del_vol 'AB$001'`

使用方法

`del_vol` は、カートリッジをオフラインの LSM から取り出し、オンラインの LSM に再挿入することによって、自動的にマウントできるようにするために使用できます。

- オフラインの LSM からカートリッジを取り出します。
- `del_vol` を使用して、カートリッジを不在としてマークします。
- オンラインの LSM にカートリッジを挿入します。

このユーティリティを使用するには、ACSLs およびデータベースが起動および実行されている (アイドル状態ではない) 必要があります。

注:

データベースからカートリッジを誤って削除した場合は、削除されたカートリッジのホームセルを含むサブパネルを監査 (*audit*) して、カートリッジをデータベースに再入力する必要があります。システムの回復中に `del_vol` を実行すると、予期しない結果となることがあります。回復シーケンスは、*vary LSM online* を実行しているときにも行われます。

例

カートリッジ情報を印刷せずに、カートリッジ `U01102` を削除するには:

```
del_vol -q U01102
```

drives_media.sh

このルーチンは、現在のリリースの ACSLS によってサポートされるすべてのドライブタイプ、メディアタイプ、およびドライブとメディアの互換性を表示します。通常、情報は標準出力に表示されます。

サポートされているライブラリ、ドライブタイプ、メディアタイプ、およびドライブとメディアの互換性の最新のリストについては、ACSLs の製品情報ガイドを参照してください。

形式

```
drives_media.sh [ -f, -h ]
```

オプション

- `-f`

情報は3つのファイルに書き込まれます。

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

- `-h`

構文メッセージが表示されます。

ejecting.sh

`ejecting.sh` ユーティリティーは、大量取り出しボルト操作を容易にします。このユーティリティーは、指定された CAP およびボリュームのリストから動作し、ジョブ全体が完了するまで、並列で複数の取り出し操作を実行します。ユーザーがボリュームをソート順に取り出すことをリクエストした場合を除き、このユーティリティーは各ボリュームをもっとも近くにある指定された CAP に取り出します。

可能な場合、その操作がオペレータによる不要な CAP 操作を減らすことができるときは、もっとも近い CAP の代わりに空き領域がある近くの CAP が使用されます。この一般的なアプローチによって、カートリッジの移動が減り、不要な LSM パススルー移行が排除され、オペレータの全体的なワークロードが軽減されます。

すべての取り出しジョブは、操作が起動された単一のシェルウィンドウでモニターされ、要約が表示されます。複数の XTERM ウィンドウで使用する場合は、以降の `-x` オプションを参照してください。特定の CAP が満杯になり、オペレータが「*Remove cartridges from CAP*」を行う準備ができるたびに、ユーザーに通知されます。操作全体が完了すると、オペレータに通知されます。

過去の10日間のすべての取り出し操作の結果を示す一連のログが、`$ACS_HOME/log/ejectingLogs` ディレクトリに保持されます。個別の各ログは、操作が完了したときに取得されたタイムスタンプによって識別されます。

形式

標準: `ejecting.sh [-dmox] -c <CAP リスト> -v <ボリュームリストファイル>`

ポリシーの指定: `ejecting.sh [dmox] -p <ポリシーファイル>`

レガシー形式: `ejecting.sh <CAP ID> <ボリュームリストファイル>`

オプション

- `-c <CAP リスト>`

コマンド行で、CAP ID のリストはスペースで区切ります。CAP リスト内のすべての CAP は、同じ ACS 内にある必要があります。

例: `-c 0,1,0 0,1,1 0,5,0 0,5,1 0,9,0`

アスタリスク (*) を使用したワイルドカード表現は、すべての LSM または LSM 内のすべての CAP、あるいはその両方を指定するために有効です。

例:

- `-c 0,1,*` (LSM-1 内のすべての CAP)
- `-c 0,*` (すべての LSM の CAP-0)
- `-c 0,*,*` (すべての LSM のすべての CAP)

オンラインで使用可能な CAP のみが操作のために選択されます。ワイルドカード表現を指定した場合は、ゼロ以外の優先順位の CAP のみが選択されます。

ACS は数値表現である必要があり、ワイルドカードによって参照できません。

大きいライブラリのコンテキストでは、ボルト操作の速度と効率性に関して、CAP の選択が大きな違いとなることがあります。指定する CAP が多すぎると、複数の部分的に満たされた CAP を保守するために、オペレータの不要な作業が増加することがあります。特定のワークロードに指定する CAP が少なすぎると、ボトルネックが発生してロボット装置の待ち時間が増加することがあります。大きいボリュームリストの場合の経験則として、選択する CAP が異なるレーン上のライブラリコンプレックスおよび異なるライブラリモジュールに分散されるようにしてください。多数のボリュームに対して少数の CAP を選択するときは、ライブラリを領域に分割して、各領域の中央にある CAP を選択します。

- `-v <ボリュームファイル>`

これは単純なテキストファイルへのフルパスまたは相対パス名を使用したファイル指定です。ファイルには、取り出すボリュームを表す VOL-ID (VOLSER) のリストが含まれるようにしてください。指定した CAP と同じ ACS 内に含まれているボリュームのみが影響を受けます。マウントされているボリュームは取り出されません。

- `-p <ポリシーファイル>`

ポリシーファイルはフルパス名または相対パス名を使用して指定します。このテキストファイルには、CAP およびボリュームに対して定義されたポリシーが含まれています。ファイルの形式には、単語「caps:」の後ろに操作に使用する CAP のリストが続くもの、および単語「vols:」の後ろにボリュームリストファイルのフルパス名が続くものがあります。

例:

```
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

display オプションは、*eject* 操作を実行する前に、ボリュームと CAP の割り当てを表示するように *ejecting.sh* に指示します。ボリュームと移動先の CAP の完全なリストを表示するか、各 LSM から各 CAP に移行するボリューム数を示すサマリーを単に表示するかを選択できます。

表示内容を確認したあとに、続行するか、操作を中止するかを選択できます。

- -m

「04」から「99」のオペレータメッセージコードでジョブにラベルを付けます。サポートされているライブラリの場合、この数値コードがオペレータコンソールに表示されます。

- -o

ボールト操作でボリュームをソート順にスタックする必要がある場合があります。このオプションを使用すると、ルーチンは、ボリュームリストで見つかった順番、および指定された CAP の順序に従って、指定されたボリュームを一覧表示されている CAP に移動します。CAP の順序は、すべてのボリュームが取り出されるまで、最初の CAP から最後の CAP まで繰り返されます。

注:

ボリュームから CAP の順序が優先されるため、このオプションは LSM パススルールートを制限することによってボリュームの移動を最適化しようとしません。

- -x

個別の各 CAP の *eject* のペイロードに専用の XTERM ウィンドウを使用します。全体的な大量取り出し操作を行なっているときに、個々の取り出しジョブの追跡を続ける場合に、

このオプションが役立つことがあります。各 CAP の *eject* が起動されると XTERM セッションがポップアップし、対応する *eject* が完了すると閉じられます。

Windows 端末から操作する場合は、X11 対応のソフトウェアがインストールされていることを確認してください。X11 は Solaris または Linux では標準です。ローカルマシンから ACSLS サーバーへの表示アクセス制御を開く必要があります。

例: `xhost + <acsls_server_hostname>`

このユーティリティは、ログイン ID (*who am i*) をチェックして表示内容の送信先を判断します。これを表示するには、ローカルコンソールまたはデスクトップマシンから ACSLS サーバーに直接ログインする必要があります。

レガシー形式

`ejecting.sh <CAP ID> <ボリュームファイル>`

このユーティリティのレガシー形式は維持されています。これは、1 つの CAP ID、およびボリュームの単純なリストを含むファイルのパス名を受け取ります。その後、CAP のサイズに最適化された一連の *eject* コマンドを生成し、結果として得られるコマンドを標準出力に表示します。指定された CAP を介して完全なボリュームリストを取り出すために必要であるため、この表示には多くの *eject* コマンドが含まれています。

オペレータは、レガシー *ejecting.sh* の出力を直接 *cmd_proc* にパイプして、操作を実行できます。

例:

```
ejecting.sh 0,1,0 /export/backup/myVolumeList | cmd_proc -lq
```

または、出力をファイルにリダイレクトして、あとで *cmd_proc* にリダイレクトできます。

例:

```
ejecting.sh 0,1,0 /export/backup/myVolList > /tmp/eject.dat cmd_proc  
-lq < /tmp/eject.dat
```

ejecting.sh のログ

ejecting.sh のすべてのインスタンスは、`$ACS_HOME/log/ejectingLogs/` ディレクトリにログ記録されます。各ログファイルは日付とタイムスタンプから名前が付けられます。次に例を示します。

ejecting.log.14-Oct_13:13:10

各 *ejecting.log* には、オペレータシェルに表示された取り出し操作全体が要約されています。検出されたエラーはここに表示されます。

ejecting.log には、次の理由でユーティリティーによって無視されたボリュームのリストが含まれていることがあります。

- ボリューム ID が無効です。
- ボリュームが指定された ACS に含まれていません。
- ボリュームは使用中です。

ejectingLogs ディレクトリに蓄積されたファイルは 10 日後に削除されます。10 日を経過した古いログは、*ejecting.sh* の新しい各呼び出しで削除されます。

free_cells.sh

free_cells.sh ユーティリティーを使用すると、ACSLs によって管理されているライブラリ内の空きセルをモニターおよび管理できます。このユーティリティーは、LSM、ACS、および ACSLS サーバーの空きセル数を報告します。

このユーティリティーは、*\$ACS_HOME/utis* ディレクトリにあります

形式

free_cells.sh

オプション

- *-a*

各 ACS、LSM、および ACSLS サーバー (の管理対象) の空きセル、割り当て済みのセル、およびセルの合計数が表示されます。

例

各 LSM 内の空きセル

- LSM が L5500 ではない (空きセルの合計のみを一覧表示)
 - オプションの選択なし

LSM 1,3

空きセルの合計数 = 2,345

- -a オプション (割り当て済みのセルおよびセルの合計数も一覧表示)

LSM 1,3

空きセルの合計数 = 3,345

割り当て済みのセル = 3,155

セルの合計数 = 6,500

- LSM が L5500 である (LTO および非 LTO の空きセルを個別に一覧表示)

- オプションの選択なし

LSM 0,2

LTO の空きセル = 573

非 LTO の空きセル = 467

空きセルの合計数 = 1,040

- -a オプション (割り当て済みのセルおよびセルの合計数も一覧表示)

LSM 0,2

LTO の空きセル = 573

非 LTO の空きセル = 467

空きセルの合計数 = 1,040

割り当て済みのセル = 4,460

セルの合計数 = 5,500

各 ACS 内の空きセル

ACS 内に LSM が 1 つだけがある場合は、次の例に示すように ACS のみが一覧表示されま
す。

- ACS に L5500 が含まれていない

- オプションの選択なし (空きセルの合計数のみ表示)

ACS 1

空きセルの合計数 = 5,342

- -a オプション (割り当て済みのセルおよびセルの合計数も表示)

ACS 1

空きセルの合計数 = 5,342

割り当て済みのセル = 5,658

セルの合計数 = 11,000

- ACS に L5500 が含まれている (LTO および非 LTO の空きセルを個別に表示)

- オプションの選択なし

ACS 0

LTO の空きセル = 1,573

非 LTO の空きセル = 968

空きセルの合計数 = 2,541

- -a オプション (割り当て済みのセルおよびセルの合計数も表示)

ACS 0

LTO の空きセル = 1, 573

非 LTO の空きセル = 968

空きセルの合計数 = 2,541

割り当て済みのセル = 2,959

セルの合計数 = 5,500

ACSL S サーバーによって管理されている空きセル

- ACS に L5500 が含まれていない

この場合、空きセルの合計数のみが表示されます

- オプションの選択なし

ACSL S Server

空きセルの合計数 = 7,883

- -a オプション

ACSL S Server

空きセルの合計数 = 7,883

割り当て済みのセル = 14,117

セルの合計数 = 22,000

- ACS に L5500 が含まれている

L5500 ライブラリが管理されている場合、LTO の空きセルおよび空きセルの合計数のみが表示されます。L5500 ライブラリは LTO と非 LTO のセルが区別される唯一のライブラリであるため、非 LTO の空きセルは表示されません。

- オプションの選択なし

ACSL S Server

LTO の空きセル = 1,573

空きセルの合計数 = 7,883

- `-a` オプション

ACSL Server

LTO の空きセル = 1,573

空きセルの合計数 = 7,883

割り当て済みのセル = 14,117

セルの合計数 = 22,000

getHba.sh

`getHba.sh` ユーティリティは、ファイバチャネル HBA ポートを管理します。

形式

`getHba.sh`

`getHba.sh` ユーティリティはインストール時に実行され、`install.sh` によって呼び出される `install_acsss.sh` によって呼び出されます。このユーティリティは、新しい HBA がシステムに追加されたとき、または HBA ポートが再配置されたときに直接実行できます。このユーティリティは、ACSL 論理ライブラリへのクライアントアクセスポイントを明らかにするために、イニシエータからターゲットモードに変更する適切な HBA ポートを識別します。

このユーティリティを使用する最適な方法は、`getHba.sh` を実行する前に FC 接続を設定することです。これにより、`getHba.sh` が既存の接続についての役立つ情報を表示できます。

このユーティリティは、最初にターゲットモードのアダプタがすでに構成されているかどうかを評価します。ターゲットポートが存在しない場合、ユーティリティのフローは次に説明されているように続行されます。ユーティリティが既存のターゲットポートを検出した場合は、次の選択肢のメニューが表示されます。

必要なアクションを選択します。

1. 既存の HBA ポートの構成を維持します。
2. 追加のターゲットモードのポートを構成します。
3. 既存のターゲットポートをイニシエータモードに復元します。

オプション 説明

1 このユーティリティを終了します。

オプション 説明

- | | |
|---|--|
| 2 | ユーティリティーは現在イニシエータモードで動作しているポートを一覧表示します。ポートに「リモート HBA に接続しました」と表示される場合は、他方の端にイニシエータがあり、それを ACSLS ターゲットポートになる潜在的な候補にすることを意味します。ポートに「ターゲットデバイスに接続しました」と表示される場合は、テープライブラリまたはディスクが接続されており、そのポートはターゲットモード操作のための良い選択ではない可能性があります。 |
| 3 | ターゲットモード操作のために構成されている各ポートを識別し、そのポートをイニシエータモードに復元することの確認を求めます。 |
-

オプション 2 の例

ターゲットモードに変更するローカルの HBA ポートを選択します。次のリストから選択します。

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.

注:

論理ライブラリ機能を使用しない場合は、「none of these」を選択します。

選択を行うと、確認を求められます。

```
2
HBA Port WWN 2101001b32055d85 /pci@0,0/pci10de,377@f/pci1077,143@0
Is this correct? (y or n):
```

変更する機会が与えられます。「n」と応答すると、使用可能なポートのリストがふたたび表示され、選択を求められます。「y」と応答すると、再構成する追加のポートがあるかどうかを尋ねられます。

このユーティリティーは、ターゲットグループおよびターゲットグループのメンバーの追加に進み、ターゲットモードの変更を有効にするためにサーバーをリブートするように指示されます。

ターゲットグループの作成: 2101001b32255d85

オプション 3 の例

このオプションを使用すると、既存のターゲットの構成を解除し、ネイティブモードのイニシエータとして HBA を復元できます。

```
# 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.
```

get_license_info

このユーティリティは、使用権ライセンスのソフトウェアでの実行が ACSLS で採用されなくなったため、ACSLs リリース 7.3.1 および 8.0.1 以降では削除されています。ACSLs によって制御されているライブラリの空きセル数を表示および管理するには、「[free_cells.sh](#)」を使用します。

greplog

特定のキーワードを含むメッセージが含まれるように、または除外されるように *acsss_event* ログをフィルタリングするには、*greplog* ユーティリティを使用します。このルーチンの構文は、UNIX の「*grep*」関数と似ています。*greplog* は *acsss_event.log* に使用するために設計されていますが、レコードが空白行で区切られた任意のタイプのメッセージファイルに対して機能する可能性があります。

形式

```
greplog [-v|i] <keyword> <logfile>
```

オプション

- *-v*

オプションです。このオプションは、キーワードが含まれているものを除く、ログ内のすべてのメッセージを表示します。

- *-i*

オプションです。このオプションを指定すると、指定されたキーワードの大文字小文字が無視されます。

- *-keyword*

キーワードを含む完全な複数行のメッセージを返します。

- `-logfile`

ログファイルのリスト。

使用方法

`greplog` はログファイル用に設計されているため、その単語を含む単一の行ではなく、キーワードを含む完全な複数行のメッセージを返します。`-i` オプションを使用すると、`greplog` は指定された keyword の大文字小文字を無視します。`-v` オプションを使用すると、`greplog` は、キーワードを含むメッセージを除く、ログ内のすべてのメッセージを表示します。`greplog` は `acsss_event .log` に使用するために設計されていますが、レコードが空白行で区切られた任意のタイプのメッセージファイルに対して機能する可能性があります。

install_scsi_Linux.sh

`install_scsi_Linux.sh` ユーティリティーは、ACSLs にライブラリを構成するときに使用できる `/dev/mchanger*` リンクを作成します。それらの `mchanger` 名は、ライブラリによって報告されるシリアル番号を使用して生成されるようになり、SAN ファブリックの変更またはサーバーのリブート (どちらもライブラリの基となるデバイスパスが変更される可能性があります) が行われても変わらない信頼できる識別子を ACSLS に提供します。

作成された `/dev/mchanger` リンクおよび関連付けられているライブラリについての情報は、`showDevs.sh` ユーティリティーを使用すると、スクリプトの出力の一部として表示されます。そのユーティリティーは、スタンドアロン操作として実行して (`mchanger` リンクが作成されたあとに)、ライブラリの情報を表示することもできます。

形式

`install_scsi_Linux.sh`

出力例:

```
=====
# 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

このルーチンは、指定された ACS ID に接続されている LSM の LSM タイプを返します。共通したタイプの複数の LSM が構成に存在する場合は、複数の LSM に対して単一のタイプのみが返されます。

形式

lib_type.sh <ACS ID>

licensekey.sh

ライセンスキー検証が採用されなくなったため、ACSL S リリース 7.3.1 および 8.0.1 以降では削除されています。

moving.sh

moving.sh ユーティリティは、複数のカートリッジを 1 つまたは複数の LSM に移動します。このユーティリティは、移動するカートリッジを一覧しているファイルを読み取ります。それらのカートリッジ:

- 1 つまたは複数の LSM 内のカートリッジ
 - 同じ LSM またはほかの LSM 内のほかのパネルに移動されるパネル上のカートリッジ
 - 選択したカートリッジのグループ

moving.sh の制限:

- *vol_list_file* 内のすべての移動先の LSM およびカートリッジは、同じ ACS 内にある必要があります
- 移動先の LSM がオフラインであるか、空きセルが含まれていない場合は、カートリッジはその LSM に移動されません

注記:

- *moving.sh* ユーティリティは、ACSL S が実行されている場合にのみ実行されます
- 内部的には、*moving.sh* は一度に 1 つのカートリッジのみを移動して、ライブラリのパフォーマンス (マウントおよびマウント解除) に影響を与えないようにされています

- ボリュームの個別のリストを作成したあとに、複数の移動ユーティリティーを並列で実行できます。次のことを確認してください
 - 移動先 LSM が同じです。すべてのカートリッジを収容するための空きセルが LSM 内に十分あることを確認します
 - 1 つの SL8500 ライブラリ内に移動している場合 - エレベータは 2 つのみであるため、一度に 2 つを超える移動ユーティリティーを実行してもパフォーマンスは向上しません

形式

`moving.sh -f vol_list_file -t lsm_id` または `lsm_id` のリスト

ここでは:

- `-f vol_list_file`

移動するボリュームのリストを含むファイルの名前。

注:

ボリューム ID は次の規則に従う必要があります。1 行に 1 つのカートリッジ ID。vol_id は有効な ACSLS ボリューム ID である必要があります。vol_id の末尾または先頭にスペースが含まれている場合は、単一引用符または二重引用符で囲む必要があります。

- `-t lsm_ids`

カートリッジが移動される 1 つ以上の LSM ID を指定します。各 LSM ID は、スペースで区切り、同じ ACS に属するようにしてください。

使用方法

`moving.sh` ユーティリティーは、カートリッジのリストをほかの LSM に移動する場合、または同じ LSM 内の 1 つのパネルから別のパネルに移動する場合に使用します。

カスタムボリュームレポートまたは `display volume` コマンドを使用すると、LSM から移動するボリュームのリストを含むファイルを作成できます。

`moving.sh` ユーティリティーは次の場合に使用します。

- SL8500 が最初にパーティション分割または再パーティション分割され、1 つ以上のレーン (LSM) が既存のパーティション (ACS) から取り外される場合、`moving.sh` はパーティションから取り外される LSM からパーティションに残る LSM にカートリッジを移動できません。

- LSM が ACS から取り外された場合、*moving.sh* は ACS 内に残っている LSM にカートリッジを移動できます。

たとえば、SL8500 がライブラリコンプレックス (ACS) から取り外される場合、*moving.sh* は取り外される SL8500 からライブラリ内に残る LSM にカートリッジを移動します。これは、9310 の LSM が 9310 の ACS から取り外されるときにも適用されます。

- ストレージ拡張モジュールが SL8500 から取り外される場合、取り外されるパネルからライブラリに残るパネルにカートリッジを移動できます。
- ライブラリのパフォーマンスを最適化するには、カートリッジをアーカイブするために使用される LSM (ドライブがないか、ほとんどない) に非アクティブなカートリッジを移動します。これにより、新しいアクティブなカートリッジのためのドライブを持つ LSM のスペースが解放されます。

ボリュームリストファイルの作成

開始する前に、LSM から移動するボリュームのリストを含むファイルを作成する必要があります。*volrpt* (カスタムボリュームレポート) または *display volume* コマンドのいずれかを使用できます。

- *vol_list_file* の作成

```
volrpt -d -f custom_volrpt_input -l lsm_id > vol_list_file
```

ここで、*custom_volrpt_input* ファイルは次のとおりです。

```
VOLUME_ID 6
```

出力例:

```
$ volrpt -d -f my_custom -l 0,2 > my_file_list
$ cat my_file_list
ABC744
ABC748
ABC756
ACS151
EN0823
000373
```

- *display volume* コマンドを使用した *vol_list_file* の作成

- a. ボリュームのリストを表示します。

例:

```
display volume * -home acs,lsm,panel,*,* -f vol_id
```

この例は、`-home` パラメータによって識別されたパネル上のすべてのボリュームを選択しています。行および列にはワイルドカードが指定されています。`vol_id` のみが出力されます。

出力例:

```
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
```

- b. `vol_list_file` を作成して名前を付けます。
- c. ボリュームのリスト (`display` コマンドから作成された) をこのファイルにカット&ペーストします。
- d. 出力を編集します。

`vol_list_file` には、空白行および先頭のスペースを含めることはできません。次の `vi` コマンドを使用してそれらを削除します。

```
:%s/^[ ]*/g
```

これを行わないと、次の例に示すようにエラーメッセージが表示されます。

```
$ moving.sh -f my_file_list -t 0,2
Error in file my_file_list.
Invalid entry
ABC748
ABC756
ACS151
```

EN0823

このエラーメッセージは、ボリューム ABC748 および ABC756 の前に余分なスペースがあるために生成されました。

カートリッジのグループを移動する手順

以降の手順では、次のことを行う方法について説明します。

- ACS から LSM を削除する前にカートリッジを移動する
- パネルを変更または削除する前にカートリッジを移動する

ACS から LSM を削除する前にカートリッジを移動する

ライブラリを再構成または再パーティション分割したあとに、LSM を ACS から取り外した場合は、LSM のすべてのカートリッジにアクセスできなくなります。このため、LSM を取り外す前に、そのすべてのカートリッジを ACS 内に残る LSM に移動してください。次の手順を使用します。

- パーティション分割された SL8500 内のレガシーパーティションからルール (LSM) を取り外す場合。
- LSM を ACS から取り外す場合。ACS には 9310 または SL8500 ライブラリを含めることができます。

1. 新しい構成を計画します。

- パフォーマンス向上のためにカートリッジとドライブを編成します。
- ライブラリ構成を変更する直前に LSM を空にします。
- 空にする LSM にあるカートリッジの数、およびカートリッジを移動する LSM 内の空きセルの数を判別します。

`free_cells.sh -a` を使用して、これらの LSM 内のカートリッジ (割り当て済みのセル) および空きセルの数を判別します。

2. 移動および再構成をスケジュールします。

- システムへの影響が最小限になるように移動をスケジュールします。

カートリッジの移動には時間がかかり、ライブラリの再構成または SL8500 の再パーティション分割は中断を伴います。

- ターゲット LSM に移動されるカートリッジのための十分な空きセルがあることを確認します。必要に応じて、カートリッジを取り出してスペースを解放します。

3. 取り外される LSM 内のすべてのドライブをオフラインに変更 (*vary*) します。

これにより、次のことを防ぐことができます。

- LSM 内のロボットの競合。
- この LSM へのマウント。

そうしないと、この LSM にマウントされているカートリッジが LSM 内の新しいホームセルにフロートされ、空にしようとしている LSM が満たされてしまうことがあります。

4. 次のコマンドを使用して、空にしている LSM を診断モードに変更 (*vary*) し、アクセスを *cmd_proc* のみに制限します。

```
vary lsm lsm_id diagnostic
```

例: *vary lsm 0,1 diagnostic*

5. 次のコマンドを使用して、カスタム *volrpt* を実行し、空にしている LSM 内のすべてのカートリッジをファイルに出力します。

```
volrpt -f custom_volrpt_input -l from_lsm_id > move_vols_list
```

ここで、*custom-volrpt_input* は次のとおりです。

```
VOLUME_ID 6
```

例: *volrpt -f volrpt_input -l 0,1 > move_vols_list*

詳細については、「[ロギングボリューム統計レポートの作成](#)」を参照してください。

6. 次のコマンドを使用して、空にしている LSM からカートリッジを移動します。

```
moving.sh -f move_vols_list -t dest_lsm_id
```

7. カートリッジが LSM に挿入されていたり、「フロート」されたりしていることがあるため、*volrpt* を使用して LSM が空であることをチェックします。

```
volrpt -l from_lsm_id
```

空でない場合は、カスタム *volrpt* をふたたび実行して、現在 LSM にあるボリュームを選択します。次に、*moving.sh* を再度実行します (手順 3 および 4)。

注:

元のボリュームのリストを使用して *moving.sh* を再度実行しないでください。

8. 空にしている LSM をオフラインに変更 (*vary*) して、ボリュームがそこに移動されることを防ぎます。

```
vary lsm lsm_id offline
```

注:

LSM をパーティションまたは ACS あるいはその両方から取り外します。

9. *config acsacs_id* または *acsss_config* のいずれかを使用して、ACS を再構成します。

パーティションからセルを取り外す前のカートリッジの移動

注:

SL3000 はドライブおよびセルレベルにパーティション分割でき、SL8500 は拡張パーティション分割を使用してドライブおよびセルアレイレベルにパーティション分割できます。あるパーティションから別のパーティションにセルが割り当て直されている場合は、それらのセル内のカートリッジは孤立し、以前に含まれていたパーティションではアクセスできなくなります。ほかのパーティションを管理するホストが、カートリッジ上のデータを上書きする可能性があります。

パーティションの境界が変更されるときにカートリッジが孤立することを防ぐために、再パーティション分割する前に、ライブラリによって、パーティション内に残るセルにそれらが移動されます。SL3000 は単一の LSM であるため、既存の ACSLS の移動コマンドは機能しません。それらは単にライブラリ内のどこかに移動されます。パーティションから削除される別のセルにそれらが移動される可能性もあります。

次のいずれかの方法を使用してカートリッジを移動します。

- StorageTek ライブラリコンソール (SL Console) を使用します。

ライブラリを監査して、ボリュームの場所を監査します。

詳細および手順については、『SL8500 ユーザーズガイド』または『SL3000 ユーザーズガイド』を参照してください。

- 次の ACSLS 手順を使用します。
 - 「[volrpt](#)」または「[display コマンドオプションの使用](#)」を使用して、ボリュームの場所を表示します。
 - 次の *display* コマンドを使用して、特定のパネル内で使用可能な (空の) セルのリストを表示します。

```
display cell a,l,p,*,* -status empty -f status
```

詳細については、「[display コマンドオプションの使用](#)」を参照してください。

- LSM ID の代わりに空きセルを指定することによって、カートリッジを特定のセルに移動します。1 つのセルを移動する場合は、`move` コマンドを使用します。

```
move AAAAAA a,l,p,r,c
```

パネルを変更または削除する前にカートリッジを移動する

9310 でセルパネルをドライブパネルに変更する前、または SL8500 内のストレージ拡張モジュールを取り外す前に、カートリッジを移動する必要があります。

手順 1 - 4 は [ACS から LSM を削除する前にカートリッジを移動する](#) と同じ手順です。

手順 5: 空にしているパネルのカートリッジを選択し、それらをファイルに出力します。

- カスタム `volrpt` を実行して、空にしている LSM 内のすべてのカートリッジをファイルに出力します。(ホームセル ID に) パネル番号を含めます。

```
volrpt -f custom_volrpt_input -l from_lsm_id > move_vols_list_1
```

ここで、`custom-volrpt_input` は次のとおりです。

```
VOLUME_ID 6
CELL_ID    14
```

空にしているパネルのボリュームを選択して、それらの `vol_id` を `move_vols_list_2` に出力します。

- `display volume` コマンドを使用して、空にしているパネル内のカートリッジを選択します。

```
display volume * -home acs,lsm,panel,*,* -f volume > move_vols_list_2
```

これは、`-home` パラメータによって識別されるパネル上のすべてのボリュームを選択します。行および列にはワイルドカードが指定されています。`vol_id` のみが出力され、出力がファイルに書き込まれます。

出力を編集し、先頭のスペースおよび末尾の空白行を削除します。

注:

宛先または「移動先」LSM がソースまたは「移動元」LSM と同じで、複数のパネルが空の場合は、一部のボリュームが空のパネルに戻されます。そのパネルからボリュームを選択し、それらを繰り返し移動して、パネルをクリアする必要があります。

手順 6 - 9 は **ACS から LSM を削除する前にカートリッジを移動する** と同じ手順です。

手順 10. `config lsm lsm_id` または `acsss_config` を使用して、LSM を再構成します。

例

- LSM 0,4 から LSM 0,0 および 0,1 へのカートリッジの移動

カートリッジを LSM 0,4 から LSM 0,0 および 0,1 に移動するには、`volrpt` を使用して LSM 0,4 内のカートリッジのリストを含むファイルを作成してから、次のように `moving.sh` ユーティリティーを実行します。

出力例:

```
$ 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
```

- LSM 0,4 0,5 0,6 および 0,7 から LSM 0,0 0,1 0,2 および 0,3 へのカートリッジの移動

各 LSM を隣接する LSM に移動することによってパフォーマンスを最適化するには:

- `volrpt` を使用して、LSM 0,4 0,5 0,6 および 0,7 内のカートリッジのリストを含むファイルを準備します。
- 個別の UNIX コマンド端末で、同時に 4 つの `moving.sh` ユーティリティーを実行します。

ソース LSM と移動先 LSM、および使用されるパススルーポートはすべて異なるため、`moving.sh` の個別のインスタンス間に競合はありません。

出力例:

```
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
```

パフォーマンスのためのカートリッジの管理

`moving.sh` ユーティリティーは、非アクティブなカートリッジをアーカイブ LSM に移動するために使用できます。アーカイブ LSM は、マウントされる確率が低いカートリッジを格納する LSM (ドライブがないか、ほとんどない) です。SL8500 内の上部レールは、CAP に直接アクセスできないため、アーカイブ LSM に適しています。

ライブラリにある必要がない非アクティブなカートリッジは、取り出すことができます。自動マウントに使用可能である必要がある非アクティブなカートリッジは、アーカイブ LSM に移動してください。

非アクティブなカートリッジをアーカイブ LSM に移動するには:

1. 非アクティブなカートリッジを識別します。たとえば、過去 3 か月間にアクセスされていないカートリッジを選択するには、次の手順に従います。
2. カスタム `volrpt` を実行して、検査している LSM 内のすべてのカートリッジを出力し、結果をファイルに出力します。

```
volrpt -f custom_volrpt_input -I from_lsm_id > move_vols_list_1
```

ここで、`custom-volrpt_input` は次のとおりです。

```
VOLUME_ID 6
```

```
ACCESS_DATE 15
```

3. `access_date` が 3 か月より前のカートリッジを選択し、移動するボリュームのリストを含むファイルにそれらの `vol_id` を出力します。
4. 非アクティブなカートリッジをアーカイブ LSM に移動します。

```
moving.sh move_vols_list_2 archival_lsm_id
```

参照:

- [「display コマンドオプションの使用」](#)
- [「volrpt」](#)

probeFibre.sh

このユーティリティは、最新のファイバチャネル HBA の背後にあるすべての直接接続されたライブラリまたは SAN 接続されたライブラリを表示します。

`probeFibre.sh` ユーティリティは、各ファイバ接続されたライブラリのモデル番号、LUN ID、および WWPN (World Wide Port Name) を表示します。`probeFibre.sh` ユーティリティは、各ライブラリの `mchanger` デバイスが作成される前でも実行できます。

このユーティリティは、`root` でアクセスする必要があります。

形式

```
probeFibre.sh [-v] [-p]
```

オプション

引数なし。

各ライブラリデバイスのベンダー、モデル、LUN ID、および WWPN を表示します。

- `-v`

ホストバスアダプタ (HBA) のモデル番号、各イニシエータポートの WWPN、および各ポート (WWNN を含む) 上で検出されたライブラリデバイスを含む構造化された出力を生成します。

- *-p*

vendor:model:version:driver:target:lun:wwpn (各フィールドはコロンで区切られます) を含む出力を生成します。

rdb.acsss

rdb.acsss ユーティリティーは、自動バックアップ機能または *bdb.acsss* ユーティリティーのいずれかによって作成されたバックアップを使用して、ACSLS データベースと ACSLS 制御ファイルを復元します。ACSLS 制御ファイルは *\$ACLS_HOME/data* 内にあり、ACSLS 用のいくつかの異なる環境変数を定義します。これらは、アクセス制御設定、スクラッチ優先順位、拡張ストア LSM、カスタム *volrpt* 設定、およびボリューム属性 (*watch_vols* ユーティリティー用)などを指定します。

テープバックアップから復元する場合は、テープから ACSLS データベースおよび制御ファイルを復元する前に、テープデバイスを巻き戻すか、位置設定してください。*rdb.acsss* を実行する前に、次のいずれかのコマンドを使用して、テープを巻き戻すか、バックアップがある場所に位置設定します。

```
mt -f /dev/rmt/0mn rewind
mt -f /dev/rmt/0mn nbsf 1
```

形式

rdb.acsss

メニューオプション

rdb.acsss を実行すると、次の例に示すように、メニューに 4 つのオプションが表示されます。

```
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

ローカルディスク上の現在のすべての ACSLS バックアップファイルが一覧表示されます。

説明: データベースはそのバックアップの時点に復元されます。ACSLs 制御ファイルはバックアップのみから復元されます。バックアップはデフォルトのバックアップディレクトリ (`$ACSD_BKUP`) に保存されます。データベースは一覧および選択されたデータベースバックアップの状態に復元されます。通常は 8 つの異なる日付が一覧表示されますが、`csss_config` に設定されているデータベースの保持期間によって異なります。

使用法: このオプションは壊れたデータベースを復元するために使用します。このオプションを使用すると、すべてのバックアップが表示され、表示されている任意のデータベースバックアップを復元できます。

例:

```
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.
```

適切なバックアップの目的の日付と時間を入力する必要があり、データベースがその時点で復元されます。

2. Restore from previous tape or file backup

説明: このオプションは、別のファイルシステム (NFS など) またはバックアップデバイス (テープなど) にコピーされたデータベースを回復する場合に選択します。ACSLs 制御ファイルが復元されます。

使用法: データベースをサーバーまたはまったく異なるサーバーに復元する必要があるハードウェア障害などの壊滅的なイベントの場合に使用します。プラットフォーム (OS バージョン/アップデートおよび ACSLS リリース/PUT レベル) が同じである必要があります。

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'.

ACSLS データベースは指定したファイルから回復されます。

-f [backup_file | tape_device] を入力してください。

例 1: -f backup_file でファイルを指定

```
Please mount tape (if used) and enter backup source: -f /export/  
backup/my_backup.bak
```

これは、*my_backup.bak* というバックアップを復元します。データベースおよび ACSLS 制御ファイルの両方が復元され、ACSLS はバックアップが実行されたときの状態になります。

例 2: テープデバイスに作成されたバックアップの復元

テープデバイスに作成されたバックアップの復元では、同じオプションを使用しますが動作が少し異なります。バックアップがテープデバイスに作成されるときに、テープ上に tar アーカイブが作成されますが、それには名前がありません。テープからバックアップを復元するときは、テープデバイスのみを指定します。

ヒント: テープデバイスで **no rewind** を使用してください。

```
Please mount tape (if used) and enter backup source: -f /dev/  
rmt/0mn
```

この場合、デバイス `/dev/rmt/0mn` が対象となり、有効なデータベースバックアップがあることが確認されます。バックアップが存在していて有効である場合は、それが復元されます。

テープを巻き戻す手順:

`rdb.acsss` を試行する前に、テープを巻き戻すか、バックアップファイルがある正しい場所に位置設定する必要があります。

注:

`tar tvbf` コマンドの実行後、テープは次のブロックに進みます。`tar tvbf` コマンドを実行したあとに `rdb.acsss` を実行する場合は、テープの巻き戻し/再位置設定を行なったことを確認します。

- a. テープは、次のコマンドを使用して巻き戻し/位置設定できます。

```
mt -f /dev/rmt/0mn rewind または mt -f /dev/rmt/0mn nbsf 1 --->
SOLARIS
```

- b. `bdb.acsss` のあとにテープの内容を確認するには、次のコマンドを使用します。

```
tar tvbf 2048 /dev/rmt/0mn ---> SOLARIS
```

3. *Restore database only (do not include ACSLS control files)*

説明: このオプションはデータのみを復元する機能を提供します。一部の環境では、ACSLs データベースをそのデータを含めて復元する必要があるが、ACSLs のデータベース以外の制御ファイルを復元する必要がない場合があります。

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*

説明: ACSLS 制御ファイルのみを復元します。`$ACS_HOME/data/internal` ディレクトリにあるファイルを復元する前に、既存のファイルの末尾に「.bak」という拡張子が追加されてバックアップが作成されます。

```
$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
```

これは、`$ACS_HOME/data/external` にあるファイルには当てはまりません。回復する前に、ACSLs 制御ファイルのバックアップは実行されません。

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]:

例:

```
Please enter -f [ backup_file | tape_device ]: -f $ACSDB_BKUP/my_file.bak
```

- `-f -$ACSDB_BKUP/my_file.bak` は、指定されたファイルから ACSLS 制御ファイルを回復します。
- `-f /dev/rmt/0mn` は、指定されたテープデバイスから ACSLS 制御ファイルを回復します

5. Exit

`rdb.acsss` ユーティリティーを終了すると、デフォルトのディレクトリ `$ACSDB_BKUP` へのバックアップが開始されます。

参照:

次の回復手順:

- 「データベースのエクスポート」
- 「データベースのエクスポート」
- 「bdb.acsss」

showDevs.sh

`showDevs.sh` ユーティリティは、`/dev` ディレクトリ内の各 `mchanger` インスタンスに関連付けられている重要なデバイス属性を表示します。重要な属性には、ライブラリモデル番号、リビジョンレベル、セルの容量、および接続されているドライブの数が含まれます。追加の属性は、次のオプションを使用して表示できます。

形式

```
showDevs.sh [-w][-s]
```

オプション

このユーティリティは、いくつかのオプションを使用して実行できます。

引数なし。

このオプションは、各 `mchanger`、ライブラリモデルとコードレベル、およびセルとドライブの数を表示します。

- `-w`

World Wide Name - 基本情報とともに、接続されているライブラリの WWPN を表示します。

- `-s`

シリアル番号 - 基本情報とともに、ライブラリのシリアル番号を表示します。

注:

サーバー側の HBA 情報 (HBA のポートの WWPN を含む) および接続されているすべてのライブラリの WWPN を表示するには、`root` ユーザーとして `probeFibre.sh` ユーティリティを使用します。

showDrives.sh

このユーティリティは、ACSL5 に接続されているすべての構成済みのドライブのリストを表示します。ドライブの場所の単純なリストは、ドライブタイプによってソートされます。冗長 (`-v`) オプションを使用した場合、このユーティリティは、各ドライブのドライブ状態、ドライブステータス、および割り当てられている論理ステータスを示すサマリーを表示します。

形式

```
showDrives.sh [-v]
```

stats_report

`stats_report` ユーティリティーは、ライブラリボリュームの統計レポートを生成します。このユーティリティーを実行するには、`acsss` としてログインする必要があります。

形式

```
stats_report [vol_statsX.log ...]
```

ここでは:

`vol_statsX.log` -

1. このオプションの引数を使用すると、1つ以上のアーカイブされたボリューム統計情報のログファイル名を指定できます。

(アーカイブされたファイルの形式は `vol_statsX.log` です (ここで、 $0 \leq X \leq 8$ です)。

1つのアーカイブされたファイルを入力として使用します。

```
$stats_report vol_stats0.log
```

時間中心およびドライブ中心のレポートは、ユーザー入力ファイルの名前がレポートファイル名の前に追加されて生成されます。

たとえば、`vol_stats0.log` を指定した場合、レポートは次の名前ですべての `$ACS_HOME/log` ディレクトリに生成されます。

```
vol_stats0_drive-centric.txt および vol_stats0_time-centric.txt
```

2. アーカイブされたすべてのボリューム統計ファイルのレポートを一度に生成するには、次の手順に従います。
 - a. 個別のファイルから完全なログを生成します

```
$cd $ACS_HOME/log
$cat vol_stats8.log .... vol_stats0.log
acsss_stats.log > vol_statsXXXX.log
```

```
where vol_statsXXXX.log
```

(文字列 `vol_stats` が必須ですが、XXXX は FULL などに行うことができます) はすべての `vol_statsX.log` (ここで、 $0 \leq X \leq 8$) および `acsss_stats.log` を逆順に連結したファイルです。

- b. `stats_report` を実行します

```
$stats_report vol_statsXXXX.log
```

レポートは、`vol_statsXXXX_drive-centric.txt` および `vol_statsXXXX_time-centric.txt` として生成されます。

ファイル名を引数として指定しない場合、時間中心およびドライブ中心のレポートは `$ACS_HOME/log/acsss_stats.log` から生成されます。

使用方法

- `stats_report` は、現在の `acsss_stats.log` を使用して、ボリューム統計情報の 2 つのレポートを準備します。ライブラリボリュームの統計情報の収集を有効にするには、`LIB_VOL_STATS` 変数を設定します。これは、`acsss_config` (オプション 3) の処理またはコマンド行のコマンド `dv_config -p LIB_VOL_STATS` を使用して行うことができます。ACSL5 は、ログがデフォルトサイズの 500K バイトに達すると、9 個の `acsss_stats.log` ファイルを自動的にまとめて保持します。
- ログファイルのサイズおよび保持するファイル数は、変数 `LIB_STATS_FILE_NUM` および `VOL_STATS_FILE_SIZE` を介して制御されます。これらの変数は、上記で説明した `LIB_VOL_STATS` と同様の方法を使用して設定します。
- レポートの 2 つのタイプ:
 - `drive-centric.txt`

このレポートには、ドライブの順序付けされたリストが含まれています。各ドライブレコードには、ドライブにマウントされているすべてのカートリッジ、リクエスト、リクエストの時間、およびマウントの経過期間が含まれています。

- `time-centric.txt`

注:

このレポートには、1 時間単位で一覧されたドライブリソースの使用状況が含まれています。各期間のレコードには、リクエスト、特定のドライブ、その期間のそのドライブのマウント数、およびその期間のドライブの使用時間が含まれています。

ドライブの使用時間が1つの期間で60分を超えている場合は、マウントが2つの期間に及んでいるため、2番目の期間には一覧表示されないことを示しています。`stats_report`によって作成される最初のレポートはドライブの表示です。

注:

- ログに `DISMOUNT` レコードがあるが、対応する `MOUNT` レコードがない場合、次の理由である可能性があります。
 - ログがロールオーバーされました、または
 - 不明なロギングの問題が原因で、操作が記録されました。

この場合、生成されるレポートからレコードが除外されます。

- ログに `MOUNT` レコードがあるが、対応する `DISMOUNT` レコードがない場合、次の理由である可能性があります。
 - `DISMOUNT` がまだ行われていません、または
 - 不明なロギングの問題が原因で、操作が記録されませんでした。

この場合、マウント経過時間は上記の場合を示す `-1` に設定されます。これらのレコードは、時間中心レポートの合計マウント経過時間の計算から除外されます。

- サマータイムから標準時間に移行する場合に、計算されたマウント経過時間が負になるシナリオが発生することがあります。それらを抑制するために、マウント経過時間の絶対値が使用されます。

userAdmin.sh

メニュー起動型の `userAdmin.sh` ユーティリティーは、ACSL S GUI ユーザーのパスワードを管理します。これは、`$ACS_HOME/install` ディレクトリにあります。ユーザーの追加、ユーザーの削除、ユーザーの一覧表示、ユーザーパスワードの変更を行うことができます。このユーティリティーを使用するには、WebLogic が動作している必要があります。動作していない場合は、メニューが表示される前に、このユーティリティーによって WebLogic が起動され、オンラインであることが確認されます。

このユーティリティーは、`root` によって実行され、`acsls_admin` 認証が必要になります。`acsls_admin` ユーザーアカウントは、ACSL S 8.4 のインストール時に構成されます。

ユーザーの追加またはユーザーのパスワードの変更を行う場合、ユーザー名および割り当てるパスワードの入力を求められます。パスワードはサイズおよび有効な文字に関する WebLogic の条件に対して検証されます。

ユーザーを削除するときに、そのアカウントの GUI セッションがまだアクティブであることがあります。ユーザーがログアウトするか、セッションを終了すると、そのユーザーは再度ログインできなくなります。セッションを強制的にすぐに終了する唯一の方法は、GUI を再起動することです。ACSLs GUI を再起動する (すべてのセッションを終了します) ためのオプションが提供されています。

このユーティリティを使用して *acsls_admin* ユーザーのパスワードを変更することはできません。*acsls_admin* のパスワードを変更またはリセットする必要がある場合は、次の手順に従います。

1. *wlinstall.sh* ユーティリティを実行します。

```
$installDir/wlinstall/wlinstall.sh
```

2. *userAdmin.sh* を実行して、残りのユーザーアカウントを再設定します。

形式

```
userAdmin.sh
```

例

```
# ./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

volrpt ユーティリティーはボリュームレポートを作成します。

形式

```

volrpt [-s vol|loc|use] [-d] [-f filename] [-z] [-a|-l|-v
identifier_list] [-i]

```

オプション

- -s

ソート順序を指定します。このオプションを指定しない場合、デフォルトではボリューム ID によってソートされます。このオプションを指定する場合は、次のいずれかの値を指定する必要があります。

- *vol*

ボリューム ID でソートします。

- *loc*

ボリュームのホーム位置でソートします。

- *use*

ボリュームの使用状況 (マウント数) によってソートします。

- *-d*

出力に改ページまたはヘッダー情報を含めないことを指定します。この出力は、*pr* などのほかのプログラムの入力として使用できます。

- *-f filename*

filename はカスタム *volrpt* テンプレートを指定します。

- *-z*

識別子フィールドをゼロで埋めます。

- *-a*

指定された ACS にレポートを制限します。複数の ACS を指定できます (空白を使用して *acs_id* を区切ります)。

- *-l*

指定された LSM にレポートを制限します。複数の LSM を指定できます (空白を使用して *lsm_ids* を区切ります)。

- *-v*

指定したボリューム (またはボリュームの範囲) にレポートを制限します。1 つの *vol_id*、空白で区切られた *vol_id* のリスト、または *vol_id-vol_id* によって示されたボリュームの範囲を指定できます。

- *identifier_list*

-v、*-a*、および *-l* オプションで説明されています。これは、ACS、LSM、およびボリューム (またはボリュームの範囲) のリストです。

- *-i*

不在のカートリッジおよび取り出し済みカートリッジを含む、すべてのボリュームを報告します。

このオプションを指定しない場合、不在のカートリッジおよび取り出し済みカートリッジは報告されません。

使用方法

volrpt ユーティリティは、ライブラリカートリッジのレポート (物理的な位置、履歴、属性、および使用状況など) を作成するために使用します。*volrpt* を使用して、データベースを復元したあとに検証することもできます。*-a*、*-l*、または *-v* オプションを使用して、レポートする

ACS、LSM、またはカートリッジを指定できます。これらのオプションをいずれも指定しない場合、`volrpt` は ACS 0 のみを報告します。

注:

先頭および末尾の空白に関する特別な考慮事項。先頭または末尾にスペースが含まれているボリュームに引数を指定する場合は、引数を単一引用符で囲んでください。単一引用符が 1 つのシェルコンポーネントから別のシェルコンポーネントに渡されるようにするには、引用符をエスケープ文字でタグ付けする必要があります。UNIX では、標準のエスケープ文字はバックスラッシュ (`\`) です。

例:

ローカルマシンで先頭にスペースがあるボリューム ID を参照する `volrpt` コマンドを記述するには、次のようにコマンドを送信します。

```
volrpt -v '/0000/'-/'9999/'
```

リモートシェル (`rsh`) を使用して同じコマンドを送信するには、引数全体を二重引用符で囲みます。

```
rsh <acsls_hostname> -l acsss bin/volrpt -v "'/ 0000/'-/' 9999/'"
```

次の例は、ボリューム ID、場所、ラベルタイプ、メディアタイプ、および使用状況の履歴のフィールドが含まれている標準のボリュームレポートを示しています。

```
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
" " " " " ""
" " " " " ""
" " " " " ""
```

`Volume Type/Media` 列で、C はクリーニングカートリッジ、D はデータカートリッジ、P はテープドライブによって消費済み (使用済み) として報告されたクリーニングカートリッジ、および S はスクラッチカートリッジを示します。

カスタマイズしたレポートを作成するには、`-f filename` オプションを使用します。詳細は、「[ロギングボリューム統計レポートの作成](#)」を参照してください。

`$ACS_HOME/data/external/volrpt/owner_id.volrpt` は、テンプレートとして実行または使用することで、カスタマイズされたボリュームレポートを作成できる入力ファイルの例

です。カスタマイズしたボリュームレポートを `$ACS_HOME/data/external/volrpt` ディレクトリに保存することもできます。

ボリュームレポートは標準の UNIX のリダイレクションを使用してファイルにリダイレクトできます。

```
volrpt > file
```

例

デフォルトでは、`volrpt` はリスト内の最初の ACS のみを報告します。ACS 0 および ACS1 の両方のカートリッジを報告するには、次のコマンドを入力します。

```
volrpt -a 0 1
```

LSM 0,1 および 2,1 のカートリッジをホームセル位置でソートして報告するには、次のコマンドを入力します。

```
volrpt -s loc -l 0,1 2,1
```

注:

- `volrpt` は、正常に完了した場合、指定されたボリュームレポートを表示します。`-f` オプションを指定して、`volrpt` が指定されたファイルを見つけられないか、複数の入力ファイルが指定された場合、`volrpt` は `stderr` にメッセージを出力して終了します。入力ファイル内のフィールドのエラーの場合、`volrpt` はメッセージを `stderr` に出力してエラーのある行を無視しますが終了しません。
- カートリッジが指定されたボリューム ID のリスト、範囲、またはライブラリコンポーネント内で見つからない場合、`volrpt` は `no volumes found` メッセージを返します。
- パラメータを指定しない場合は、デフォルトの ACS 0 が使用されます。
- ライブラリコンポーネントが `-a`、`-l`、または `-v` オプションを使用して指定されたがボリュームが見つからない場合は、次のようなメッセージが表示されます。
 - `-a` オプション (ACS)

メッセージ:

```
単一の acs_id が指定され、ボリュームがない場合は、次のエラーが表示されます: No Volumes found for ACS: (<acsid>)
```

例:

```
$ volrpt -a 2
```

No Volumes found for ACS: (2)

複数の *acs_id* が指定され、それらにボリュームがない場合は、次のエラーが表示されます:

No Volumes found for ACS: (<acsid1>(<acsid2>)

例:

```
$ volrpt -a 0 1
No Volumes found for LSM: (0) (1)
```

- *-l* オプション (LSM)

メッセージ:

単一の *lsm_id* が指定され、ボリュームがない場合は、次のエラーが表示されます: *No Volumes found for LSM: (<lsmid>)*

例:

```
$ volrpt -l 1,1
No Volumes found for LSM: (1,1)
```

複数の *lsm_id* が指定され、それらにボリュームがない場合は、次のエラーが表示されます: *No Volumes found for LSM: (<lsmid1>(<lsmid2>)*

例:

```
$ volrpt -l 1,1 1,2
No Volumes found for LSM: (1,1) (1,2)
```

- *-v* オプション (ボリューム)

メッセージ:

単一の *volid* が指定され、ボリュームがない場合は、次のエラーが表示されます: *Volume(s) not: (<volid>)*

例:

```
$ volrpt -v BBB112  
No Volumes found: (BBB112)
```

複数の `valid` が指定され、それらにボリュームがない場合は、次のエラーが表示されま
す: *Volume(s) not found: (<valid1>)(<valid2>)*

例:

```
$ volrpt -v BBB112 BBB114  
No Volumes found: (BBB112) (BBB114)
```

`-v` オプションはボリュームの範囲に使用することもでき、ボリュームがない場合は同様の
メッセージが生成されます。

単一のボリュームの範囲が指定され、ボリュームがない場合は、次のエラーが表示され
ます: *Volume(s) not: (<volrange>)*.

例:

```
$ volrpt -v BBB112-BBB116  
No Volumes found: (BBB112-BBB116)
```

複数のボリュームの範囲が指定され、ボリュームがない場合は、次のエラーが表示され
ます: *Volume(s) not: (<volrange1>) (<volrange2>)*

例:

```
$ volrpt -v BBB112-BBB116 BBB220-BBB224  
No Volumes found: (BBB112-BBB116) (BBB220-BBB224)
```

ACS または LSM が構成されていない場合

`volrpt` が存在しない `acs_id` または `lsm_id` とともに使用された場合は、識別子に従っ
てメッセージが表示されます。

- `-a` (ACS)

ACS 識別子 (<acsid>) は構成されていません

- `-l` (LSM)

LSM 識別子 (<lsmid>) は構成されていません

[「ロギングボリューム統計レポートの作成」](#)を参照してください。

watch_vols

このユーティリティーは、次のボリュームに事前定義されたポリシーを適用します。

- 新しく挿入されました
- 監査またはカートリッジ回復によって検出されました
- 監査、カートリッジ回復、または挿入 (*enter*) によって再アクティブ化されました

これらのポリシーは次のファイルに定義されます。

```
$ACS_HOME/data/external/vol_attr.dat
```

このファイルには、ユーザー定義のボリューム ID またはボリュームの範囲のリスト、および記録されている各ボリュームのユーザー指定のポリシーが含まれています。そのファイルに一覧されている各ボリュームまたはボリュームの範囲について、ボリュームが挿入されたときのボリュームの所有権、プールの関連付け、推奨される LSM の位置、および論理ライブラリの割り当てを定義できます。ポリシーを定義する具体的な手順については、`vol_attr.dat` ファイル内に詳しく説明されています。

`watch_vols` ユーティリティーは、`acsss_stats.log` を使用して、新しく挿入されたボリューム、あるいは監査またはカートリッジの回復によって検出または再アクティブ化されたボリュームの存在を識別します。この機能を有効にするには、`acsss_config` (オプション 3) を使用してボリューム統計情報を有効にする必要があります。ボリュームの統計を有効にすると、`watch_vols` は、`vol_attr.dat` に定義されているエン트리と一致するボリュームを見つけるために `acsss_stats.log` の末尾をモニターします。一致が見つかったら、そのボリュームに対して定義されているポリシーが自動的に適用されます。

ボリューム ID は次の規則に従う必要があります。

- 1 行に 1 つの `vol_id` またはボリュームの範囲。
- `vol_id` は有効な ACSLS ボリューム ID である必要があります。

- `vol_id` の末尾または先頭にスペースが含まれている場合は、それらをアンダースコア (`_`) として表す必要があります。たとえば、「`_V234_`」です。

形式

```
watch_vols [start|stop]
```

使用方法

ユーティリティの実行中のステータスを確認するには、パラメータなしで `watch_vols` を起動します。`watch_vols` のステータス (実行中または停止) が不明な場合は、コマンド `watch_vols` を引数なしで実行すると、現在のステータスが表示されます。

`watch_vols` には 2 つのオプション (`start` および `stop`) があります。

- `watch_vols start`

`start` パラメータを指定して呼び出された場合、`watch_vols` は `vol_attr.dat` に定義されているポリシーを確認します。形式または構文にエラーがある場合、`watch_vols` は、エラーを表示して、`vol_attr.dat` に必要な訂正を行います。定義されたポリシーが `watch_vols` によって受け入れられると、ユーティリティはバックグラウンドで実行されるデーモンを起動します。このデーモンは、`ACSLs` が実行されている場合、継続して実行されます。これは `ACSLs` が再起動されるたびに自動的に起動されます。

`vol_attr.dat` のポリシーテーブルはいつでも更新できます。ポリシーを更新するために `watch_vols` を停止する必要はありません。実行しているプログラムに更新をコミットするには、単に `watch_vols start` を実行します。

- `watch_vols stop`

このコマンドは、指定したボリュームに対する以降のポリシーの適用を停止します。

すべての `watch_vols` アクティビティのログは、次のログファイルに保持されます。

```
$ACS_HOME/log/watch_vols_event.log
```

ボリュームの所有権、`pool_id`、または `LSM` のホーム位置が変更されるたびに、このファイルに記録されます。

例

挿入操作を実行していて、特定のボリュームが挿入されたときにターゲット `LSM` に移動するとします。

1. `watch_vols disabled` を使用してターゲット LSM を監査します。
2. ターゲット LSM を監査したら、`watch_vols start` を実行します。
3. ポリシーによって `vol_attr.dat` に定義されているボリュームが挿入されます。

`watch_vols` は、指定されたボリュームが挿入されたあとに、それらを移動先 LSM に移動します。

第13章 コマンドリファレンス

この章では、一般的なコマンド構文および参照情報を含む ACSLS コマンドの使用方法について学習します。

- `audit`

ライブラリコンポーネント内のボリュームのデータベースインベントリを作成または更新します。

- `cancel`

現在または保留中のリクエストを取り消します。

- `clear lock`

ドライブまたはカートリッジのすべてのアクティブなロックおよび保留中のロックを解除します。

- `define pool`

スクラッチプールを作成または変更します。

- `delete pool`

空のスクラッチプールを削除します。

- `dismount`

カートリッジをマウント解除します。

- `eject`

1つまたは複数のカートリッジを ACS から取り出します。

- `enter`

CAP を挿入モードに設定します。

- `idle`

ACSLS の新しいリクエストの処理を停止します。

- `lock`

ユーザーに対しカートリッジまたはドライブをロック (専用) にします。

- `logoff`

`cmd_proc` を終了します。

- `move`

指定されたカートリッジを指定された LSM 内の使用可能なストレージセルに移動します。

- `mount`

データまたはスクラッチカートリッジをマウントします。

- `query コマンド`

ライブラリコンポーネントのステータスを表示します。

- `set コマンド`

さまざまなライブラリコンポーネントの各種属性を設定します。

- `show`

ロック ID またはユーザー ID を表示します。

- `start`

ACSLS のリクエスト処理を開始します。

- `switch lmu`

ACS の管理を ACS のアクティブ LMU からスタンバイ LMU に手動で切り替えます。

- `unlock`

カートリッジまたはドライブのアクティブなロックを解除します。

- `vary`

ACS、LSM、CAP、ドライブ、またはポートの状態を変更します。

- `venter`

1 つまたは複数のラベルが見つからない、または読み取れないカートリッジを ACS に挿入します。

注:

L5500、SL500、および SL8500 ライブラリはラベルのないカートリッジをサポートしていません (venter)。

一般的なコマンド構文

このセクションでは、ACSL5 コマンドの一般的な構文について説明します。次のセクションでは、構文を含む各コマンドについて詳しく説明します。

ACSL5 コマンドは、次の一般的な構文を使用します。

command type identifier state options

ここでは:

type identifier

は ACS コンポーネントとその識別子です。詳細については、「[コンポーネントのタイプと識別子](#)」を参照してください。

- *state*

は、*vary* コマンドのみに対するデバイス状態です。

- *options*

はコマンドオプションです。実行するコマンドの説明を参照してください。

次の構文ルールに注意してください。

- コマンドは、上記の順序 (コマンド名のあとにコンポーネントとその識別子、状態、および任意のオプション) で入力します。
- この章では、小文字でコマンドを示していますが、小文字と大文字の任意の組み合わせでコマンドを入力できます。
- 下線はコマンドとキーワードの最小の省略名を示しています。たとえば、**query server** と *q ser* はともに *query server* コマンドの有効な形式です。
- 省略記号 (...) は識別子を繰り返せることを示します。
- 大カッコ [] はオプションのオプションを囲んでいます。

コンポーネントのタイプと識別子

次の表では、ACS コンポーネント識別子のそれぞれの有効な値の範囲について説明します。各タイプにつき、最大 42 個の識別子を指定できます。各コマンドの有効なコンポーネントタイプについては、特定のコマンドの説明を参照してください。

注:

次の表に指定された識別子は、ソフトウェアによってサポートされている値の有効な範囲を表していません。LSM のタイプとライブラリ構成によって、特定のサイトに有効な識別子の値が決まります。

表13.1 ACSLS コンポーネントのタイプと識別子

コンポーネント	タイプ	識別子	有効な値
ライブラリ全体	<u>server</u>	なし	なし
ACS	<u>acs</u>	<i>acs_id</i>	acs(0-31)、 lsm(0-99)
LSM	<u>lsm</u>	<i>lsm_id</i>	acs(0-31)、 lsm(0-99)
LSM パネル	<u>panel</u>	<i>panel_id</i>	acs(0-31)、 lsm(0-99)、 panel(0-50)、
LSM サブパネル ¹	<u>subpanel</u>	<i>subpanel_id</i>	acs(0-31)、 lsm(0-99)、 panel(0-50)、 startrow(0-51)、 startcolumn(0-23)、 endrow(0-51)、 endcolumn(0-23)
LSM ストレージセル	<u>subpanel</u>	<i>cell_id</i>	acs(0-31)、 lsm(0-99)、

コンポーネント	タイプ	識別子	有効な値
			panel(0-50)、 row(0-51)、 column(0-23)
CAP	<u>cap</u>	<i>cap_id</i>	acs(0-31)、 lsm(0-99)、 cap(0-11) <i>cap_id</i> 内のアスタリスク (*) は次を行います。 acs,lsm,* - ACSLS は LSM 内で使用可能な CAP のうち、もっとも優先度の高いものを選択します。 acs,* - ACSLS は ACS 内で使用可能な CAP のうち、もっとも優先度の高いものを選択します。 * - 挿入リクエストの場合、ACSLs は LSM 内で空きセルのもっとも多い CAP を選択します。 * — 取り出しリクエストの場合、ACSLs は取り出しに指定されたカートリッジを含む各 ACS 内の CAP のうち、もっとも優先度の高いものを選択します。
ドライブ	<u>drive</u>	<i>drive_id</i>	acs(0-31)、 lsm(0-99)、 panel(0-50)、 drive(0-31)
ドライブタイプ	<u>drive</u>	<i>drive_type</i>	10 文字までのドライブタイプ識別子。数字 (0-9) または文字 (A-Z) の任意の組み合わせにできます。スペースは使用できません。
ポート	<u>port</u>	<i>port_id</i>	acs(0-31)、

コンポーネント	タイプ	識別子	有効な値
			port(0-3)
データまたはスクラッチカートリッジまたはクリーニングカートリッジのボリュームシリアル番号	<u>volume</u> 、 <u>scratch</u> 、 <u>clean</u>	<i>vol_id</i>	<p>6文字の識別子で、数字(0-9)、文字(A-Z)、ドル記号(\$)、シャープ記号(#)、およびスペースの任意の組み合わせで構成されます</p> <p>先頭または末尾にスペースを使用する場合は単一引用符または二重引用符で <i>volser</i> を囲みます。</p> <p>スペースを埋め込んだ <i>volser</i> は指定しないでください。</p>
ボリュームシリアル番号の範囲	<u>volume</u>	<i>volrange</i>	<p>ボリュームの昇順の範囲をダッシュ(-)で区切って指定します</p> <p>ALPHANUM_VOL_RANGES 動的変数が FALSE に設定されている場合 (デフォルト): ACSLS は完全な <i>vol_id</i> として指定された範囲の開始および終了 <i>vol_id</i> をサポートしています (例: AAA000-AAA999)。<i>volser</i> の右端の数値部分のみを範囲として指定します。先行する文字はすべて同じである必要があります。</p> <p>***</p> <p>ALPHANUM_VOL_RANGESab 動的変数が TRUE に設定されている場合: ACSLs は数字(0-9)、文字(A-Z)、ドル記号(\$)、シャープ記号(#)、スペース(先頭と末尾)の任意の組み合わせから構成されるボリューム範囲をサポートしています。***</p> <p>*** 単一引用符または二重引用符を使用して、先頭または末尾のスペースを含む範囲で <i>volser</i> を囲みます。範囲は ACSII 照合順序に従った昇順である必要があります。</p>
ボリュームメディアタイプ	<u>media</u>	<i>media_type</i>	10文字までのメディアタイプ識別子。数字(0-9)、文字(A-Z)、およびダッシュ(-)の任意

コンポーネント	タイプ	識別子	有効な値
			の組み合わせにできます。スペースは使用できません。
ボリュームの所有者	<code>owner</code>	<code>owner_id</code>	
スクラッチプール	<code>pool</code>	<code>pool_id</code>	10 進数 (0-65534)。pool_id にアスタリスク (*) を指定すると、ボリュームがその現在の pool_id に再割り当てされます。
ACSL S リクエスト	<code>request</code>	<code>request_id</code>	ACSL S によって割り当てられた一意の 10 進数 (0-65535)。
ドライブまたはボリュームのロック	<code>lock</code>	<code>lock_id</code>	10 進数 (0-32767)
ACSL S リクエスト	<code>request</code>	<code>request_id</code>	ACSL S によって割り当てられた一意の数値 (0-65535) リクエスト識別子。

¹終了行 (および列) は先頭行 (および列) 以上の大きさがある必要があります。先頭と終了のマトリックス内のセルのみが監査されます。マトリックスは先頭行と先頭列から始まり、終了行と終了列まで拡張されます。

一般的なコマンドエラーメッセージ

一般的なエラーメッセージは、ACSL S が構文エラー、無効な識別子、型、オプション、プロセス障害、データベースエラーなどのため、コマンドを拒否する場合に表示されます。一般的なエラーメッセージの詳細については、ACSL S メッセージを参照してください。

使用できないコマンドを入力すると、次のメッセージが表示されます。

Command access denied.

アクセスできないボリュームを指定した場合、次のメッセージが表示されます。

Volume access denied.

コマンド

次のセクションでは、ACSL S コマンドについて説明します。

audit

audit コマンドは、ライブラリカートリッジの実際のインベントリと一致するように ACSL S データベースを更新します。

次の場合は、常に監査を実行する必要があります。

- これが新しいライブラリである場合。
- ライブラリの内容で ACSLS データベースを再同期するため。
- 1 つまたは複数のモジュールが追加、削除、またはスワップされました。
- ドアからカートリッジを手動で追加または取り外した場合。
- ライブラリ構成設定が変更されたあと。
- ACSLS がライブラリ内の 1 つ以上のカートリッジの場所を認識しない場合。
- ライブラリが展開されていて、それをはじめて監査する場合。

SL8500 が展開されていて、それをはじめて監査するには、[「SL8500 内部アドレスと ACSLS アドレスについて」](#)で説明されている手順を参照してください。

形式

audit cap_id type identifier...

- *cap_id*

ACSLS が障害のあるボリュームを取り出すために使用する CAP を指定します。特定の上限を指定したり、アスタリスク (*) を入力したりでき、これにより、ACSLS にもっとも優先度の高い CAP を選択させます。

サーバーまたは複数の ACS を指定できるのは、各 ACS に CAP の優先度が設定されている場合に限りです。各 ACS 内の CAP の自動選択を許可するには、*cap_id* にアスタリスクを指定する必要があります。

単一の LSM の *audit* は、ACSLS がデータベースを更新するまで待機し、障害のあるボリュームを取り出すために必要な場合に CAP を予約します。

- *type identifier*

ライブラリコンポーネントを指定します。次の表に、監査 (*audit*) できるコンポーネントの一覧を示します。

表13.2 audit に有効なコンポーネント

ライブラリコンポーネント	タイプ	識別子
すべてのライブラリ	<i>server</i>	なし

ライブラリコンポーネント	タイプ	識別子
ACS	<i>acs</i>	<i>acs_id</i>
LSM	<i>lsm</i>	<i>lsm_id</i>
LSM パネル	<i>panel</i>	<i>panel_id</i>
LSM サブパネル	<i>subpanel</i>	<i>subpanel_id</i>

1つの *audit* リクエストで、複数の ACS、LSM、パネル、またはサブパネルを指定できます。重複するサブパネルは指定できません。

サーバーまたは複数の ACS を指定できるのは、各 ACS に CAP の優先度が設定されている場合に限りです。各 ACS 内の CAP の自動選択を許可するには、*cap_id* にアスタリスクを指定する必要があります。これが行われていない場合、監査は見つかった無効なカートリッジを取り出せません。

サーバーまたは ACS を監査する場合、親監査プロセスは LSM ごとに個別の監査プロセスを作成します。これらの LSM 監査は並列で実行し、*panel_id* の昇順でパネルごとに処理します。すべてのライブラリまたは ACS 内のすべての LSM を監査するには、単にサーバーまたは ACS を監査します。複数の個別の LSM の監査を実行する必要はありません。

ただし、LSM 内では、複数のコンポーネントを指定する順序に関係なく、監査は、これらのコンポーネントを *acs_id*、*lsm_id*、*panel_id*、*subpanel_id* によって昇順に処理します。たとえば、サブパネルが最初に指定されている場合でも、監査はサブパネル 0,0,10,1,7 の前にパネル 0,0,9 を処理します。

使用方法

audit は、ライブラリカートリッジの実際のインベントリと一致するように ACSLS データベースを更新します。*audit* は次の目的で実行します。

- CAP 経由でカートリッジを挿入しない場合に、データベースにボリューム情報を作成します。たとえば、LSM をライブラリに追加し、LSM ドアを開き、手動でカートリッジを LSM に追加する場合に *audit* を実行します。
- ライブラリとデータベースの間の不一致を解決します。たとえば、LSM ドアを開き、CAP 経由でカートリッジを取り出す代わりにそれらを手動で取り外す場合に、*audit* を実行します。*audit* は、取り外されたカートリッジの情報をデータベースから削除します。
- *audit* によって指定されたセルの内容を表示します (LSM ロボットの視覚システムに接続されたディスプレイモニターが必要です)。

注:

監査は、クリーニングカートリッジの特定のモデルを認識し、それらをクリーニングカートリッジとしてデータベースに記録します。*audit* コマンドは、検出した新しいクリーニングカートリッジに対し、クリーニングカートリッジのタイプごとに適切に *max_uses* も設定します。

注:

audit が、記録された場所 (セルとおそらくドライブ) でカートリッジを見つけられない場合、*audit* はカートリッジを *absent* としてマークする (不在ボリュームの保持が有効にされている場合) か、またはボリュームを削除します。*audit* はボリュームを削除する際に、ボリューム情報とアクセス制御、スクラッチステータス、ロック ID、およびプール ID に関する顧客提供情報を含むボリュームに関するすべての情報をデータベースから削除します。*audit* があとで別の場所にカートリッジを見つけた場合は、ボリューム情報を再追加しますが、顧客提供情報は失われます。

audit は障害のあるボリュームを取り出し、それらの情報をデータベースから削除します。障害のあるボリュームは次のものを持ちます。

- すでにスキャンされたものと重複する外部ラベル。
- 見つからないまたは読み取れない外部ラベル、および仮想ラベルがない。
- 無効なメディアタイプ。

監査は、ドライブや CAP ではなく、LSM ストレージセルにのみ適用されます。ACSLs は、監査中に *cmd_proc* メッセージを表示し、*audit* からのすべてのデータベースの変更をイベントログに記録します。ボリューム統計のロギングが有効になっている場合、ACSLs は追加のボリュームが見つかったメッセージとボリュームが見つからなかったメッセージを *acsss_stats.log* に記録します。詳細については、「[ロギングボリューム統計レポートの作成](#)」を参照してください。

ヒント: *audit* を実行する場合は次のガイドラインを使用します。

- 監査される ACS または LSM はオンライン状態または診断状態である必要があります。通常のライブラリ処理 (マウントとマウント解除を含む) が *audit* 中に行われる可能性があります。ライブラリ処理によって監査が遅くなります。

オンライン状態と診断状態の違い: 診断状態の *audit* は *cmd_proc* によってのみ実行できます。LSM が診断状態にある間、クライアントリクエストは拒否されます。この状態は、クライアントから送信される可能性のある *mount/dismount* リクエストと競合せず、*audit* がライブラリを排他的に実行するため、速くなります。

- LSM 全体に対して *audit* を開始すると、同じ LSM に対して別の *audit* を開始 (*start*) できません。取り消し (*cancel*) て、監査を再実行する必要があります。

- 監査リクエストは取り消し (*cancel*) ですが、ACSLs は常に現在のパネルまたはサブパネルの監査を終了します。監査を取り消す (*cancel*) と、取り出し用にマークされたカートリッジの一部またはすべてが取り出されないことがあります。監査を取り消し (*cancel*) た場合、すでに取り出されたカートリッジがふたたび挿入されることはありません。

注意:

監査を取り消し (*cancel*) た場合、または監査中に ACSLS ハードウェアまたはソフトウェアの障害が発生した場合は、同じ監査を再実行する必要があります。取り出し用にマークされていたが、最初の監査中に実際に取り出されなかったカートリッジは、データベースに存在しなくなり、ACSLs の制御下になくなります。

例

- ライブラリ全体を監査して、各 ACS 内で取り出し用にもっとも優先度が高い CAP を指定するには:

```
audit * server
```

- LSM 0,1 を監査 (*audit*) して、取り出し用に CAP 0,1,1 を指定するには:

```
audit 0,1,1 lsm 0,x
```

- LSM 0,1 のパネル 10 を監査 (*audit*) して、LSM 0,1 内で取り出し用にもっとも優先度が高い CAP を指定するには:

```
audit 0,1,* panel 0,1,10
```

注:

セルが別のプロセスによって予約されている場合は監査できません。セルが予約されている場合、ACSLs は最大 60 回の再試行で、セルが使用可能になるまでデータベースを再チェックします。セルがまだ使用できない場合、*audit* は、セルをスキップし、イベントログにメッセージを記録します。

関連項目:

情報の内容	参照先
ライブラリの監査のガイドライン	「ACS 番号の指定」
コマンドの取り消し	「cancel」
CAP ステータスの表示	「query cap」
ACSLs およびライブラリのステータスの表示	「query server」
ACS ステータスの表示	「query acs」
LSM ステータスの表示	「query lsm」

情報の内容	参照先
CAP 選択の優先度の設定	「set cap priority」
CAP モード (手動または自動) の設定	「set cap mode」
ライブラリコンポーネントの状態の変更	「vary」

コマンド領域メッセージ

次のセクションでは、監査メッセージについて説明します。

成功メッセージ

- *audit* が正常に完了すると、次のメッセージが表示されます。

Audit: Audit completed, Success.

- さらに、監査されたコンポーネントを確認する次のいずれかのメッセージが表示されます。

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

ヒント: 無効なセルの場所 (つまり、ドライブのすぐ上かすぐ下、または追加の列上) を監査 (*audit*) すると、ACSLS は成功メッセージを返し、監査が PTP と連携できるようにします。

中間メッセージ

中間メッセージは次を最初の行とする 2 行表示で構成されます。

Audit: Intermediate response: Audit activity.

2 行目には次のいずれかのメッセージが表示されます。

- *Audit: Volume ejected, unreadable label.*

説明: ACSLS は次を持つカートリッジを取り出しました。

外部ラベルなし

仮想ラベルなし、または

読み取れないラベル

- *Audit: Volume vol_id will be ejected, duplicate label.*

説明: ACSLS は監査対象のセルの範囲内の重複した外部ラベルによって取り出されました。

変数: *vol_id* は、重複するラベルを持つボリュームです。

- *Audit: Volume vol_id found.*

説明: 監査は ACSLS データベースにない ACS にボリュームを見つけました。監査はボリュームをデータベースに追加しました。

変数: *vol_id* はデータベースに追加されたボリュームです。

- *Audit: Volume vol_id not found.*

説明: ACSLS データベースに一覧表示されているボリュームが ACS にありません。ボリュームはデータベースから削除されました。

変数: *vol_id* はデータベースから削除されたボリュームです。

- *Audit: Volume will be ejected, invalid media type*

説明: ACSLS は無効なメディアタイプのボリュームを取り出しました。

エラーメッセージ

- *Audit in progress.*

説明: 同じ LSM に対して別の監査が進行中のため、ACSLS は *audit* を開始しませんでした。

- *CAP cap_id in use.*

説明: 監査に指定された CAP は使用中です。

変数: *cap_id* は使用中の CAP です。

- *Multiple ACS audit.*

説明: *audit* コマンドが *cap_id* に * を指定せずに複数の ACS を指定したため、監査が失敗しました。

- *Not in same ACS.*

説明: 指定された *cap_id* と識別子が同じ ACS 内にないため、*audit* が失敗しました。

変数:

cap_id は監査に指定された CAP です。

識別子は、監査に指定されたライブラリコンポーネントです。

表示領域メッセージ

- *cap_id Remove cartridges from CAP.*

説明: 監査は取り出し済みカートリッジで CAP をいっぱいにしました。CAP を空にしてから、それを閉じて監査を続行します。

変数: *cap_id* は取り出し済みカートリッジを含む CAP です。

- *CAP cap_id: Place magazines in CAP.*

説明: CAP は監査にマガジンが必要です。CAP を開き、マガジンの中に入れて、CAP を閉じます。

変数: *cap_id* マガジンを必要とする CAP です。

- *CAP cap_id: No CAP available, waiting...*

説明: カートリッジの取り出しに使用できる CAP がありません。

変数: *audit* コマンドで指定されたとおりに *cap_id* が表示されます。

- *audit* コマンドで CAP を明示的に指定していた場合 *acs, lsm, cap*。
- *audit* コマンドで CAP を *acs, lsm, ** と指定した場合、*acs, lsm, **。
- *audit* コマンドで CAP を *acs, ** または *** と指定した場合、*acs, *, **。

cancel

cancel コマンドは、現在または保留中のリクエストを取り消します。

形式

cancel request_id

オプション

- *request_id*

取り消すリクエストの識別子を指定します。

cancel コマンドは、*audit*、*define pool*、*delete pool*、*eject*、*enter*、*lock*、*query*、*set*、または *venter* コマンドまたはクライアントアプリケーションによって発行された現在または保留中のリクエストを取り消すために使用します。取り消すリクエストの ID を表示するには、*query request* コマンドを使用します。

サーバー、ACS、または LSM の監査を取り消し (*cancel*) できます。ACSLs は内部でサーバー、ACS、または LSM 監査を一連のパネル監査に変換するため、ACSLs は残りの監査を取り消す前に、現在のパネルの *audit* を完了します。パネルまたはサブパネルの監査は取り消し (*cancel*) できません。監査を取り消し (*cancel*) た場合、すでに取り出されたカートリッジがふたたび挿入されることはありません。

注意:

監査を取り消した場合、または監査中にライブラリまたは ACSLS ハードウェアまたはソフトウェアの障害が発生した場合は、同じ監査を再実行する必要があります。取り出し用にマークされていたが、最初の監査中に実際に取り出されなかったカートリッジは、データベースに存在なくなり、ACSLs の制御下になります。

ヒント: 取り消すリクエストを発行した *cmd_proc* と異なる *cmd_proc* から *cancel* コマンドを入力します。

cancel コマンドはすべての保留中のリクエストを取り消して、次のように現在のリクエストを処理します。

- *audit*

ACSLs は内部でサーバー、ACS、または LSM 監査を一連のパネル監査に変換するため、ACSLs は残りの監査を取り消す前に、現在のパネルの監査を完了します。

注:

監査を取り消し (*cancel*) た場合、または監査中に ACSLS ハードウェアまたはソフトウェアの障害が発生した場合は、同じ監査を再実行する必要があります。取り出し用にマークされていたが、最初の監査中に実際に取り出されなかったカートリッジは、データベースに存在なくなり、ACSLs の制御下になります。

- *define pool*

ACSLS はスクラッチプールの定義を停止しますが、すでに定義されているスクラッチプールは削除しません。

- *delete pool*

ACSLS はスクラッチプールの削除を停止しますが、すでに削除されているスクラッチプールは再定義しません。

- *eject*

ACSLS は取り出しを停止し、*cmd_proc* はすでに取り出されたカートリッジを取り外すメッセージ表示し、それらはふたたび挿入されません。

取り出しはカートリッジが取り外され、CAP が閉じ、ACSLS が CAP が空であることを確認するまで終了しません。

- *enter*

ACSLS は *enter* を停止し、いずれかのカートリッジが CAP 内に残っている場合、*cmd_proc* がこれらのカートリッジを取り外すメッセージを表示します。LSM にすでに挿入されたカートリッジは取り出されません。

enter は、カートリッジが取り外され、CAP が閉じて、ACSLS が CAP が空であることを確認するまで終了しません。

注:

Automatic enter.cancel コマンドを使用して、進行中の自動挿入操作をキャンセル (*cancel*) することはできません。進行中の自動挿入を終了するには:

- CAP のドアが開いている場合、すべてのカートリッジを取り外して、ドアを閉じます。
- CAP のドアが閉じ、カートリッジがライブラリに移動されると、残りのカートリッジをライブラリに挿入できます。その後 *enter* が終了します。

- *lock*

指定されたリクエストによってリソースロックが停止されます。リクエストでまだ指定されたすべてのリソースを取得していない場合、どのリソースもロックされません。

- *query*

ACSLS はクエリーを取り消します。

- *set*

`set cap` の場合、ACSL S は CAP 属性の設定を停止します。ただし、すでに設定されている属性は変更しません。

`set scratch` または `set clean` リクエストの場合、ACSL S はスクラッチカートリッジまたはクリーニングカートリッジの属性の設定を停止しますが、すでに設定されている属性は変更しません。

- `venter`

ACSL S は挿入を停止 (`stops`) し、いずれかのカートリッジが CAP に残っている場合、`cmd_proc` はこれらのカートリッジを取り外すメッセージを表示します。LSM にすでに挿入されたカートリッジは取り出されません。

例

- 現在および保留中のすべてのリクエスト ID を表示するには:

```
query request all
```

`query request all` の出力の例:

<i>Identifier</i>	<i>Command</i>	<i>Status</i>
13	<code>enter</code>	<i>Current</i>
15	<code>query</code>	<i>Pending</i>

- 上の例のリクエスト 13 (現在の挿入リクエスト) を取り消すには:

```
cancel 13
```

関連項目:

情報の内容	参照先
ライブラリカートリッジの実際のインベントリに一致させるための ACSLS データベースの更新	「audit」
スクラッチプールの作成または変更	「define pool」
空のスクラッチプールの削除	「delete pool」
ライブラリからのカートリッジの取り出し	「eject」
ラベル付きカートリッジをライブラリに挿入するための CAP (手動モード) の準備	「enter」
現在のロック ID へのドライブやカートリッジのロック (専用化)	「lock」

情報の内容	参照先
ライブラリコンポーネントのステータスの表示	「 query コマンド 」
さまざまなライブラリコンポーネントの各種属性の設定	「 set コマンド 」
ラベルのないカートリッジをライブラリに挿入するための CAP の準備	「 venter 」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- *Request request_id canceled.*

説明: ACSLS がリクエストされたコマンドを取り消しました。

変数: *request_id* は取り消されたコマンドのリクエスト識別子です。

中間メッセージ

なし。

エラーメッセージ

- *Request request_id can not be canceled: status.*

説明: ACSLS は指定されたコマンドを取り消しできません。

変数:

- *request_id* は ACSLS が取り消しできないコマンドのリクエスト識別子です。
- *status* は、次のいずれかになります。

- *Request identifier request_id invalid.*

cancel コマンドは無効なリクエスト識別子を指定しました。

- *Request identifier request_id not found.*

cancel コマンドは、現在または保留中でないリクエストのリクエスト識別子を指定しました。

表示領域メッセージ

なし。

clear lock

clear lock コマンドは、指定されたドライブやカートリッジへのすべてのアクティブおよび保留中のロックを解除します。

形式

clear lock type identifier

オプション

type identifier

ライブラリコンポーネントを指定します。次の表に、リソースロックをクリアできるコンポーネントの一覧を示します。

表13.3 clear lock に有効なコンポーネント

ライブラリコンポーネント	タイプ	識別子
ドライブ	ドライブ	drive_id
ボリューム	ボリューム	vol_id

使用方法

clear lock コマンドは、指定されたドライブやカートリッジへのすべてのアクティブおよび保留中のロックを解除するために使用します。現在のロック ID は 0 か、またはドライブやカートリッジのロック ID に一致する必要があります。

unlock コマンドは、ドライブまたはカートリッジへのアクティブなロックのみを解除します。ただし、*unlock* コマンドを使用して、すべてのドライブまたはすべてのカートリッジへのアクティブなロックを解除できます。

注:

clear lock コマンドはロック ID を常に 0 にリセットします。

例

- ドライブ 1,1,5,2 のすべてのロックをクリアするには:

```
clear lock drive 1,1,5,2
```

- ボリューム NN0108 のすべてのロックをクリアするには:

`clear lock volume NN0108`

注:

`clear lock` コマンドは取り消し (`cancel`) できません。

関連項目:

情報の内容	参照先
ドライブとカートリッジのロック	「lock」
ドライブまたはカートリッジのロックステータスの表示	「query lock」
ロック ID の設定	「set lock」
ロック ID またはユーザー ID の表示	「show」
ドライブまたはカートリッジのアクティブなロックの解除	「unlock」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- `clear` リクエストが成功すると、次のメッセージが表示されます。

Clear: Completed, Success.

- さらに、リクエスト内の識別子ごとに、ライブラリのコンポーネント (タイプ) に応じて、次のいずれかのメッセージが表示されます。
 - *Clear: Drive drive_id all locks cleared.*
 - *Clear: Volume vol_id all locks cleared.*

中間メッセージ

なし。

エラーメッセージ

- *Clear: Clear lock of drive drive_id failed,*

Drive identifier drive_id available.

説明: ACSLS は指定されたドライブがロックされていないため、ロックをクリアできません。

変数: `drive_id` は指定されたドライブの識別子です。

- *Clear: Clear lock of volume vol_id failed,
Volume identifier vol_id available.*

説明: ACSLS は指定されたボリュームがロックされていないため、ロックをクリアできません。

変数: *vol_id* は指定されたボリュームの識別子です。

表示領域メッセージ

なし。

define pool

define pool コマンドはスクラッチプールを作成または変更します。

形式

```
define pool low_water_mark high_water_mark pool_id...[overflow]
```

オプション

- *low_water_mark*

ボリューム警告の下限しきい値です。スクラッチカートリッジカウントがこのしきい値を下回る場合、ACSLs はイベントログに警告メッセージを記録します。有効な値は 0 から $2^{31}-1$ です。デフォルトは 0 です。

- *high_water_mark*

ボリューム警告の上限しきい値です。スクラッチカートリッジカウントがこのしきい値に達するか超える場合、ACSLs は警告メッセージをイベントログに記録します。この値は *low_water_mark* の値よりも大きい必要があります。

- *pool_id*

プール識別子を指定します。プール 0 は共通スクラッチプールで、常に存在します。共通スクラッチプールの属性は変更できます。

- *overflow*

このプールが *mount scratch* リクエストを満たせない場合、ACSLs が共通プール (プール 0) からカートリッジを選択するように指定します。

使用方法

`define pool` コマンドは、スクラッチプールを作成または変更するために使用します。

例

- 新しいプール 1 に、0 の下限しきい値、600 の上限しきい値、およびオーバーフローを定義するには:

```
define pool 0 600 1 overflow
```

- 既存のプール 5 に、0 の下限しきい値、600 の上限しきい値、およびオーバーフローなしを定義するには:

```
define pool 0 600 5
```

注:

スクラッチプールは 1 つのクライアントアプリケーションまたはユーザー ID によって所有されません。ただし、ボリュームアクセス制御を使用して特定のスクラッチカートリッジへのアクセスを制限できます。

関連項目:

情報の内容	参照先
コマンドの取り消し	「cancel」
空のスクラッチプールの削除	「delete pool」
スクラッチカートリッジの管理のガイドラインと手順	「LSM の設定」
ドライブへのスクラッチカートリッジのマウント	「mount *」
スクラッチプール属性の表示	「query pool」
カートリッジスクラッチ属性の設定またはクリア	「set scratch」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- `define pool` コマンドが成功した場合、次のメッセージが表示されます。

```
Define: Define completed, Success.
```

- 作成または変更されたプールごとに次のメッセージが表示されます。

```
Define: Pool pool_id created.
```

中間メッセージ

なし。

エラーメッセージ

なし。

表示領域メッセージ

- *Pool pool_id: low water mark warning.*

説明: 指定されたスクラッチプール内のボリュームの数がボリュームの下限しきい値以下です。

変数: *low_water_mark* は指定されたスクラッチプールのボリュームの下限しきい値です。

- *Pool pool_id: high water mark warning.*

説明: 指定されたスクラッチプール内のカートリッジの数はボリュームの上限しきい値以上です。

変数: *high_water_mark* は指定されたスクラッチプールのボリュームの上限しきい値です。

delete pool

delete pool コマンドは、空のスクラッチプールを削除します。

形式

```
delete pool pool_id... |all
```

オプション

- *pool_id*

プール ID を指定します。

プール 0 は共通プールで、削除できません。

- *all*

すべての空のスクラッチプールを指定します。

使用方法

`delete pool` コマンドは、空のスクラッチプールを削除するために使用します。プールにスクラッチカートリッジが含まれている場合、最初のプールを削除する前に、これらのカートリッジを別のプールに再割り当てする必要があります。スクラッチカートリッジがマウントされている場合、それはデータカートリッジになりますが、そのスクラッチプール内に残ります。`set scratch off` コマンドは、データカートリッジを共通プールに再割り当てするために使用します。

例

- すべての空のスクラッチプールを削除するには:

```
delete pool all
```

空のプールのみが削除されます。カートリッジが割り当てられたプールは影響を受けません。

- スクラッチプール 1 を削除するには、次の手順に従います。

- スクラッチプール 1 をクエリーします。

```
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
1)	34815>	0,0,1,8,1>	home>	3480

プール 1 には、34813 と 34815 の 2 つのカートリッジがあります。

- カートリッジをプール 5 に再割り当てして、プール 1 を空にします。

```
set scratch 5 348013 348015
```

- プール 1 を削除します。

```
delete pool 1
```

関連項目:

情報の内容	参照先
コマンドの取り消し	「cancel」
スクラッチプールの作成または変更	「define pool」

情報の内容	参照先
スクラッチカートリッジの管理のガイドラインと手順	「LSM の設定」
スクラッチプール属性の表示	「query pool」
カートリッジの場所とメディアタイプの表示	「query volume」
カートリッジスクラッチ属性の設定またはクリア	「set scratch」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- *delete pool* コマンドが正常に完了すると、次のメッセージが表示されます。

Delete: Delete completed, Success.

- 削除されたプールごとに次のメッセージが表示されます。

Delete: Pool pool_id deleted.

中間メッセージ

なし。

エラーメッセージ

- *Delete: Pool pool_id failed, Pool not empty.*

説明: ACSLS は、スクラッチプールが空でないためそれを削除できません。

変数: *pool_id* はリクエストされたプールの識別子です。

dismount

dismount コマンドはドライブからカートリッジをマウント解除します。

形式

```
dismount vol_id drive_id [force]
```

オプション

- *vol_id*

カートリッジを指定します。

- *drive_id*

ドライブを指定します。

- *force*

指定されたドライブ内のカートリッジの *vol_id* が、指定された *vol_id* と一致しない場合でも、そのドライブ内の実際のボリュームを強制的にマウント解除します。

このオプションは、ドライブがアンロードされていない場合でも、強制的にマウント解除します。

使用方法

dismount コマンドは、ドライブからボリュームをマウント解除し、使用可能なストレージセルにカートリッジを配置するために使用します。

dismount

指定されたドライブから指定されたカートリッジをマウント解除するには、*force* オプションを付けずに *dismount* コマンドを使用します。強制されないマウント解除の成功には次が必要です。

- ドライブはオンラインである必要があります
- ドライブ内のカートリッジの *vol_id* が *dismount* コマンドに指定された *vol_id* と一致している必要があります
- ドライブはアンロードされている必要があります

注意:

通常のマウント解除では、SL500 および SL150 ライブラリが自動的に巻き戻され、ドライブにロードされているのが見つかったカートリッジがアンロードされます。カートリッジの巻き戻しとアンロードのために、*dismount force* は必要ありません。これらのライブラリに *dismount* を発行する前に、ACSLS クライアントアプリケーションがテープドライブから読み取っていたり、テープドライブに書き込んでいたりしていないことを確認してください。

dismount force

指定されたドライブにマウントされている実際のカートリッジを強制的にマウント解除するには、*force* オプションを付けて *dismount* コマンドを使用します。ドライブ内のカートリッジの *vol_id* は、*dismount* コマンドで指定する *vol_id* と一致している必要はありません。さらに、カートリッジがマウント解除の準備ができていない場合、ACSLS は強制的にドライブにカー

トリッジの巻き戻し、アンロード、マウント解除を自動で行わせます。ドライブはオンラインである必要があります。

dismount force は、応答を受信していないマウントまたはマウント解除操作によって予約済みとマークされたままになっていた場合でも続行します。予約されているドライブは *cmd_proc* または ACSAPI クライアントに使用中として報告されます。

読み取れないか不明なラベルのあるカートリッジまたはクライアントアプリケーションがマウント解除していないカートリッジをマウント解除するために、*force* オプションを使用できます。ACSL S は、ラベルが読み取れないか見つからない場合でも、使用可能なストレージセルにカートリッジを戻すことに注意してください。

注:

ライブラリがカートリッジを巻き戻してアップロードする前に、ドライブと現在ドライブを使用しているアプリケーション間のすべての読み取り/書き込みアクティビティが停止または一時停止される必要があります。読み取り/書き込みアクティビティが中断されずに続行している場合、ライブラリの待機タイムアウトを超えたあとに *dismount force* が失敗します。

この考慮事項は、データパスによって、ホストからドライブにリクエストされたメディア検証にも適用されます。メディア検証が完了するか、またはホストがデータパス経由でそれを停止するまで、ACSL S からの *dismount force* は失敗します。

例

- ドライブ 0,1,10,2 からカートリッジ EDU200 をマウント解除するには:

```
dismount EDU200 0, 1, 10, 2
```

関連項目:

情報の内容	参照先
コマンドの取り消し	「cancel」
ライブラリからのカートリッジの取り出し	「eject」
ラベル付きカートリッジをライブラリに挿入するための CAP (手動モード) の準備	「enter」
ドライブへのスクラッチカートリッジのマウント	「mount *」
ドライブ上のデータカートリッジのマウント	「mount」
CAP ステータスの表示	「query cap」
ドライブステータスの表示	「query drive」
CAP モード (手動または自動) の設定	「set cap mode」

情報の内容	参照先
CAP 選択の優先度の設定	「set cap priority」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- *Dismount: vol_id dismounted from drive_id.*

説明: 強制的なマウント解除が成功しました。

変数:

- *vol_id* はマウント解除されたカートリッジの識別子です。

表示された *vol_id* は、実際にマウント解除されたボリュームであり、必ずしも *dismount force* コマンドで指定されたボリュームではありません。

- *drive_id* は指定されたドライブの識別子です。

- *Dismount: Forced dismount of vol_id from drive_id.*

説明: 強制的なマウント解除が成功しました。

変数:

- *vol_id* はマウント解除されたカートリッジの識別子です。

表示された *vol_id* は、実際にマウント解除されたボリュームであり、必ずしも *dismount force* コマンドで指定されたボリュームではありません。

- *drive_id* は指定されたドライブの識別子です。

中間メッセージ

なし。

エラーメッセージ

- *Dismount: Dismount failed, ACS acs_id full.*

説明: ACS に空きストレージセルがないため、ACSL はカートリッジをマウント解除できません。ACSL がドライブからカートリッジをマウント解除できるように、1 つ以上のほかのカートリッジを ACS から取り出します。

変数: *acs_id* は、カートリッジが格納されている ACS の識別子です。

注:

空きセルの検索ではオンライン LSM のみが使用されます。そのため、このメッセージは、LSM に空きストレージセルがあり、その LSM がオフラインの場合にも発生することがあります。

- *Dismount: Dismount failed, Audit in progress.*

説明: 進行中の監査によって ACS の最後の空きセルの場所へのアクセスがロックされているため、ACSL S はカートリッジをマウント解除できません。

- *Dismount: Dismount failed, Cartridge in drive drive_id, unreadable label*

説明: カートリッジに外部ラベルがないか、読み取れない外部ラベルか、または仮想ラベルがないため、ACSL S はカートリッジをマウント解除できません。カートリッジをマウント解除するには、強制的なマウント解除を使用します。

変数: *drive_id* は指定されたドライブの識別子です。

- *Dismount: Dismount failed, Drive identifier drive_id available.*

説明: 指定されたドライブにマウントされたカートリッジがありません。

変数: *drive_id* は指定されたドライブの識別子です。

- *Dismount: Dismount failed, Drive identifier drive_id in use.*

説明: カートリッジが巻き戻しおよびアンロードされていないため、ACSL S はカートリッジをマウント解除できません。クライアントアプリケーションがカートリッジを巻き戻してアンロードするまで待つてから、*dismount* コマンドを入力するか、または *forced dismount* を使用してカートリッジをマウント解除します。

変数: *drive_id* は指定されたドライブの識別子です。

- *Dismount: Dismount failed, Misplaced tape.*

説明: カートリッジの外部ラベルが ACSLS データベース内のカートリッジ識別子と一致しないため、ACSL S はカートリッジをマウント解除できません。ACSL S データベース内のカートリッジの識別子が、カートリッジの外部ラベルと一致するように更新されます。マウント解除を再試行します。

- *Dismount: Dismount failed, Cartridge not in drive.*

説明: カートリッジの外部ラベルが、マウント解除に指定されたカートリッジの識別子と一致しないため、ACSL5 はカートリッジをマウント解除できません。正しいカートリッジ識別子で *dismount* コマンドを再入力します。

表示領域メッセージ

なし。

eject

eject コマンドは、ロボットに LSM の内部からカートリッジを取得し、それらをオペレータによって取り外しできる CAP に入れるように指示します。

lsm_id オプションを選択すると、単一の LSM で複数の CAP を使用してカートリッジを取り出せます。

形式

```
eject cap_id|lsm_id [opmsg opmsg_nbr] vol_id|volrange...
```

オプション

- *cap_id*

カートリッジの取り出しに使用される CAP を指定します。

- *lsm_id*

lsm_id オプションを選択すると、単一の LSM で使用可能な複数の CAP を使用してカートリッジを取り出せます。要件 (手動、ゼロ以外の優先度、使用可能など) を満たすすべての CAP はカートリッジの取り出しのためにロック解除されます。選択した CAP のいずれかまたはすべてから、任意の順序でカートリッジを取り出すことができます。*eject* は最初、最高の優先度の順番で、カートリッジを CAP に詰め込みます。

例: 優先度 2 のものが 1 つと優先度 5 のものが 1 つの 2 つの CAP があります。*eject* は最初に優先度 5 のステータスの CAP、次に優先度 2 のステータスの CAP に詰め込みます。1 つの CAP に取り出すだけで十分なカートリッジしかない場合は、優先度 5 の CAP に詰め込まれます。

- *opmsg opmsg_nbr*

`cmd_proc` を使用して入力された SL8500 Bulk CAP 取り出しコマンドに、カスタムオペレータパネルメッセージ番号を指定できます。取り出されるカートリッジを取り外せるように、Bulk CAP がロックされていない場合、メッセージが表示されます。

有効なメッセージ番号は 4 から 99 です。

- 現在、カスタム `opmsg` は SL8500 ライブラリの Bulk CAP に対してのみ表示されます。
- `opmsg` パラメータはオプションです。指定されていない場合は、カートリッジを取り外すデフォルトのメッセージが送信されます。
- カスタムの `opmsg` 番号は、ACSAPI クライアント、ACSL S GUI、または `lib_cmd eject` からの取り出しには指定できません。これらの場合は、デフォルトのメッセージが表示されます。

`opmsg` 番号に対して表示されるメッセージを作成するには、SL コンソールを使用して、次を選択します。

Tools

Configuration

CAP Usage Message

`opmsg` を表示する SL Console の CAP ステータス表示を表示するには、次を選択します。

Tools

System Detail

CAP Folder

Status

SL Console に、取り出し中に ACSLS から送信されたメッセージ番号に基づいて Bulk CAP メッセージが表示されます。取り出すカートリッジを取り外すために CAP がロック解除されたあとに、「*System Details CAP Status*」ページにメッセージが表示されます。

例: Bulk CAP 1,2,1 からカートリッジを取り出すために、カスタムオペレータパネルメッセージ番号 55 を指定するには:

```
eject 1,2,1 opmsg 55 T10001 T10033-T10067
```

- `vol_id | volrange`

取り出すカートリッジまたはカートリッジの範囲の外部または仮想ラベルタイプを指定します。

使用方法

eject コマンドは、ライブラリからカートリッジを取り出すために使用します。これにより、それらが ACSLS の制御から解除されます。ロボットは指定された CAP に指定されたカートリッジを配置し、次に ACSLS はカートリッジが格納されていたセルの場所を解放します。カートリッジ情報は、*ABSENT_VOLUME_RETENTION_PERIOD* 変数にゼロ以外の値がある場合に保持され、*ABSENT_VOLUME_RETENTION_PERIOD* が 0 に設定されている場合に削除されます。*eject* コマンドで、カートリッジがいくつかの複数の CAP を指定した場合、CAP がいくつかになったらそれを空にし、CAP を閉じて、すべてのカートリッジが取り出されるまで取り出しを続行します。

単一の *eject* コマンドを使用して、スペースで区切られた複数のカートリッジ ID を指定して、範囲内にない複数のカートリッジを取り出すことができます。

ライブラリに 2 台の LSM が接続されていて、PTP が停止した場合、カートリッジを正常に取り出すには、次のいずれかを実行します。

- カートリッジが格納されている LSM 内の CAP を指定します。たとえば、カートリッジ NN0100 が LSM 0,0 に格納されている場合、CAP0,0,0 からこのカートリッジを取り出すには:

```
eject 0,0,0 NN0101
```

- *cap_id* には、カートリッジが格納されている LSM を指定しますが、CAP 番号にはワイルドカード (*) を使用します。ACSLs は LSM 内の最高の優先度の CAP を選択します。たとえば、カートリッジ NN0114 が LSM 0,0 に格納されている場合、LSM 0,0 の最高の優先度の CAP からこのカートリッジを取り出すには:

```
eject 0,0,* NN0114
```

例

- CAP 0,0,0 からカートリッジ NN0101 を取り出すには:

```
eject 0,0,0 NN0101
```

- ACS 0 上の最高の優先度の CAP からカートリッジ範囲 NN0101-NN0109 を取り出すには:

```
eject 0, * NN0101-NN0109
```

- ACS 0 上の最高の優先度 CAP からカートリッジ NN0101、NN0103、NN0105、および NN0107 を取り出すには:

```
eject 0, * NN0101 NN0103 NN0105 NN0107
```

- LSM 1,2 内のカートリッジを取り出すために、複数のゼロ以外の優先度の CAP を使用するには:

```
eject 1,2 RB1000-RB2000
```

注:

CAP から取り出し済みのすべてのカートリッジを取り外していることを確認します。CAP を別の操作 (挿入や監査など) に使えるようにする前に、取り出し済みのすべてのカートリッジをアンロードし、CAP のドアを閉じて、取り出しを完了する必要があります。

関連項目:

情報の内容	参照先
コマンドの取り消し	「cancel」
ドライブからのカートリッジのマウント解除	「dismount」
カートリッジの取り出しのガイドラインと手順	「LSM の設定」
ラベル付きカートリッジをライブラリに挿入するための CAP (手動モード) の準備	「enter」
CAP ステータスの表示	「query cap」
ドライブステータスの表示	「query drive」
カートリッジの場所とメディアタイプの表示	「query volume」
CAP モード (手動または自動) の設定	「set cap mode」
CAP 選択の優先度の設定	「set cap priority」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- *Eject: Eject complete, a cartridges ejected*
- *Eject: vol_id ejected from cap_id*

説明: ACSLS は指定されたカートリッジを取り出しました。

変数:

- *nn* は取り出し済みカートリッジの数です。
- *vol_id* は取り出し済みカートリッジのカートリッジ識別子です。
- *cap_id* は取り出し済みカートリッジが格納されている CAP です。

中間メッセージ

なし

エラーメッセージ

- *Eject: vol_id Eject failed, CAP cap_id full.*

説明: CAP がいっぱいなため、カートリッジが取り出されませんでした。

変数:

- *vol_id* は取り出されなかったカートリッジのカートリッジ識別子です。
- *cap_id* は取り出しに指定された CAP です。
- *Eject: vol_id Eject failed, CAP cap_id in use.*

説明: CAP が使用中であるため、カートリッジが取り出されませんでした。

変数:

- *vol_id* は取り出されなかったカートリッジのカートリッジ識別子です。
- *cap_id* は取り出しに指定された CAP です。
- *Eject: vol_id Eject failed, Misplaced tape.*

説明: カートリッジの外部ラベルがストレージセルのデータベースのカートリッジ識別子と一致しないため、ACSLs はカートリッジを取り出せません。データベースは、ストレージセル内のカートリッジのカートリッジ識別子、メディアタイプ、およびクリーニングカートリッジ属性を更新します。

変数: *vol_id* は取り出されなかったカートリッジのカートリッジ識別子です。

- *Eject: vol_id Eject failed, Not in same ACS.*

説明: カートリッジは *cap_id* に指定された ACS がないため、取り出されませんでした。

変数: *vol_id* は取り出されなかったカートリッジのカートリッジ識別子です。

- *Eject: vol_id Eject failed, Volume identifier vol_id not found.*

説明: カートリッジは次の状況にないため、取り出されませんでした。

- データベース内の指定されたストレージセル内
- 移動中、または
- ドライブ内。
- *ACSLs deletes the volume entry from the database.*

変数: *vol_id* は、データベースから削除されたカートリッジ識別子です。

- *Eject: vol_id Eject failed, Cartridge in drive.*

説明: カートリッジはドライブにマウントされているため、取り出されませんでした。

変数: *vol_id* は取り出されなかったカートリッジのカートリッジ識別子です。

- *Eject: vol_id Eject failed, Volume vol_id in use.*

説明: カートリッジは、別のリクエストで使用するために予約されているため、取り出されませんでした。

変数: *vol_id* は取り出されなかったカートリッジのカートリッジ識別子です。

表示領域メッセージ

- *CAP cap_id Remove cartridges from CAP.*

説明: CAP がいっぱいになったか、またはリクエストされたすべてのカートリッジが CAP 内にあります。CAP を空にします。

変数: *cap_id* は取り出し済みカートリッジを格納する CAP です。

- *CAP cap_id Place magazines in CAP.*

説明: CAP は取り出し用のマガジンが必要です。CAP を開き、マガジンの中に入れて、CAP を閉じます。

変数: *cap_id* はマガジンを必要とする CAP です。

enter

このコマンドを使用すると、CAP を手動モードまたは自動モードで動作するように設定できます。CAP を自動モードまたは手動モードに設定する手順については、「[カートリッジの挿入](#)」セクションを参照してください。

- 自動モード

CAP が自動モードの場合、enter コマンドを発行せずに挿入操作を開始できます。これを行うには、CAP ドアを開き、1 つまたは複数のカートリッジを中に入れ、CAP を閉じます。挿入の処理中、CAP はロックされます。挿入操作が競合すると、CAP はロック解除されません。

- 手動モード

手動モードの場合、CAP はロックされており、CAP を開いてカートリッジを挿入する前に、次のコマンドを発行することによってのみ使用できます。

```
enter cap_id [opmsg opmsg_nbr]
```

オプション

- *cap_id*

CAP を指定します。これは、アスタリスク (*) を使用したワイルドカードにして、1,1* などの LSM または 0,*,* などの ACS 内でゼロ以外の最高の優先度を持つ CAP を選択できます。

- *opmsg opmsg_nbr*

cmd_proc を使用して入力される SL8500 Bulk CAP 挿入コマンドには、カスタムオペレータパネルメッセージ番号を指定できます。このメッセージは、カートリッジを挿入するために Bulk CAP がロック解除されたときに表示されます。

有効なメッセージ番号は 4 から 99 です。

- 現在、カスタム *opmsg* は SL8500 ライブラリの Bulk CAP に対してのみ表示されます。
- *opmsg* パラメータはオプションです。指定されていない場合は、カートリッジを挿入するデフォルトのメッセージが送信されます。
- 挿入が開始される前に CAP にカートリッジが残されていたか、読み取れないかまたは重複したカートリッジを挿入できない場合、これらのカートリッジを取り外すメッセージが表示されます。これは、挿入に対して *opmsg* パラメータが指定されたときにも発生することがあります。
- カスタム *opmsg* 番号は ACSAPI クライアントまたは ACSLS GUI からの挿入には指定できません。これらの場合は、デフォルトのメッセージが表示されます。

opmsg 番号に対して表示されるメッセージを作成するには、SL コンソールを使用して、次を選択します。

```
Tools
  Configuration
    CAP Usage Message
```

opmsg を表示する SL Console の CAP ステータス表示を表示するには、次を選択します。

```
Tools
  System Detail
    CAP Folder
      Status
```

SL Console に、挿入時に ACSLS から送信されたメッセージ番号に基づいて Bulk CAP メッセージが表示されます。メッセージは、カートリッジを挿入するために CAP がロック解除されたあとに、「*System Details CAP Status*」ページに表示されます。

例: Bulk CAP 1,3,0 からカートリッジを挿入するために、カスタムオペレータパネルメッセージ番号 66 を指定するには:

```
enter 1,3,0 opmsg 66
```

- *lsm_id*

lsm_id オプションを選択すると、単一の LSM で使用可能な複数の CAP を使用してカートリッジを挿入できます。要件 (手動、ゼロ以外の優先度、使用可能など) を満たすすべての CAP はカートリッジの挿入のためにロック解除されます。選択したいずれかまたはすべての CAP から、任意の順序でカートリッジを挿入できます。*venter* は有効でなく、CAP の無効エラーを返します。

使用方法

enter コマンドは、手動モードの CAP にラベル付きカートリッジを挿入する準備をさせるために使用します。

例

- CAP 0,0,2 にカートリッジを挿入する準備をさせるには:

```
enter 0,0,2
```

- LSM 0,0 のゼロ以外の最高の優先度の CAP にカートリッジを挿入する準備をさせるには:

```
enter 0, 0, *
```

- LSM 1,2 にカートリッジを挿入するために、複数の CAP を使用するには:

```
enter 1, 2
```

注:

LSM に挿入されるカートリッジはその LSM と互換性がある必要があります。たとえば、L5500 LSM に挿入できるのは、T9840、T9940、および LTO カートリッジのみです。

関連項目:

情報の内容	参照先
コマンドの取り消し	「cancel」
スクラッチプールの作成または変更	「define pool」
ライブラリからのカートリッジの取り出し	「eject」
カートリッジを挿入するガイドラインと手順	「カートリッジの挿入」
ドライブへのスクラッチカートリッジのマウント	「mount *」
ドライブ上のデータカートリッジのマウント	「mount」
CAP ステータスの表示	「query cap」
CAP モード (手動または自動) の設定	「set cap mode」
CAP 選択の優先度の設定	「set cap priority」
ラベルのないカートリッジをライブラリに挿入するための CAP の準備	「venter」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

すべてのラベル付けカートリッジが挿入されたあとに、次のメッセージが表示され、正常に挿入されたカートリッジごとに、メッセージの 2 行目が繰り返されます。

- *Enter: Enter complete, nn volumes entered*
- *Enter: vol_id Entered through cap_id*
 - *nn* は挿入されたカートリッジの合計数です
 - *vol_id* は挿入されたカートリッジのカートリッジ識別子です

- *cap_id* は、カートリッジの挿入に使用された CAP です

中間メッセージ

なし。

エラーメッセージ

- *Enter: vol_id Enter failed, ACS acs_id full.*

説明: ACS に空きストレージセルがないため、カートリッジは挿入されませんでした。ACSL S がカートリッジを挿入できるように、ACS から 1 つ以上のカートリッジを取り出す (*eject*) 必要があります。

変数:

- *vol_id* は挿入されなかったカートリッジの外部ラベルです。
- *acs_id* は空きストレージセルがない ACS の識別子です。

注:

ACSL S はオンラインの LSM でのみ空きセルを検索します。このメッセージはオフライン LSMS に空きセルがある場合でも発生することがあります。

- *Enter: vol_id Enter failed, Audit in progress.*

説明: *audit* によって、挿入に必要なセルの場所へのアクセスがロックされているため、カートリッジが挿入されませんでした。

変数: *vol_id* は挿入されなかったカートリッジの外部ラベルです。

- *Enter: vol_id Enter failed, CAP cap_id in use.*

説明: 指定された CAP が監査、カートリッジの取り出し、または別の挿入プロセスで使用されているため、カートリッジは挿入されませんでした。

変数:

- *vol_id* は挿入されなかったカートリッジの外部ラベルです。
 - *cap_id* は使用中の CAP です。
- *Enter: vol_id Enter failed, Duplicate label.*

説明: CAP 内のカートリッジの識別子はすでに ACSLS データベース内に存在しているため、カートリッジは挿入されませんでした。

変数: *vol_id* は挿入されなかったカートリッジの外部ラベルです。

- *Enter: Enter failed, Unreadable label.*

説明: カートリッジに外部ラベルがないか、読み取れない外部ラベルであるため、カートリッジは挿入されませんでした。

- *Enter: vol_id Enter failed, Unknown media type label.*

説明: カートリッジはその外部ラベルにメディア識別子がないため、挿入されませんでした。

変数: *vol_id* は挿入されなかったカートリッジの外部ラベルです。

表示領域メッセージ

- *CAP cap_id: Place cartridges in CAP.*

説明: CAP はカートリッジを挿入する準備ができています。CAP を開き、カートリッジを挿入 (*enter*) します。

変数: *cap_id* は、カートリッジの挿入に使用される CAP です。

- *CAP cap_id: Remove cartridges from CAP.*

説明: 1 つまたは複数のカートリッジを挿入できません。

変数: *cap_id* は、カートリッジの挿入に使用される CAP です。CAP を開き、カートリッジを取り外し (*remove*) ます。

- *CAP cap_id: CAP cap_id Place magazines in CAP.*

説明: CAP はカートリッジの挿入にマガジンを使用します。正しいマガジンにカートリッジをロードし、CAP を開いて、マガジンを挿入します。

変数: *cap_id* は、カートリッジの挿入に使用される CAP です。

idle

idle コマンドは、ACSL5 の新しいリクエストの処理を停止します。

形式

- *idle [force]*

idle コマンドには完全なコマンド名を入力します。ACSL S は、*i*、*id*、または *idl* などのその他のすべての形式のコマンドを拒否します。

オプション

- *force*

新しいリクエスト処理を強制的に終了します。

使用方法

idle コマンドは、ACSL S の新しいリクエストの処理を停止するために使用します。たとえば、ACSL S の保守を行う前または終了する前に、ACSL S をアイドル状態 (*idle*) にします。

注:

リクエスト処理を再開するには、*start* コマンドを使用します。

- *idle*

force オプションを付けずに *idle* コマンドを入力すると、ACSL S はアイドル保留状態になります。ACSL S は現在および保留中のリクエスト (保留中のロックリクエストを除きます。これは取り消されます) を完了し、「注」に示されているものを除く新しいリクエストを拒否します。ACSL S はアイドル状態になり、リクエスト処理が再開されるまで、後続のリクエストを処理しません。

- *Idle force*

force オプションを付けて *idle* コマンドを入力すると、ACSL S がアイドル状態になります。ACSL S は現在および保留中のすべてのリクエストを取り消し、「例」の下の「注」に示されているものを除く新しいリクエストを拒否します。再開されるまで、ACSL S は後続のリクエストを処理しません。ACSL S は現在のリクエストを完了しないため、ACSL S を強制的にアイドル状態にすると、データベースがハードウェアと整合性のない状態になる可能性があります。この修正には *audit* が必要です。

ヒント: *idle force* を入力したときの現在のリクエスト処理によっては、イベントログがプロセス障害を報告する場合があります。その場合、影響を受ける LSM をオフラインに変更 (*vary*) し、オンラインに戻します。*vary* コマンドについては、「[vary](#)」を参照してください。

例

- ACSLS をアイドル保留状態にするには:

idle

- ACSLS を強制的にアイドル状態にするには:

idle force

注:

idle または *idle-pending* 状態では、ACSLs は *cancel*、*idle*、*query*、*start*、および *vary* リクエストの新しいリクエストを受け付けます。

関連項目:

情報の内容	参照先
ライブラリコンポーネントのステータスの表示	「query コマンド」
ACSLs リクエスト処理の開始	「start」
ライブラリコンポーネントの状態の変更	「vary」

コマンド領域メッセージ

このセクションでは、領域コマンドメッセージについて説明します。

成功メッセージ

ACSLs リクエスト処理を停止すると、次のメッセージが表示されます。

ACSLM Request Processing Stopped: Success.

中間メッセージ

なし。

エラーメッセージ

ACSLM Request Processing Stopped: status

説明: ACSLS はリクエストの処理を停止しませんでした。

変数: *status* は失敗の理由です。一般的なステータスメッセージの詳細については、ACSLs メッセージを参照してください。

表示領域メッセージ

変数:

- *Server system idle*

説明: リクエスト処理が停止して、ACSL S はアイドル状態です。

- *Server system idle is pending*

説明: ACSLS は現在または保留中のリクエストを処理中です。アイドル状態が保留されま
す。

lock

lock コマンドは、ACSL S が割り当てるロック ID にカートリッジまたはドライブをロックしま
す。

形式

lock type identifier...[wait]

オプション

- *type identifier*

ライブラリコンポーネントを指定します。次の表に、ロックできるコンポーネントの一覧を示
します。単一の *lock* コマンドには、ドライブまたはボリュームの両方ではなくいずれかを指
定できます。ただし、ドライブとボリュームの両方に同じ *lock ID* を使用できます。

表13.4 *lock* に有効なコンポーネント

ライブラリコンポーネント	タイプ	識別子
ドライブ	<i>drive</i>	<i>drive_id</i>
ボリューム	<i>volume</i>	<i>vol_id</i>

- *wait*

コンポーネントが使用できない (ロックされているか使用中) 場合にロックが保留中である
ことを指定します。ACSL S はコンポーネントが使用可能になると、それをロック (*locks*) し
ます。保留中のロックリクエストをクリアまたは取り消し (*cancel*) できます。ACSL S をアイ
ドル状態にすると、保留中のロックリクエストも取り消されます。

使用方法

lock コマンドは、ACSL S が割り当てるロック ID にボリュームまたはドライブをロックするた
めに使用します。使用可能な (ロックされておらず、使用中でない) ボリュームまたはドライブ
のみをロックできます。

注:

`lock` コマンドを入力して、カートリッジまたはドライブをロックすると、ACSLs はボリュームまたはドライブにロック ID を割り当て、ロック ID をボリュームまたはドライブのロック ID に変更します。`set lock` コマンドを使用して、ロック ID を設定することはできないため、`lock` コマンドを使用して、`set lock` で設定したロック ID でボリュームまたはドライブをロックします。

例

- ドライブ 0,1,10,2 をロックするには:

```
lock drive 0,1,10,2
```

- ボリューム EDU445 をロックするには:

```
lock volume EDU445
```

関連項目:

情報の内容	参照先
指定したドライブまたはカートリッジのすべてのアクティブなロックまたは保留中のロックの解除	「clear lock」
ドライブまたはカートリッジのロックステータスの表示	「query lock」
ロック ID の設定	「set lock」
ロック ID またはユーザー ID の表示	「show」
ドライブまたはカートリッジのアクティブなロックの解除	「unlock」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- `lock` リクエストが成功すると、次のメッセージが表示されます。

```
Lock: Lock completed, Success.
```

- リクエスト内の識別子ごとに、次のいずれかのメッセージが表示されます。
 - `Lock: Drive drive_id locked under lock_id lock_id.`
 - `Lock: Volume vol_id locked under lock_id lock_id.`

ここでは:

- `drive_id` はロックされたドライブです。
- `vol_id` はロックされたカートリッジのカートリッジ識別子です。

› *lock_id* はロック ID です。

中間メッセージ

なし。

エラーメッセージ

lock リクエストが失敗した場合、次のいずれかのメッセージが表示されます。

- *Lock: Lock of drive drive_id failed, Drive in use.*

説明: ドライブがすでにロックされているか、使用中であるため、ACSL S は指定されたドライブをロックできません。

変数: *drive_id* は、ACSL S がロックできないドライブです。

- *Lock: Lock of drive drive_id failed, Lock failed.*

説明: ACSLS は指定されたドライブをロックできません。正しい構文とドライブ識別子で *lock* コマンドを再入力します。

変数: *drive_id* は、ACSL S がロックできないドライブです。

- *Lock: Lock of volume vol_id failed, Volume in use.*

説明: カートリッジがすでにロックされているか、使用中であるため、ACSL S は指定されたカートリッジをロックできません。

変数: *vol_id* は、ACSL S がロックできないカートリッジです。

- *Lock: Lock of volume vol_id failed, Lock failed.*

説明: ACSLS は指定されたカートリッジをロックできません。正しい構文とカートリッジ識別子で *lock* コマンドを再入力します。

変数: *vol_id* は、ACSL S がロックできないカートリッジです。

表示領域メッセージ

なし。

logoff

logoff コマンドは、*cmd_proc* を終了します。

形式

logoff

オプション

なし。

使用方法

logoff コマンドは、*cmd_proc* を終了するために使用します。対話式 (ウィンドウ) を実行すると、*cmd_proc*、*logoff* によって、*cmd_proc* ウィンドウも終了します。

例

- *cmd_proc* を終了するには:

```
logoff
```

注:

logoff は *cmd_proc* のみ終了し、すべての ACSLS 状態で有効で、ACSLs 操作に影響しません。

関連項目:

情報の内容	参照先
<i>cmd_proc</i> の開始	「<i>cmd_proc</i> の起動」
<i>cmd_proc</i> の使用	「<i>cmd_proc</i> の起動」

コマンド領域メッセージ

なし。

表示領域メッセージ

なし。

mount

mount コマンドはデータカートリッジをマウントします。

形式

```
mount vol_id drive_id [bypass] [readonly]
```

オプション

- *vol_id*

カートリッジを指定します。

- *drive_id*

ドライブを指定します。

- *bypass*

bypass オプションは次のようにして、カートリッジをマウントしようとする前に、ACSLs が行うチェックをオーバーライドします。

- *bypass* オプションは常に、ACSLs の外部ラベルに対するカートリッジ ID の検証をオーバーライドします。
- それは、テープドライブとカートリッジのメディアタイプ間の互換性の ACSLS 検証をオーバーライドできます。

ライブラリは *mount* リクエストを受け取ると、カートリッジのメディアがテープドライブと互換性があるか検証します。メディアタイプが互換性がないか不明なタイプの場合、ライブラリはマウントを失敗させます。

- *readonly*

カートリッジを書き込み保護でマウントするように指定します。

注意:

LTO ドライブは書き込み保護でのマウントをサポートしません。LTO ドライブに「*mount read-only*」が試みられた場合、それはイベントログ内に「Drive cannot honor write protect」メッセージを伴って失敗します。

また、一部の初期 97xx SCSI 接続ライブラリでは、書き込み保護でのマウントをサポートしていません。*mount* コマンドに *read-only* オプションを指定した場合でも、ドライブはカートリッジに書き込むことがあります。これらのドライブに対してカートリッジを書き込み保護するには、カートリッジの読み取り専用保護 (サムホイールなど) を使用します。

使用方法

mount コマンドは、データカートリッジをマウントするために使用します。*mount* コマンドを入力するたびに、1 つのドライブに 1 つのカートリッジのみマウントできます。

マウントの成功には次が必要です。

- カートリッジとドライブが同じ ACS 内にある必要があります。
- カートリッジは使用可能である必要があります、ドライブがオンラインで使用可能である必要があります。

例

ボリューム EDU010 をドライブ 0,0,10,2 にマウントするには:

```
mount EDU010 0,0,10,2
```

次の例では、YUMA15 は、カートリッジラベルに 7 番目の文字が見つからない DD3C ボリュームです。*bypass* オプションは、メディア互換性チェックをバイパスし、ドライブ 0,0,4,0 の SD3 ドライブへのカートリッジの *mount* を強制的に行います。

bypass オプションを使用して、ドライブ 0,0,4,0 に YUMA15 をマウントするには:

関連項目:

情報の内容	参照先
ドライブからのカートリッジのマウント解除	「dismount」
ラベル付きカートリッジをライブラリに挿入するための CAP (手動モード) の準備	「enter」
CAP ステータスの表示	「query cap」
ドライブステータスの表示	「query drive」
ドライブまたはカートリッジのロックステータスの表示	「query lock」
指定されたデータカートリッジのメディア互換ドライブのステータスの表示	「query mount」
カートリッジの場所とメディアタイプの表示	「query volume」
CAP モード (手動または自動) の設定	「set cap mode」
CAP 選択の優先度の設定	「set cap priority」
ラベルのないカートリッジをライブラリに挿入するための CAP の準備	「venter」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

```
Mount: vol_id mounted on drive_id
```

説明: ACSLS は指定されたカートリッジをマウントしました。

変数:

- *vol_id* は ACSLS がマウントしたカートリッジのカートリッジ識別子です。
- *drive_id* はカートリッジをマウントしたドライブです。

中間メッセージ

なし。

エラーメッセージ

- *Mount: Mount failed, Audit in progress.*

説明: *audit* によって、指定されたカートリッジのセルの場所へのアクセスがロックされているため、ACSLS はカートリッジをマウントできません。

- *Mount: Mount failed, In use.*

説明: ドライブが使用中か、またはリクエストされたカートリッジが別のコマンド用に予約されているため、ACSLS はカートリッジをマウントできません。

- *Mount: Mount failed, Misplaced tape.*

説明: カートリッジの外部ラベルがストレージセルのデータベースのカートリッジ識別子と一致しないため、ACSLS はカートリッジをマウントできません。データベースは、ストレージセル内のカートリッジのカートリッジ識別子、メディアタイプ、およびクリーニングカートリッジ属性を更新します。

- *Mount: Mount failed, Not in same ACS.*

説明: 指定されたカートリッジとドライブが同じ ACS 内にないため、ACSLS はカートリッジをマウントできません。

- *Mount: Mount failed, Cartridge in drive.*

説明: 指定されたカートリッジが、あるドライブにすでにマウントされているため、ACSLS はカートリッジをマウントできません。

- *Mount: Mount failed, Unreadable label.*

説明: カートリッジにラベルがないか、読み取れないラベルか、または仮想ラベルがないため、ACSLS はカートリッジをマウントできません。

- *Mount: Mount failed, Invalid media type.*

説明: 指定されたカートリッジのメディアタイプが無効なため、ACSL S はカートリッジをマウントできません。

- *Mount: Mount failed, Invalid drive type.*

説明: 指定されたドライブタイプが無効なため、ACSL S はカートリッジをマウントできません。

- *Mount: Mount failed, Incompatible media type.*

説明: カートリッジのメディアタイプが指定されたドライブと互換性がないため、ACSL S はカートリッジをマウントできません。

表示領域メッセージ

なし。

mount *

*mount ** コマンドは、スクラッチカートリッジを選択して、それをマウントします。

形式

*mount * drive_id [pool_id] [media media_type | media *]*

オプション

- *drive_id*

ドライブを指定します。

- *pool_id*

ACSL S がスクラッチカートリッジを選択するプールを指定します。*pool_id* はオプションです。*pool_id* を指定しない場合、ACSL S は共通プール (プール 0) からスクラッチカートリッジを見つけようとしています。

pool_id を指定し、プールにスクラッチカートリッジ (または混在メディアライブラリに正しいいずれかのメディア) が含まれていない場合、またはプールがオーバーフロー用に設定されている場合、ACSL S は共通プール (プール 0) からスクラッチカートリッジを見つけようとしています。

- *media media_type | media **

カートリッジメディアタイプを指定します。メディアタイプの指定はオプションです。

使用方法

mount * コマンドは、スクラッチカートリッジを選択して、それをマウントするために使用します。次のセクションでは、ACSL S がマウントするスクラッチカートリッジを選択する方法およびスクラッチカートリッジのカートリッジメディアタイプを指定する方法について説明します。

ACSL S がスクラッチカートリッジを選択する方法

mount * コマンドは、次によってスクラッチカートリッジを選択します。

- 指定されたドライブを格納する LSM との近接度に基づいた、ACS 内の LSM のリストの作成。
- プールおよびメディアタイプの条件を満たすスクラッチカートリッジが見つかるまで、リスト内の各 LSM の調査。
- その LSM で最終アクセス日がもっとも古いスクラッチカートリッジの選択。

選択されたスクラッチカートリッジがドライブにマウントされます。

メディアタイプの指定

次のいずれかの方法でカートリッジのメディアタイプを指定できます。

- *mount* * コマンドのこの形式でメディアタイプを明示的に指定します。

```
mount * drive_id [pool_id] media media_type
```

例: 共通プール (プール 0) から、9940 スクラッチカートリッジをマウントするには:

```
mount * 0,0,10,2 media STK2P
```

- ACSLS がスクラッチ優先順位によってメディアを選択するように、メディアタイプにワイルドカード (*) を使用します。詳細については、「[拡張ストア機能の使用](#)」を参照してください。スクラッチ優先順位を使用するには、*mount* * コマンドのこの形式を入力します。

```
mount * drive_id [pool_id] media *
```

例: スクラッチ優先順位を使用して、共通プール (プール 0) からスクラッチカートリッジをマウントするには:

```
mount * 0,0,10,2 media *
```

- `mount` * コマンドのこの形式でメディアオプションを省略します。これにより、ドライブと互換性のあるメディアタイプを選択するように ACSLS に指示します。

```
mount * drive_id [pool_id]
```

例: 共通プール (プール 0) からドライブと互換性のあるスクラッチカートリッジをマウント (`mount`) するには:

```
mount * 0, 0, 10, 2
```

例

次のセクションでは、単一メディアライブラリおよび混在メディアライブラリの `mount scratch` の例を示します。

単一メディアライブラリ

- プール 5 からスクラッチカートリッジをドライブ 0,0,10,2 にマウント (`mount`) するには:

```
mount * 0, 0, 10, 2 5
```

ヒント: プール 5 から使用可能なカートリッジがなく、`overflow` 用に設定されている場合、ACSLS は共通プール (プール 0) からカートリッジを選択します。

- 共通プール (プール 0) からスクラッチカートリッジをドライブ 0,0,10,0 にマウント (`mount`) するには:

```
mount * 0, 0, 10, 0
```

混在メディアライブラリ

- プール 5 から T10000T2 のメディアタイプのスクラッチカートリッジをドライブ 0,0,10,2 にマウント (`mount`) するには:

```
mount * 0, 0, 10, 2 5 media T10000T2
```

ヒント: プール 5 から使用可能なカートリッジがなく、`overflow` 用に設定されている場合、ACSLS は共通プール (プール 0) から指定されたメディアタイプのカートリッジを選択します。

- プール 10 からスクラッチ優先順位によって判断されたメディアタイプのスクラッチカートリッジをドライブ 0,0,2,3 にマウント (`mount`) するには:

```
mount * 0,0,2,3 10 media *
```

ヒント: プール 10 から使用可能なカートリッジがなく、`overflow` 用に設定されている場合、ACSLs は共通プール (プール 0) から指定されたメディアタイプのカートリッジを選択します。

- 共通プール (プール 0) から `T10000T2` のメディアタイプのスクラッチカートリッジをドライブ `0,0,10,2` にマウント (`mount`) するには:

```
mount * 0,0,10,2 media T10000T2
```

- 共通プール (pool 0) からスクラッチ優先順位によって判断されたメディアタイプのスクラッチカートリッジをドライブ `0,0,2,3` にマウント (`mount`) するには:

```
mount * 0,0,2,3 media *
```

- 共通プール (pool 0) からドライブ `0,0,2,3` と互換性のあるメディアのスクラッチカートリッジをマウント (`mount`) するには:

```
mount * 0,0,2,3
```

関連項目:

情報の内容	参照先
スクラッチプールの作成または変更	「define pool」
空のスクラッチプールの削除	「delete pool」
ドライブからのカートリッジのマウント解除	「dismount」
スクラッチカートリッジの管理	「LSM の設定」
ドライブステータスの表示	「query drive」
ドライブまたはカートリッジのロックステータスの表示	「query lock」
指定されたスクラッチプールのメディア互換ドライブのステータスの表示	「query mount *」
スクラッチプール属性の表示	「query pool」
スクラッチカートリッジステータスの表示	「query scratch」
カートリッジの場所とメディアタイプの表示	「query volume」
カートリッジスクラッチ属性の設定またはクリア	「set scratch」
スクラッチ優先順位の設定	「拡張ストア機能の使用」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- *Mount: vol_id mounted on drive_id*

説明: ACSLS は指定されたカートリッジをマウントしました。

変数:

- *vol_id* は ACSLS がマウントしたカートリッジのカートリッジ識別子です。
- *drive_id* はカートリッジをマウントしたドライブです。

中間メッセージ

なし。

エラーメッセージ

- *Mount: Mount failed, Audit in progress.*

説明: *audit* によって、指定されたカートリッジのセルの場所へのアクセスがロックされているため、ACSLS はカートリッジをマウントできません。

- *Mount: Mount failed, In use.*

説明: ドライブが使用中か、またはリクエストされたカートリッジが別のコマンド用に予約されているため、ACSLS はカートリッジをマウントできません。

- *Mount: Mount failed, Misplaced tape.*

説明: カートリッジの外部ラベルがストレージセルのデータベースカートリッジ識別子と一致しないため、ACSLS はカートリッジをマウント (*mount*) できません。データベースは、ストレージセル内のカートリッジのカートリッジ識別子、メディアタイプ、およびクリーニングカートリッジ属性を更新します。

- *Mount: Mount failed, Not in same ACS.*

説明: 指定されたカートリッジとドライブが同じ ACS 内にないため、ACSLS はカートリッジをマウント (*mount*) できません。

- *Mount: Mount failed, Cartridge in drive.*

説明: 指定されたカートリッジが、あるドライブにすでにマウントされているため、ACSLS はカートリッジをマウントできません。

- *Mount: Mount failed, Unreadable label.*

説明: カートリッジにラベルがないか、読み取れないラベルか、または仮想ラベルがないため、ACSL S はカートリッジを *mount* できません。

- *Mount: Mount failed, Invalid media type.*

説明: 指定されたカートリッジのメディアタイプが無効なため、ACSL S はカートリッジをマウント (*mount*) できません。

- *Mount: Mount failed, Invalid drive type.*

説明: 指定されたドライブタイプが無効なため、ACSL S はカートリッジをマウント (*mount*) できません。

- *Mount: Mount failed, Incompatible media type.*

説明: カートリッジのメディアタイプが指定されたドライブと互換性がないため、ACSL S はカートリッジをマウント (*mount*) できません。

- *Mount: Mount failed, No compatible scratch cartridges in pool.*

説明: 指定されたドライブの ACS 内にスクラッチカートリッジのメディアタイプに一致するスクラッチカートリッジがないため、ACSL S はカートリッジをマウント (*mount*) できません。さらに、プールにオーバーフロー属性が設定されている場合、有効なメディアタイプのスクラッチカートリッジがありません。

表示領域メッセージ

- *Pool pool_id: low water mark warning.*

説明: 指定したスクラッチプール内のカートリッジの数がカートリッジの下限しきい値以下です。

変数: *low_water_mark* は指定されたスクラッチプールの下限しきい値です。

- *Pool pool_id: high water mark warning.*

説明: 指定されたスクラッチプール内のカートリッジの数はカートリッジの上限しきい値以上です。

変数: *high_water_mark* は指定されたスクラッチプールのカートリッジの上限しきい値です。

move

`move` コマンドは、指定されたカートリッジを指定された LSM 内の使用可能なストレージまたは特定のストレージセルに移動します。

注:

SL3000 では、カートリッジを特定のセルに移動できます。詳細については、「[ライブラリのパーティション分割またはパーティション ID の変更](#)」を参照してください。

形式

```
move vol_id lsm_id or move vol_id cell_id
```

オプション

- `vol_id`

カートリッジを指定します。

- `lsm_id`

移動されるカートリッジを格納する LSM を指定します。

- `cell_id`

カートリッジを移動するセルを指定します。

使用方法

`move` コマンドは、指定されたカートリッジを次のいずれかにある使用可能なストレージに移動するために使用します。

- 同じ LSM 内の別のパネル。たとえば、LSM 内のパネル全体を空にする場合に、そのパネル内のすべてのカートリッジを同じ LSM 内の別の場所に移動します。
- 別の LSM。

`move` コマンドを入力するたびに、1 つのカートリッジのみを移動できます。カートリッジが現在存在する LSM が指定された場合、ACSL5 はその LSM 内の別のパネルにカートリッジを移動します。そうでない場合、ACSL5 は指定された LSM にそれを移動します。`move` コマンドは、取り消しできません。

移動の成功には次が必要になります。

- カートリッジが使用可能であり、指定された LSM と同じ ACS 内にある必要があります。

- カートリッジが現在存在する LSM と、指定された LSM がどちらもオンラインである必要があります。指定された LSM に 1 つ以上使用可能なストレージセルがある必要があります。移動が同じ LSM 内の場合、その LSM 内の別のパネルに 1 つ以上使用可能なセルがある必要があります。移動でパススルーが必要な場合、使用されるすべての LSM もオンラインである必要があります。

例

- カートリッジ EDU010 (LSM 0,1 に存在する) をこの LSM 内の別のパネルに移動するには:

```
move EDU010 0,1
```

- カートリッジ EDU010 (LSM 0,1 に存在する) を LSM 0,2 に移動するには:

```
move EDU010 0,2
```

関連項目:

情報の内容	参照先
ドライブまたはカートリッジのロックステータスの表示	「query lock」
LSM のステータスの表示	「query lsm」
カートリッジの場所とメディアタイプの表示	「query volume」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- *Move: vol_id moved to location cell_id*

説明: ACSLS は指定されたカートリッジを指定されたセルの場所に移動しました。

変数:

- *vol_id* は ACSLS が移動したカートリッジのカートリッジ識別子です。
- *cell_id* は指定されたカートリッジの新しいセルの場所です。

中間メッセージ

なし。

エラーメッセージ

Move: Move failed

query コマンド

`query` コマンドは、ライブラリコンポーネントのステータスを表示します。形式、オプション、および使用方法などの各 `query` コマンドの詳細については、次のセクションを参照してください。

形式

次に、`query` コマンドの一般的な形式を示します。

```
query type [subtype | *] identifier... | all
```

注:

`cancel` リクエストが保留中であるか、または現在の `query` リクエストに対して発行された場合、情報の表示は停止されます。

関連項目:

情報の内容	参照先
コマンドの取り消し	「cancel」
display コマンド	「display コマンドオプションの使用」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

なし。

注:

`query` が成功した場合は、リクエストされたステータスが表示されます。

中間メッセージ

なし。

エラーメッセージ

- *Library not available.*

説明: ACSLS が回復中であるため、*query* が失敗しました (query server を除くすべてのクエリー)。

- *Cartridge identifier vol_id not found.*

説明: 指定されたカートリッジがライブラリにないため、*query mount ** コマンドはそのステータスを表示できません。

変数: *vol_id* は指定されたカートリッジです。

- *Invalid media type*

説明: 指定されたカートリッジが無効なメディアタイプであるか、コマンドに無効なメディアタイプが指定されたため、*query mount ** コマンドはそのステータスを表示できません。

変数: *vol_id* は指定されたカートリッジです。

表示領域メッセージ

なし。

query acs

query acs コマンドは ACS ステータスを表示します。

形式

```
query acs acs_id... | all
```

オプション

```
acs_id | all
```

query する ACS またはすべての ACS の場合は *all* を指定します。

使用方法

query acs コマンドは、次の形式で ACS のステータスを表示するために使用します。

```
yyy-mm-dd hh:mm:ss ACS Status
```

Identifier	State	Free Cell Count	Audit C/P	Mount C/P	Dismount C/P	Enter C/P	Eject C/P
acs_id	state	count	n/n	n/n	n/n	n/n	n/n

ここでは:

- *acs_id*

ACS 識別子です。

- *state* は、次の ACS 状態のいずれかになります。

- *online*

ACS はオンラインです。

- *offline*

ACS はオフラインです。

- *offline pending*

ACS は現在および保留中のリクエストを処理して、オフラインになります。ACS は新しいリクエストを拒否します。

- *diagnostic*

ACS は現在および保留中のリクエストのみを処理し、新しいリクエストを拒否 (*reject*) します。ACS はクライアントアプリケーションには使用できず、*cmd_proc* を使用してのみ制御できます。*vary* コマンドは ACS をオンラインに変更するために使用します。

- *recovery*

ACS は初期化中であるか、またはエラーから回復中です。ACS がオンラインになるまで待ちます。

- *count*

ACS 内の空きセルの数です。

- *n*

ライブラリリソースを必要とする各コマンド (*audit*、*mount*、*dismount*、*enter*、および *eject*) の ACS に対する現在 (C) と保留中 (P) のリクエストの数です。

例

- ACS 1 をクエリー (*query*) するには:

```
query acs 1
```

- ライブラリ内のすべての ACS をクエリー (*query*) するには:

```
query acs all
```

関連項目:

情報の内容	参照先
ライブラリコンポーネントの状態の変更	「vary」
リクエストのステータスの表示	「query request」

query cap

query cap コマンドは、CAP のステータスを表示します。

形式

```
query cap cap_id... | all
```

オプション

- cap_id* | *all*

クエリー (*query*) する CAP またはすべての CAP の場合は *all* を指定します。

注:

アスタリスクを含む *cap_id* を指定することはできません。

使用方法

query cap コマンドは、CAP のステータスを表示するために使用します。

query cap コマンドは、次の形式で CAP のステータスを表示します。

```

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*

CAP 識別子です。

- cap_priority*

CAP の優先度です。

- *cap_size*

CAP 内のセルの数です。

- *cap_state*

次のいずれかの CAP 状態です。

- *online*

CAP はオンラインです。

- *offline*

CAP はオフラインです。

- *offline-pending*

CAP は現在および保留中のリクエストを処理して、オフラインになります。CAP はすべての新しいリクエストを拒否します。

- *diagnostic*

CAP は現在および保留中のリクエストのみを処理し、新しいすべてのリクエストを拒否します。CAP はクライアントアプリケーションには使用できず、*cmd_proc* を使用してのみ制御できます。*vary* コマンドは CAP をオンラインに変更するために使用します。

- *recovery*

CAP は初期化中であるか、またはエラーから回復中です。CAP がオンラインになるまで待ちます。

- *cap_mode*

次のいずれかの CAP 挿入モードです。

- *manual*

カートリッジを挿入する前に CAP をロック解除する必要があります。

- *automatic*

CAP はカートリッジを挿入する準備ができています。

- *status*

次のいずれかの CAP のステータスです。

- *available*

CAP は使用可能です。

- *enter*

CAP は使用できません (カートリッジを挿入するために予約されています)。

- *eject*

CAP は使用できません (カートリッジを取り出すために予約されています)。

- *audit*

CAP は使用できません (監査処理のために予約されています)。

例

- CAP 0,1,0 をクエリー (*query*) するには:

```
query cap 0,1,0
```

- ライブラリ内のすべての CAP をクエリーするには:

```
query cap all
```

関連項目:

情報の内容	参照先
リクエストのステータスの表示	「query request」
CAP モード (手動または自動) の設定	「set cap mode」
CAP 選択の優先度の設定	「set cap priority」
ライブラリコンポーネントの状態の変更	「vary」

query clean

query clean コマンドは、クリーニングカートリッジのステータスを表示します。不在または取り出し済みのカートリッジは表示されません。テープドライブによって使用済み (使い切った) と報告されたクリーニングカートリッジは報告されません。

形式

```
query clean vol_id... | all
```

オプション

- *vol_id* | *all*

クエリーするクリーニングカートリッジまたはすべてのカートリッジの場合は *all* を指定します。

使用方法

query clean コマンドは、次の形式でクリーニングカートリッジのステータスを表示するために使用します。

```
yyy-mm-ddhh:mm:ssDrive Status
Identifier State Status volumeType
drive_id state status vol_idtype
```

ここでは:

- *vol_id*

クリーニングカートリッジのカートリッジ識別子です。

- *cell_id*

クリーニングカートリッジの場所です。

- *max_usage*

クリーニングカートリッジを使用できる回数です。

- *current_usage*

クリーニングカートリッジが使用された回数です。

- *status*

クリーニングカートリッジの場所です。

- *home*

カートリッジはストレージセル内にあります。

- *in drive*

カートリッジはドライブ内にあります。

- *in transit*

カートリッジは移動中です。

- *type*

カートリッジのメディアタイプ (*3480*、*DD3D*、*DLTIII*、*STK1R* など) です。

例

- クリーニングカートリッジ *J35992* のステータス情報を表示するには:

```
query clean J35992
```

- すべてのクリーニングカートリッジをクエリーするには:

```
query clean all
```

関連項目:

情報の内容	参照先
ドライブのクリーニングのガイドラインと手順	「LSM の設定」
クリーニングカートリッジの属性の設定	「set clean」
クリーニングカートリッジおよび使用済みのクリーニングカートリッジの表示	「display コマンドオプションの使用」

query drive

query drive コマンドは、ドライブのステータスを表示します。

形式

```
query drive drive_id... | all
```

オプション

- *drive_id* | *all*

クエリーするドライブまたはすべてのドライブの場合は *all* を指定します。

使用方法

query drive コマンドは、次の形式でドライブのステータスを表示するために使用します。

ここでは:

- *drive_id*

ドライブ識別子です。

- *state*

次のいずれかです。

- *online*

ドライブはオンラインです。

- *offline*

ドライブはオフラインです。

- *diagnostic*

ドライブは現在および保留中のリクエストのみを処理し、新しいすべてのリクエストを拒否します。ドライブはクライアントアプリケーションには使用できず、*cmd_proc* を使用してのみ制御できます。*vary* コマンドはドライブをオンラインに変更するために使用します。

- *recovery*

ドライブは初期化中であるか、またはエラーから回復中です。ドライブがオンラインになるまで待ちます。

- *status*

次のいずれかのドライブのステータスです。

- *In use*

ドライブはカートリッジがマウントされているか、またはマウント用に予約されています。

考えられるシナリオ: *query drive all* を実行し、ドライブが使用中であるというメッセージを受け取ります。次に、*display drive ** を実行し、ドライブが予約されているというメッセージを受け取ります。これは、ドライブの予約済みステータスが、*mount* リクエストの進行中で、カートリッジがドライブまでの途中にあることを示していることを意味します。同時に、ドライブは使用中であると見なされます。

- *Available*

ドライブはマウントに使用できます。

- *vol_id*

ドライブ内のカートリッジの識別子です。ドライブにカートリッジがない場合、またはカートリッジの外部ラベルが読み取れないか不明な場合、このフィールドは空白になります。

- *drive_type*

ドライブタイプです。

例

- ドライブ 0,3,1,0 をクエリーするには:

```
query drive 0,3,1,0
```

- すべてのドライブをクエリーするには:

```
query drive all
```

関連項目:

情報の内容	参照先
ライブラリコンポーネントの状態の変更	「vary」
ドライブからのカートリッジのマウント解除	「dismount」
ドライブシリアル番号	「display コマンドオプションの使用」 および 「display コマンドオプションの使用」

query lmu

query lmu コマンドは、シングル LMU とデュアル LMU の両方の ACS 構成の LMU とポートのステータスと、ACS およびポートの望ましい状態を表示します。ライブラリがパーティション分割されている場合は、パーティション ID も表示します。

query lmu コマンドは、管理するライブラリとの ACSLS の通信をモニターする最適な方法です。*query lmu*:

- ライブラリへの ACSLS 接続のステータスを示します。
- 一連のライブラリのいずれかが RE を報告した場合に冗長電子装置 (RE) を示します。
- RE ライブラリ以外または SL8500 以外のシングルまたはデュアル LMU を示します。
- ACS、ポート接続、LSM、およびテープドライブの目的の状態を表示します。
- パーティション分割されたライブラリのパーティション ID を表示します。

注:

ACSLs は、ホスト/LMU マイクロコード互換性レベル 12 の 9330 LMU に対してのみデュアル LMU 構成をサポートしています。同じマイクロコードレベルが両方の LMU にロードされている必要があります。

形式

```
query lmu acs_id... | all
```

オプション

- `acs_id | all`

その LMU をクエリーする ACS を指定するか、すべての ACS の LMU をクエリーする場合は `all` を選択します。

使用方法

`query lmu` コマンドは、シングル LMU とデュアル LMU の両方の ACS 構成の LMU とポートのステータスと、目的の状態を表示するために使用します。次の例に、冗長電子装置のないライブラリと冗長電子装置を備えたライブラリの出力を示します。

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

ここでは:

- *acs_id*

ACS 識別子です。

- *mode*

LMU モード (*Dual LMU*, *Single LMU*, または *SCSI LMU*) です。

- *status*

アクティブまたはスタンバイ LMU のステータス (*Communicating*, *Not Communicating*, または *Offline*) です。

- *partition_status*

ライブラリがパーティション分割されているかどうかを示します。パーティション分割されている場合、ライブラリは接続先のパーティションを表示します。パーティションのステータス:

- *Not Partitioned*

- *Partition 1-n*

パーティション番号。

- *acs_state*

実際の ACS の状態です。状態:

- *online*

ACS はオンラインです。

- *diagnostic*

ACS は現在および保留中のリクエストのみを処理し、新しいすべてのリクエストを拒否します。ACS はクライアントアプリケーションには使用できず、*cmd_proc* を使用してのみ制御できます。*vary* コマンドは ACS をオンラインに変更するために使用します。

- *offline*

ACS はオフラインです。

- *offline pending*

ACS は現在および保留中のリクエストを処理して、オフラインになります。ACS は新しいリクエストを拒否します。

- *acs_desired_state*

ACS をこの状態にしたいと考えます。目的の状態:

- *online*
- *diagnostic*
- *offline*

- *port_id*

ポート識別子です。

- *port_state*

次のいずれかの実際のポートの状態です。

- *online*

ポートはオンラインです。

- *offline*

ポートはオフラインです。

- *port_desired_state*

ポートをこの状態にしたいと考えます。目的の状態:

- *online*
- *offline*

- *role (des)*

LMU の役割と指定 (A または B) です。ここで役割は:

- *Active*

LMU はアクティブな役割です (LMU は ACS を管理しています)。

- *Standby*

LMU はスタンバイ *role* です (ACS を管理しておらず、アクティブな LMU と通信しており、スイッチオーバーに使用できます)。

注:

スイッチオーバー時に、*role* フィールドの情報が最新でなく、単一のダッシュ (-) として表示されることがあります。情報が最新になると、ACSL5 は各 LMU の実際の役割で「*role*」フィールドをリフレッシュします。

- *compat_level*

ホスト/LMU マイクロコードの互換性レベルです。デュアル LMU 構成には、レベル 11 以上が必要です。

- *dev_name*

ポートデバイス名です。

例

- すべての ACS を管理するすべての LMU の LMU とポートのステータスを表示するには:

```
query lmu all
```

- ACS 0 および 1 を管理するすべての LMU の LMU とポートのステータスを表示するには:

```
query lmu 0 1
```

関連項目:

情報の内容	参照先
ACS のアクティブ LMU からスタンバイ LMU への ACS 管理の手動での切り替え	「switch lmu」

query lock

query lock コマンドは、ドライブまたはカートリッジのロックステータスを表示します。

形式

```
query lock type identifier... | all
```

オプション

- *type identifier | all*

次の表に示すように、クエリーするドライブまたはカートリッジ、またはすべてのドライブまたはカートリッジの場合は *all* を指定します。

表13.5 query lock に有効なロックのタイプ

ライブラリコンポーネント	タイプ	識別子
ドライブ	<i>drive</i>	<i>drive_id</i>
ボリューム	<i>volume</i>	<i>vol_id</i>

使用方法

query lock コマンドは、次の形式でドライブまたはカートリッジのロックステータスを表示するために使用します。

```

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
    
```

ここでは:

- *vol_id*

指定されたカートリッジの識別子です。

- *drive_id*

指定されたドライブの識別子です。

- *lock_id*

ロック ID です。

- *duration*

秒単位でのロックがアクティブにされていた時間です。

- *pending*

カートリッジまたはドライブを待機しているロックリクエストの数です。

- *status* は次のいずれかのステータスです。

- *available*

カートリッジまたはドライブは使用可能です。

- *in use*

カートリッジまたはドライブが使用中か、またはマウント用に予約されています。

- *user_id*

カートリッジまたはドライブをロックしているユーザー ID です。*user_id* は 80 文字のあとに折り返します。

例

- ドライブ 1,0,4,0 のロックステータス情報を表示するには:

```
q loc dr 1,0,4,0
```

- すべてのドライブのロックステータス情報を表示するには:

```
query lock drive all
```

- カートリッジ SL4493 のロックステータス情報を表示するには:

```
query lock cartridge SL4493
```

- すべてのカートリッジのロックステータス情報を表示するには:

```
query lock cartridge all
```

関連項目:

情報の内容	参照先
指定したドライブまたはカートリッジのすべてのアクティブなロックまたは保留中のロックの解除	「clear lock」
ドライブとカートリッジのロック	「lock」
ロック ID の設定	「set lock」
ロック ID またはユーザー ID の表示	「show」
アクティブなロックの解除	「unlock」

query lsm

query lsm コマンドは、LSM のステータスを表示します。

形式

```
query lsm lsm_id... | all
```

オプション

- *lsm_id | all*

クエリーする LSM またはすべてのロックの場合は *all* を指定します。

`query lsm` コマンドは、次の形式で LSM のステータスを表示するために使用します。

```
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     C/P
```

ここでは:

- *lsm_id*

LSM 識別子です。

- *state* は次のいずれかの LSM の状態です。

- *diagnostic*

LSM は現在および保留中のリクエストのみを処理し、新しいすべてのリクエストを拒否します。LSM はクライアントアプリケーションには使用できず、`cmd_proc` 使用してのみ制御できます。`vary` コマンドは LSM をオンラインに変更するために使用します。

- *offline*

LSM はオフラインです。

- *offline pending*

LSM は現在および保留中のリクエストを処理して、オフラインになります。LSM は新しいリクエストを拒否します。

- *online*

LSM はオンラインです。

- *recovery*

LSM は初期化中であるか、またはエラーから回復中です。LSM がオンラインになるまで待ちます。

- *count*

LSM 内の空きストレージセルの数です。

- *n*

ライブラリリソースを必要とする各コマンド (`audit`、`mount`、`dismount`、`enter`、および `eject`) の LSM に対する現在 (C) と保留中 (P) のリクエストの数です。

例

- ACS 0 の LSM 1 のステータス情報を表示するには:

```
query lsm 0,1
```

- すべての LSM のステータス情報を表示するには:

```
query lsm all
```

関連項目:

情報の内容	参照先
リクエストのステータスの表示	「 query request 」
ライブラリコンポーネントの状態の変更	「 vary 」

query mount

`query mount` コマンドは、SL8500 または接続された一連の SL8500 など、複数の LSM のあるライブラリのパフォーマンスを最適化するために使用します。指定されたデータカートリッジのメディア互換ドライブのステータスを表示します。これらのドライブは、カートリッジが不在か、または取り出し済みの場合に表示されません。さらに:

- `query mount` リクエストにドライブのリストが返されるように選択する場合、ドライブは指定したボリュームと互換性がある必要があります。
- ドライブは主にパススルー距離によって順序付けされます。LSM 内のカートリッジにもっとも近いドライブが最初に表示されます。
- ACSLS 7.3 以降のリリースでは、カートリッジから同じパススルー距離にあるドライブは、もっとも以前に使用された順になります。

例: カートリッジがマウント解除されてからの時間がもっとも長い互換性のあるドライブが最初で、次に長い時間のドライブが 2 番目というようになります。

形式

```
query mount vol_id
```

オプション

- `vol_id`

クエリーするカートリッジを指定します。

使用方法

`query mount` コマンドは、カートリッジと同じ ACS に接続されており、指定されたカートリッジのメディアタイプと互換性があるすべてのライブラリドライブのステータスを表示するために使用します。`query mount` は、あらゆる状態 (オンライン、オフライン、オフライン保留、または診断) の LSM のドライブのステータスを表示します。

互換性のあるドライブは、次の形式で指定されたカートリッジまでの近接度によって順序付けされます。

```

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

```

ここでは:

- `vol_id`

指定されたカートリッジの識別子です。

- `vol_stat`

カートリッジの場所です。

- `home`

カートリッジはストレージセル内にあります。

- `in drive`

カートリッジはドライブ内にあります。

- `in transit`

カートリッジが移動中であるか、または見つかりません。

- `drive_id`

指定されたカートリッジのメディアタイプと互換性のあるすべてのライブラリドライブのリストです。

- `drive_id`

ドライブ識別子です。

- `state`

次のいずれかのドライブの状態です。

- *online*

ドライブはオンラインです。

- *offline*

ドライブはオフラインです。

- *diagnostic*

ドライブは現在および保留中のリクエストのみを処理し、新しいすべてのリクエストを拒否します。ドライブはクライアントアプリケーションには使用できず、*cmd_proc* を使用してのみ制御できます。*vary* コマンドはドライブをオンラインに変更するために使用します。

- *recovery*

ドライブは初期化中であるか、またはエラーから回復中です。ドライブがオンラインになるまで待ちます。

- *status*

次のいずれかのドライブのステータスです。

- *In use*

ドライブはカートリッジがマウントされているか、またはマウント用に予約されています。

- *Available*

ドライブはマウントに使用できます。

- *inu_id*

ドライブ内のカートリッジの識別子です。カートリッジ ID は、*drive_stat* が使用中の場合にのみ表示されます。

- *drive_type*

ドライブタイプです。

例

- カートリッジ ZUNI14 までの近接度によって順序付けされたドライブのステータス情報を表示するには:

query mount ZUNI14

関連項目:

情報の内容	参照先
ドライブ上のデータカートリッジのマウント	「mount」
ドライブまたはカートリッジのロックステータスの表示	「query lock」
カートリッジの場所とメディアタイプの表示	「query volume」
ライブラリコンポーネントの状態の変更	「vary」

query mount *

*query mount ** コマンドは、1 つ以上の指定されたスクラッチプール内のメディアと互換性のあるドライブ (および、オプションで、プール内の特定のカートリッジメディアタイプと互換性のあるドライブのみ) のステータスを表示します。

形式

*query mount * pool_id... [media media_type | media *]*

オプション

- *pool_id*

クエリーするスクラッチプールまたは複数のプールを指定します。

- *media media_type | media **

メディアタイプを指定します。

使用方法

*query mount ** コマンドは、指定されたスクラッチプール内およびカートリッジと同じ ACS 内のすべてのカートリッジのメディアタイプと互換性のあるすべてのライブラリドライブのステータスを表示するために使用します。プール 0 は共通のスクラッチプールです。*media_type* オプションは、表示をプール内の特定のカートリッジメディアタイプと互換性のあるドライブに制限するために指定します。表示されるドライブは、もっとも深いスクラッチプールまでの近接度によって順序付けされます。*query mount ** はあらゆる状態 (オンライン、オフライン、オフライン保留、または診断) の LSM 内のドライブのドライブステータスを表示します。

*query mount ** コマンドは、次の形式でドライブのステータスを表示します。

<i>yyy-mm-dd</i>	<i>hh:mm:ss</i>	Mount	Scratch	Status		
Identifier	Drive	State	Volume	Status		Drive Type
<i>pool_id</i>	<i>drive_id</i>	<i>state</i>	<i>vol_id</i>	<i>drive_stat</i>		<i>drive_type</i>

ここでは:

- *pool_id*

指定されたスクラッチプールです。

- *drive_id*

指定されたプール内のすべてのメディアタイプ、またはメディアタイプが指定された場合にプール内の特定のメディアタイプと互換性のあるライブラリのすべてのドライブのリストです。

- *state*

次のいずれかのドライブの状態です。

- *online*

ドライブはオンラインです。

- *offline*

ドライブはオフラインです。

- *diagnostic*

ドライブは現在および保留中のリクエストのみを処理し、新しいすべてのリクエストを拒否します。ドライブはクライアントアプリケーションには使用できず、*cmd_proc* を使用してのみ制御できます。*vary* コマンドはドライブをオンラインに変更するために使用します。

- *recovery*

ドライブは初期化中であるか、またはエラーから回復中です。ドライブがオンラインになるまで待ちます。

- *vol_id*

ドライブ内のカートリッジの識別子です。カートリッジ ID は、*drive_stat* が使用中の場合にのみ表示されます。

- *drive_stat*

次のいずれかのドライブのステータスです。

- *In use*

ドライブはカートリッジがマウントされているか、またはマウント用に予約されています。

- *Available*

ドライブはマウントに使用できます。

- *drive_type*

ドライブタイプです。

例

プール 5 内のスクラッチテープの最大集中度への近接度によって一覧表示された互換性のあるドライブのステータスを表示するには:

```
query mount * 5
```

共通プール 0 内の 3480 スクラッチテープの最大集中度への近接度によって一覧表示された互換性のあるドライブのステータスを表示するには:

```
query mount * 0 media 3480
```

関連項目:

情報の内容	参照先
スクラッチプールの作成または変更	「define pool」
空のスクラッチプールの削除	「delete pool」
ドライブへのスクラッチカートリッジのマウント	「mount *」
ドライブまたはカートリッジのロックステータスの表示	「query lock」
スクラッチプール属性の表示	「query pool」
スクラッチカートリッジステータスの表示	「query scratch」
カートリッジスクラッチ属性の設定またはクリア	「set scratch」
ライブラリコンポーネントの状態の変更	「vary」

query pool

`query pool` コマンドはスクラッチプール属性を表示します。

形式

```
query pool pool_id... | all
```

オプション

- *pool_id* | *all*

クエリーするスクラッチプールまたはすべてのプールの場合には *all* を指定します。プール 0 は共通プールです。

使用方法

query pool コマンドは、次の形式でスクラッチプール属性を表示するために使用します。

```
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
```

ここでは:

- *pool_id*

指定されたスクラッチプールです。

- *vol_count*

プール内のスクラッチカートリッジの数です。

カウントされないカートリッジは、プール内の不在および取り出し済みのスクラッチカートリッジとデータカートリッジです。

- *low_water_mark*

カートリッジ警告の下限しきい値です。スクラッチカートリッジカウントがこのしきい値を下回る場合、ACSLs はイベントログに警告メッセージを記録します。

値のあとの「-」は、スクラッチカートリッジカウントがカートリッジの下限しきい値を下回っていることを意味します。

- *high_water_mark*

カートリッジ警告の上限しきい値です。スクラッチカートリッジカウントがこのしきい値に達するか超える場合、ACSLs は警告メッセージをイベントログに記録します。値のあとの「+」は、スクラッチカートリッジカウントがカートリッジの上限しきい値以上であることを意味します。

- *attribute*

指定されたスクラッチプールに *overflow* が設定されている (*set scratch* コマンドを使用して) 場合に表示されます。*overflow* は、*mount scratch* * リクエストが、指定されたスクラッチプールからのカートリッジで満足できない場合に、共通スクラッチプール (プール 0) からスクラッチカートリッジを選択することを指定します。

例

- スクラッチプール 5 のステータス情報を表示するには:

```
query pool 5
```

- すべてのスクラッチプールのステータス情報を表示するには:

```
query pool all
```

関連項目:

情報の内容	参照先
スクラッチプールの作成または変更	「define pool」
空のスクラッチプールの削除	「delete pool」
スクラッチカートリッジステータスの表示	「query scratch」
カートリッジスクラッチ属性の設定またはクリア	「set scratch」

query port

query port コマンドは、ポートのステータスを表示します。

形式

```
query port port_id... | all
```

オプション

- port_id* | *all*

クエリーするポート、またはすべてのポートの場合は *all* を指定します。

使用方法

query port コマンドは、次の形式でポートのステータスを表示するために使用します。

```
yyy-mm-dd hh:mm:ss Port Status
State Identifier
```

```
state port_id
```

ここでは:

- *state*

次のいずれかのポート状態です。

- *online*

ポートはオンラインです。

- *offline*

ポートはオフラインです。

- *port_id*

ポート識別子です。

例

- ポート 0,0 のステータス情報を表示するには:

```
query port 0,0
```

- すべてのポートをクエリーするには:

```
query port all
```

関連項目:

情報の内容	参照先
ライブラリコンポーネントの状態の変更	「vary」

query request

query request コマンドは、リクエストのステータスを表示します。

形式

```
query request request_id... | all
```

オプション

- *request_id* | *all*

クエリーするリクエストまたはすべてのリクエストの場合は `all` を指定します。

使用方法

`query request` コマンドは、次の形式でリクエストのステータスを表示するために使用します。

ここでは:

- `request_id`

ACSL S リクエスト識別子です。

- `command`

リクエスト識別子に対応する ACSLS コマンドです。

- `status`

次のいずれかのリクエストのステータスです。

- `Current`

ACSL S はリクエストを処理しています。

- `Pending`

リクエストは処理を待機しています。

- `Not found`

指定されたリクエストは有効な ACSLS リクエストではありません。

例

- リクエスト 33179 のステータス情報を表示するには:

```
query request 33179
```

- 現在および保留中のすべてのリクエストを表示するには:

```
query request all
```

関連項目:

情報の内容	参照先
コマンドの取り消し	「cancel」

query scratch

`query scratch` コマンドは、アクセス日の昇順でソートされた、プール内のスクラッチカートリッジのステータスを表示します。アクセス日がもっとも早いカートリッジがリストの先頭に、もっとも最近使用されたカートリッジがリストのもっとも下にあります。最初の列のプール ID は、特定の順序で表示されません。アクセス制御経由で許可されるカートリッジだけが表示されます。

形式

```
query scratch pool_id... | all
```

オプション

```
pool_id | all
```

クエリーするスクラッチプールまたはすべてのプールの場合は `all` を指定します。プール 0 は共通プールです。

使用方法

`query scratch` コマンドは、次の形式でプール内のスクラッチカートリッジのステータスを表示するために使用します。

ここでは:

- `pool_id`

指定されたスクラッチプールです。

- `vol_id`

スクラッチカートリッジの識別子です。

- `cell_id`

カートリッジを格納するストレージセルです。

- `status`

カートリッジの場所です。

- `home`

カートリッジはストレージセル内にあります。

- `in drive`

カートリッジはドライブ内にあります。

- *in transit*

カートリッジは移動中です。

- *media_type*

カートリッジのメディアタイプ (3480、3490E、DD3D、DLTIV) です。

注:

query scratch 出力には、カートリッジのステータスが不在か、取り出し済みか、見つからないスクラッチカートリッジは含まれません。

プール別にソートされたカートリッジを表示するには、プールごとに連続して *query scratch* コマンドを発行します。または、*display volume* コマンドを発行して、情報を表示し、プール別にそれをソートできますが、もっとも最近使用されたスクラッチカートリッジが取得される保証はありません。

例

- スクラッチプール 29015 内のスクラッチカートリッジのステータス情報を表示するには:

```
query scratch 29015
```

- すべてのスクラッチプール内のスクラッチカートリッジのステータス情報を表示するには:

```
query scratch all
```

関連項目:

情報の内容	参照先
スクラッチプールの作成または変更	「define pool」
空のスクラッチプールの削除	「delete pool」
カートリッジスクラッチ属性の設定またはクリア	「set scratch」

query server

query server コマンドは、ACSL5 およびライブラリのステータスを表示します。

形式

```
query server
```

オプション

なし。

使用方法

`query server` コマンドは、次の形式で ACSLS のステータスおよびライブラリを表示するために使用します。

ここでは:

- *Identifier*

空白です

- *state*

次のいずれかの ACSLS の状態です。

- *idle*

ACSLS はアイドルです (リクエストを処理していません)。

- *idle pending*

ACSLS は、現在および保留中のリクエストを処理し、新しいリクエストを拒否し、その後アイドルになります。

- *recovery*

ACSLS が初期化中 (run 状態になる) であるか、またはエラーからの回復中です。ACSLS はリクエストを処理していません。

- *run*

ACSLS は実行中 (リクエストの処理中) です。

- *count*

ライブラリ内の空きストレージセルの数です。

- *n*

ライブラリリソースを必要とする各コマンド (*audit*、*mount*、*dismount*、*enter*、および *eject*) の現在 (C) と保留中 (P) の ACSLS リクエストの数です。

ヒント: `query server` リクエストを取り消すことはできません。

例

- サーバーのステータス情報を表示するには:

```
query server
```

関連項目:

情報の内容	参照先
ライブラリコンポーネントの状態の変更	「 vary 」
リクエストのステータスの表示	「 query request 」

query volume

`query volume` コマンドはカートリッジの場所を表示します。

形式

```
query volume vol_id... | all
```

オプション

- `vol_id | all`

クエリーするボリュームまたはすべてのボリュームの場合は `all` を指定します。

使用方法

`query volume` コマンドは、次の形式でボリュームの場所を表示するために使用します。

ここでは:

- `vol_id`

ボリューム識別子です。

- `status`

カートリッジの場所です。

- `home`

カートリッジはストレージセル内にあります。

- `in drive`

カートリッジはドライブ内にあります。

- *in transit*
カートリッジが移動中であるか、または見つかりません。
- *absent*
カートリッジが見つかりません。
- *ejected*
カートリッジがライブラリから取り出されました。
- *location*
次のいずれかとして場所を指定します。
 - *status* が *home* の場合、*location* はストレージセル識別子です。
 - *status* が *in transit* の場合、*location* は、セル識別子またはドライブ識別子です。
 - *status* が *in drive* の場合、*location* はドライブ識別子です。
- *media_type*
ボリュームのメディアタイプ (3480、3490E、DD3D、または DLTIV など) です。

例

- ボリューム 2903B のステータス情報を表示するには:

```
query volume 2903B
```

- ライブラリ内のすべてのボリュームを表示するには:

```
q volume all
```

関連項目:

情報の内容	参照先
その他のボリューム情報	「ログインボリューム統計レポートの作成」
display コマンド	「display コマンドオプションの使用」 および 「display コマンドオプションの使用」
ドライブへのデータボリュームのマウント	「mount」
ドライブまたはボリュームのロックステータスの表示	「query lock」
指定されたデータボリュームのメディア互換ドライブのステータスの表示	「query mount」
スクラッチプール属性の表示	「query pool」

情報の内容	参照先
ボリュームスクラッチ属性の設定またはクリア	「set scratch」
ライブラリコンポーネントの状態の変更	「vary」

set コマンド

`set` コマンドは、さまざまなライブラリコンポーネントの各種属性を設定します。形式、オプション、使用方法、およびメッセージなどの各 `set` コマンドの詳細については、次のセクションを参照してください。

形式

次に `set` コマンドの一般的な形式を示します。

```
set type [off | subtype] [*] identifier...
```

関連項目:

情報の内容	参照先
コマンドの取り消し	「cancel」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

`set` コマンドが成功した場合、次のメッセージが表示されます。

```
Set: Set completed, Success.
```

特定の成功メッセージについては、各 `set` コマンドを参照してください。

中間メッセージ

なし。

エラーメッセージ

特定のメッセージについては、各 `set` コマンドを参照してください。

表示領域メッセージ

特定のメッセージについては、各 `set` コマンドを参照してください。

set cap mode

`set cap mode` コマンドは、CAP の挿入モードを設定します。

形式

```
set cap mode cap_mode cap_id
```

オプション

- `cap_mode`

手動または自動 CAP モードを指定します。

- `cap_id`

CAP 識別子を指定します。

優先度が設定されていないかぎり、アスタリスク (*) を含む CAP 識別子を指定できません。

使用方法

`set cap mode` コマンドは、CAP の挿入モードを次のいずれかのモードに設定するために使用します。

- `manual`

カートリッジを挿入する前に `enter` コマンドを入力する必要があります。

- `automatic`

最初に `enter` コマンドを入力せずにカートリッジを挿入できます。

注:

パーティション分割されたライブラリには、CAP のモードを自動的に設定することはできません。

ヒント: CAP のモードは、CAP の使用中に変更できません。つまり、手動または自動挿入操作中にドアが開いている場合、`enter` 操作が完了するまで、そのモードを変更できません。

例

- CAP 0,3,1 を手動モードに設定するには:

```
set cap mode manual 0,3,1
```

- CAP 0,3,1 を自動モードに設定するには:

```
set cap mode automatic 0,3,1
```

関連項目:

情報の内容	参照先
ラベル付きカートリッジをライブラリに挿入するための CAP (手動モード) の準備	「enter」
CAP ステータスの表示	「query cap」
ライブラリコンポーネントの状態の変更	「vary」
ラベルのないカートリッジをライブラリに挿入するための CAP の準備	「venter」

コマンド領域メッセージ

このセクションでは、コマンドメッセージについて説明します。

成功メッセージ

- *Set: CAP cap_id, mode changed to cap_mode.*

説明: ACSLS は指定された CAP のモードを変更しました。

変数:

- *cap_id* はモードを変更する CAP です。
- *cap_mode* は CAP の新しい挿入モードです。

中間メッセージ

なし。

エラーメッセージ

- *Set: CAP cap_id Set failed, Incorrect attribute.*

説明: 無効な CAP モードが指定されたため、ACSLs は指定された CAP のモードを変更できません。

変数: *cap_id* はモードを変更しない CAP です。

- *CAP cap_id: Automatic mode.*

説明: ACSLS は指定された CAP のモードを自動に変更しました。

変数: *cap_id* はモードを変更する CAP です。

- CAP *cap_id*: *Manual mode*.

説明: ACSLS は指定された CAP のモードを手動に変更しました。

変数: *cap_id* はモードを変更する CAP です。

set cap priority

set cap priority コマンドは、CAP の自動選択の優先度を設定します。

形式

```
set cap priority cap_priority cap_id
```

オプション

- *cap_priority*

CAP の優先度を指定します。有効な値は 0 - 16 です。ここで、16 がもっとも優先度が高いです。すべての CAP は最初に 0 の優先度になり、ACSLS はその CAP を自動的に選択しないことを意味します。

注:

AEM オプションを使用する場合。少数の挿入と取り出しには AEM を使用しないことが必要になるので、最大の CAP 優先順位は 1 になります。これは *audit*、*enter*、または *eject* で CAP ID がアスタリスクでワイルドカード化されている場合に、AEM が選択されることを防ぐために役立ちます。AEM の詳細については、「[AEM の使用](#)」を参照してください。

- *cap_id*

CAP 識別子を指定します。特定の CAP を指定する必要があり、すべての CAP に同じ優先度を設定するアスタリスク (*) を指定することはできません。

使用方法

set cap priority コマンドは、CAP の自動選択の優先度を設定するために使用します。

CAP リクエストで CAP ID にアスタリスク (*) を指定した場合、ACSLS は、リクエストに指定された ACS ごとに、ゼロ以外の最高の優先度を持つ、使用可能な CAP を自動的に選択します。

例

- CAP 0,3,1 に優先度 16 を割り当てるには:

```
set cap priority 16 0,3,1
```

関連項目:

情報の内容	参照先
ライブラリカートリッジの実際のインベントリに一致させるための ACSLS データベースの更新	「audit」
ライブラリからのカートリッジの取り出し	「eject」
ラベル付きカートリッジをライブラリに挿入するための CAP (手動モード) の準備	「enter」
CAP ステータスの表示	「query cap」
ライブラリコンポーネントの状態の変更	「vary」
ラベルのないカートリッジをライブラリに挿入するための CAP の準備	「venter」

コマンド領域メッセージ

ライブラリカートリッジの実際のインベントリに一致させるために ACSLS データベースを更新します。

成功メッセージ

- *Set: CAP cap_id, priority changed to cap_priority.*

説明: ACSLS は指定された CAP の優先度を変更しました。

変数:

- *cap_id* は優先度を変更する CAP です。
- *cap_priority* は新しい CAP の優先度です。

中間メッセージ

なし。

エラーメッセージ

- *Set: CAP cap_id Set failed, Incorrect attribute.*

説明: 無効な CAP 優先度が指定されたため、ACSL S は指定された CAP のモードを変更できません。

変数: *cap_id* は優先度を変更しない CAP です。

表示領域メッセージ

なし。

set clean

set clean コマンドはクリーニングカートリッジ属性を設定します。

すべての最新のクリーニングカートリッジでは、*audit*、*enter*、または *Cartridge Recovery* によってクリーニングカートリッジが追加されたときに、クリーニングカートリッジ属性が自動的に設定されます。これには、クリーニングカートリッジの *max_usage* の設定が含まれます。

形式

```
set clean max_usage | off vol_id | volrange
```

オプション

- *max_usage | off*

ACSL S がドライブをクリーニングするカートリッジの選択を停止するまでに、クリーニングカートリッジが使用される回数を指定します。*off* は、ACSL S がカートリッジを選択せず、カートリッジをデータカートリッジとして再定義することを指定します。

注:

データカートリッジ専用予約されているメディアタイプに、最大クリーニング使用回数を指定できません。クリーニングカートリッジのみであるメディアタイプにクリーニング *off* を設定できません。

- *vol_id | volrange*

クリーニングカートリッジまたはカートリッジの範囲を指定します。

使用方法

set clean コマンドは、ACSL S がクリーニングカートリッジを選択する回数を設定するために使用します。さらに、*set clean* は、カートリッジのクリーニングカートリッジ属性を *off* に設定

するためにも使用します。これは、データカートリッジをクリーニングカートリッジと誤って定義した場合に実行します。

例

- クリーニングカートリッジ *CLN108* - *CLN112* の最大使用回数を 10 に設定するには:

```
set clean 10 CLN108-CLN112
```

- クリーニングカートリッジ属性 *off* を設定し、カートリッジ *HRR234* - *HRR244* をデータカートリッジとして再定義するには:

```
set clean off HRR234-HRR244
```

関連項目:

情報の内容	参照先
ドライブのクリーニングのガイドラインと手順	「LSM の設定」
クリーニングカートリッジのステータスの表示	「query clean」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- *Set: volume vol_id is a cleaning cartridge.*

説明: ACSLS はクリーニングカートリッジがドライブのクリーニングに有効であることを指定しました。

変数: *vol_id* は有効なクリーニングカートリッジです。

- *Set: volume vol_id is not a cleaning cartridge.*

説明: ACSLS はクリーニングカートリッジがドライブのクリーニングに有効でないことを指定しました。

変数: *vol_id* は無効なクリーニングカートリッジです。

中間メッセージ

なし。

エラーメッセージ

- *Set: Clean vol_id Set failed, Incorrect attribute.*

説明: 指定されたカートリッジがクリーニングカートリッジでないため、ACSL S はクリーニング属性を変更できません。

変数: *vol_id* は、データまたはスクラッチカートリッジです。

表示領域メッセージ

なし。

set lock

set lock コマンドは、ロック ID を設定します。

形式

```
set lock lock_id
```

オプション

- *lock_id*

ロック ID を指定します。有効なロック ID は 0 - 32767 です。

使用方法

set lock コマンドは、ロック ID を設定したり、変更したりするために使用します。ロック ID が現在のロック ID に一致しないドライブまたはカートリッジのロックを解除する場合、次のように現在のロック ID を変更します。

- 指定されたドライブまたはカートリッジのすべてのロックをクリアするには、ロック ID を 0 に設定して、*clear lock* コマンドを入力します。
- ドライブまたはカートリッジのアクティブなロックを解除するには、ロック ID を、ロックされているコンポーネントのロック ID に設定して、*unlock* コマンドを入力します。

注:

lock コマンドを入力して、カートリッジまたはドライブをロックし、ロック ID がゼロ (0) の場合、ACSL S はカートリッジまたはドライブにロック ID を割り当ててから、ロック ID をカートリッジまたはドライブのロック ID に変更します。*set lock* コマンドを使用して、ロック ID を設定してから、*lock* コマンドを使用して、*set lock* コマンドで設定したロック ID でカートリッジまたはドライブをロックすることはできません。

例

- 現在のロック ID を新しいロック ID 354 に変更するには:

```
set lock 354
```

関連項目:

情報の内容	参照先
指定したドライブまたはカートリッジのすべてのアクティブなロックまたは保留中のロックの解除	「clear lock」
ドライブとカートリッジのロック	「lock」
ドライブまたはカートリッジのロックステータスの表示	「query lock」
ロック ID またはユーザー ID の表示	「show」
アクティブなロックの解除	「unlock」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- Set: Changed lock identifier from previous_lock_id to lock_id.*

説明: ACSLS はロック ID を変更しました。

変数:

- *previous_lock_id* は以前のロック ID です。
- *lock_id* は新しいロック ID です。

中間メッセージ

なし。

エラーメッセージ

なし。

表示領域メッセージ

なし。

set owner

`set owner` コマンドは、ボリュームの所有権を設定します。

形式

```
set owner owner_id volume vol_id | volrange
```

オプション

- `owner_id`

所有者の識別子を指定します。この値は引用符 (") で囲む必要があります。

- `volume vol_id | volrange`

ボリュームまたはボリュームの範囲を指定します。

使用方法

`set owner` コマンドは、ボリュームの所有権を設定するために使用します。`cmd_proc` から `set owner` を入力します。クライアントアプリケーションからボリュームの所有権を設定することはできません。

例

- ボリューム YUMA06 の所有権を割り当てるには:

```
set owner "cray" volume YUMA06
```

注:

ボリューム YUMA06 のボリューム所有権を削除するには:

```
set owner:"" volume 0YUMA06
```

コマンド領域メッセージ

なし。

表示領域メッセージ

なし。

set scratch

`set scratch` コマンドはボリュームのスクラッチ属性を設定またはクリアし、ボリュームをプールに割り当てます。

また、`watch_vols` ユーティリティーを使用して、`audit`、`enter`、または `Cartridge Recovery` によってスクラッチカートリッジが追加または再アクティブ化されたときに、自動的にスクラッチカートリッジ属性を設定することもできます。

形式

```
set scratch [off] pool_id vol_id | volrange
```

オプション

- `off`

ボリュームがデータカートリッジであることを指定します。

- `pool_id`

カートリッジのスクラッチプールを指定します。データカートリッジをその現在のプールに再割り当てするにはアスタリスク (*) を指定します。

- `vol_id | volrange`

ボリュームまたはボリュームの範囲を指定します。

使用方法

`set scratch` コマンドはボリュームのスクラッチ属性を設定またはクリアし、ボリュームをプールに割り当てるために使用します。

例

- ボリューム YUMA10-YUMA20 をスクラッチボリュームとして定義し、それらをスクラッチプール 5 に割り当てるには:

```
set scratch 5 YUMA10-YUMA20
```

- スクラッチボリューム YUMA10-YUMA15 をプール 10 に移動するには:

```
set scratch 10 YUMA10-YUMA15
```

- ボリューム YUMA16-YUMA20 を「スクラッチ解除」(スクラッチからデータに変更) し、それらを共通プール (プール 0) に移動するには:

```
set scratch off 0 YUMA16-YUMA20
```

- データボリューム YUMA16-YUMA20 をスクラッチし、それらを現在のプールで保持するには:

```
set scratch * YUMA16-YUMA20
```

関連項目:

情報の内容	参照先
スクラッチカートリッジステータスの表示	「query scratch」
スクラッチプール属性の表示	「query pool」
スクラッチプールの作成または変更	「define pool」
空のスクラッチプールの削除	「delete pool」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- *Set: volume vol_id in tape pool pool_id is a scratch cartridge.*

説明: ACSLS は指定されたデータカートリッジをスクラッチカートリッジに変更して、それをプールに割り当てます。

変数:

- 変数: *vol_id* は指定されたボリュームです。
- *pool_id* はボリュームの割り当て先のプールです。

- *Set: volume vol_id in tape pool pool_id is a data volume.*

説明: ACSLS は指定されたスクラッチカートリッジをデータカートリッジに変更しました。

変数: *vol_id* は、指定されたボリュームです。

中間メッセージ

なし。

エラーメッセージ

- *Set: Scratch vol_id Set failed, Incorrect attribute.*

説明: 指定されたカートリッジがクリーニングカートリッジであるため、ACSLs はスクラッチ属性を変更できません。

変数: *vol_id* はクリーニングカートリッジです。

表示領域メッセージ

- *Pool pool_id: low water mark warning.*

説明: 指定されたスクラッチプール内のカートリッジの数が下限しきい値以下です。

変数: *low_water_mark* は指定されたスクラッチプールの下限しきい値です。

- *Pool pool_id: high water mark warning.*

説明: 指定されたスクラッチプール内のカートリッジの数は上限しきい値以上です。

変数: *high_water_mark* は指定されたスクラッチプールの上限しきい値です。

show

show コマンドはロック ID またはユーザー ID を表示します。

形式

show type

オプション

- *type*

次のいずれかのタイプを指定します。

- *lock*

ロック ID。

- *user*

ユーザー ID。

使用方法

show コマンドはロック ID またはユーザー ID を表示するために使用します。

例

- リクエストの `user_id` を表示するには:

```
show user
```

- 現在の `lock_id` を表示するには:

```
show lock
```

関連項目:

情報の内容	参照先
指定したドライブまたはカートリッジのすべてのアクティブなロックまたは保留中のロックの解除	「clear lock」
ドライブとカートリッジのロック	「lock」
ドライブまたはカートリッジのロックステータスの表示	「query lock」
ロック ID の設定	「set lock」
アクティブなロックの解除	「unlock」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- *Show: Current lock identifier is lock_id*

説明: 現在のロック ID は `lock_id` です。

- *Show: User identifier is user_id*

説明: 現在のユーザー ID は `user_id` です。

中間メッセージ

なし。

エラーメッセージ

なし。

表示領域メッセージ

なし。

start

`start` コマンドは ACSLS リクエスト処理を開始します。

形式

`start`

オプション

なし。

使用方法

`start` コマンドは ACSLS を実行状態にして、ACSLM リクエスト処理を開始するために使用します。一般に `start` コマンドは、ACSLM がアイドル状態の場合に、リクエスト処理を再開するために使用します。

例

- ACSLM リクエスト処理を再開するには:

```
start
```

ACSLM が実行状態にある場合、`start` コマンドを入力しても効果はありません。

関連項目:

情報の内容	参照先
ACSLM の新しいリクエストの処理の停止	「idle」
ライブラリコンポーネントのステータスの表示	「query コマンド」
ライブラリコンポーネントの状態の変更	「vary」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- ACSLM Request Processing Started: Success.*

説明: ACSLS はリクエストの処理を開始しました。

中間メッセージ

なし。

エラーメッセージ

- *ACSLM Request Processing Not Started: status*

説明: ACSLS はリクエストの処理を開始しませんでした。

変数: *status* は失敗の理由です。一般的なステータスメッセージの詳細については、ACSLM メッセージを参照してください。

表示領域メッセージ

- *Server system running*

説明: ACSLS はリクエストの処理を開始しました。

switch lmu

Redundant Electronics (RE) またはデュアル LMU 構成では、`switch lmu` コマンドは、アクティブなライブラリコントローラ (LC) からスタンバイ LC に、または ACS のアクティブな LMU からスタンバイ LMU に ACS の管理を手動で切り替えます。次に注意してください。

- `switch lmu` は単一の RE ライブラリの ACS、SL3000 またはスタンドアロン SL8500 のみをサポートします。
- `switch lmu` はパーティション分割された SL8500 または SL3000 をサポートしていません。
- ACSLS は、ホスト/ライブラリインタフェース互換性レベル 11 以上がロードされた 9330 LMU のみに対してデュアル LMU 構成をサポートしています。同じマイクロコードレベルが両方の LMU にロードされている必要があります。

形式

```
switch lmu acs_id
```

オプション

- *acs_id*

ACS にライブラリ管理をアクティブからスタンバイ LC または LMU に切り替えるように指定します。

使用方法

`switch lmu` コマンドは、ライブラリのアクティブな LC または LMU からスタンバイ LC または LMU にライブラリ管理を手動で切り替えるために使用します。`switch lmu` コマンドを入力する前に、次のことを確認します。

- ACSLS は実行状態です
- 指定した ACS がオンライン状態または診断状態です
- 各 LC または LMU に対して少なくとも 1 つのポートがオンラインです

例

RE またはデュアル LMU 構成では、`switch lmu` コマンドは、ライブラリのアクティブな LC または LMU からスタンバイ LC または LMU に ACS 管理を切り替えるために使用します。次の RE 構成を想定します。

- ライブラリで、LC A はアクティブの役割で LC B はスタンバイの役割です。

ACSLs が LC A への通信を失いますが、LC B とはまだ通信できる場合、`switch lmu` を使用して LC B をアクティブな LC にします。

関連項目:

情報の内容	参照先
ACS ステータスの表示	「query acs」
ACS およびポートのステータスと詳細の表示	「query lmu」
ポートステータスの表示	「query port」
ACSLs およびライブラリのステータスの表示	「query server」
ACSLs リクエスト処理の開始	「start」
ライブラリコンポーネントの状態の変更	「vary」
冗長電子装置	「概要」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- Switch: Switch lmu completed for acs_id, Success.

説明: ACSLS はアクティブからスタンバイ LMU にライブラリ管理を切り替えました。

変数: *acs_id* は、LC または LMU が役割を切り替えた ACS です。

- *Switch: Switch lmu initiated for acs_id, Success.*

説明: ACSLS はアクティブからスタンバイ LC または LMU へのライブラリ管理の切り替えを開始しました。

変数: *acs_id* は、LC または LMU が役割を切り替えている ACS です。

- *Switch: Switch lmu failed, acs_id not found.*

説明: 無効な ACS を指定しました。

変数: *acs_id* は、*switch lmu* コマンドに指定した ACS 識別子です。

- *Switch: Switch lmu failed, acs_id is offline.*

説明: オフラインの ACS を指定しました。*vary* コマンドは、ACS をオンラインまたは診断モードに変更するために使用します。

変数: *acs_id* は、*switch lmu* コマンドに指定した ACS 識別子です。

- *Switch: Switch lmu failed, acs_id not configured for dual lmu.*

説明: アクティブおよびスタンバイ LC または LMU で構成されていない ACS を指定しました。次のいずれか:

- 指定された ACS を再構成してから、*switch lmu* コマンドを再入力します
- アクティブおよびスタンバイ LC または LMU で構成されているライブラリを指定する *switch lmu* コマンドを入力します。

変数: *acs_id* は、*switch lmu* コマンドに指定した ACS 識別子です。

- *Switch: Switch lmu failed, not communicating.*

説明: *switch lmu* スイッチはスタンバイ LC または LMU が通信していないため失敗しました。スタンバイ LC または LMU の通信の問題をチェックしてください。

- *Switch: Switch lmu failed, switch already active.*

説明: 指定された ACS に対して *switch lmu* が進行中です。*switch* が完了するまで待つてから、*switch lmu* コマンドを再入力します。

- *Switch: Switch lmu failed, port is offline.*

説明: オンラインのポートがない ACS を指定しました。各 LC または LMU に対して少なくとも 1 つのポートをオンラインに変更 (*vary*) します。

表示領域メッセージ

なし。

unlock

unlock コマンドは、指定されたドライブまたはカートリッジのアクティブなロック (現在のロック ID に関連付けられた) を解除するか、またはすべてのアクティブなロックを解除します。

形式

unlock type identifier... | all

オプション

- *type identifier*

ライブラリコンポーネントを指定します。次の表に、ロック解除できるコンポーネントの一覧を示します。

表13.6 unlock に有効なコンポーネント

ライブラリコンポーネント	タイプ	識別子
ドライブ	<i>drive</i>	<i>drive_id</i>
ボリューム	<i>volume</i>	<i>vol_id</i>

- *all*

すべてのアクティブなロックを指定します。

使用方法

unlock コマンドは、指定されたカートリッジおよびドライブのアクティブなロックを解除したり、すべてのアクティブなロックを解除したりするために使用します。コンポーネントの *lock ID* は、現在のロック ID に一致する必要があります。

使用されていないコンポーネントをロック解除すると、ACSL S は、コンポーネントの保留中のロックをチェックします。保留中のロックがある場合、ACSL S は保留中のロックのロック ID を使用して、コンポーネントをロックします。

注:

`unlock` コマンドは常にロック ID を 0 にリセットします。

例

- ドライブ 0,0,2,0 のアクティブなロックを解除するには:

```
unlock drive 0,0,2,0
```

- ロックされているすべてのカートリッジのアクティブなロックを解除するには:

```
unlock volume all
```

関連項目:

情報の内容	参照先
指定したドライブまたはボリュームのすべてのアクティブなロックまたは保留中のロックの解除	「clear lock」
ドライブおよびボリュームのロック	「lock」
ドライブまたはカートリッジのロックステータスの表示	「query lock」
ロック ID の設定	「set lock」
ロック ID またはユーザー ID の表示	「show」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- `Unlock: Unlock completed, Success.`
- `Unlock: drive drive_id unlocked.`

説明: ACSLS は指定されたドライブのロックを解除しました。

変数: `drive_id` はロック解除されたドライブです。

- `Unlock: Unlock completed, Success.`
- `Unlock: volume vol_id unlocked.`

説明: ACSLS は指定されたボリュームのロックを解除しました。

変数: `vol_id` は、ロック解除されたボリュームです。

中間メッセージ

なし。

エラーメッセージ

- *Unlock: Unlock of drive drive_id failed, status.*

説明: ACSLS はドライブのロックを解除できません。

変数: *status* は失敗の理由です。一般的なステータスメッセージの詳細については、ACSLs メッセージを参照してください。

- *Unlock: Unlock of volume vol_id failed, status.*

説明: ACSLS はボリュームのロックを解除できません。

変数: *status* は失敗の理由です。一般的なステータスメッセージの詳細については、ACSLs メッセージを参照してください。

表示領域メッセージ

なし。

vary

vary コマンドは、ライブラリコンポーネントの状態を変更します。

形式

vary type identifier...state [force]

オプション

- *type [identifier]*

ライブラリコンポーネントを指定します。次の表に、変更 (*vary*) できるコンポーネントの一覧を示します。

表13.7 *vary* に有効なコンポーネント

ACS コンポーネント	タイプ	識別子
ACS	<i>acs</i>	<i>acs_id</i>

ACS コンポーネント	タイプ	識別子
LSM	<i>lsm</i>	<i>acs_id,lsm_id</i>
CAP	<i>cap</i>	<i>cap_id</i>
ドライブ	<i>drive</i>	<i>drive_id</i>
ポート	<i>port</i>	<i>port_id</i>

- *state*

次のいずれかのデバイスの状態を指定します。*online* (クライアントと *cmd_proc* に対してオンライン)、*offline*、または *diagnostic* (*cmd_proc* のみに対してオンライン)。

ACS、LSM、CAP、またはドライブをオンライン、オフライン、または診断に変更 (*vary*) できます。ポートをオンラインまたはオフラインに変更 (*vary*) できます。デバイスの状態の詳細については、「[使用方法](#)」を参照してください。

- *force*

現在のロボットリクエストのみを処理したあとに、ACS、LSM、CAP、またはドライブを *offline* に変更します。

使用方法

vary コマンドは、ACS、LSM、CAP、ドライブ、またはポートの状態を変更するために使用します。次のセクションでは、デバイスの各状態がライブラリコンポーネントに与える影響について説明します。

- *vary offline*

次の表に、各 ACS コンポーネントの *vary offline* の結果を示します。コンポーネントをオフラインに変更し、それがまずオフライン保留状態になると、ACSLS はコンポーネントに対するすべてのアクティブおよび保留中のリクエストを処理し、新しいリクエストを拒否して、コンポーネントをオフラインに変更します。

表13.8 vary offline の結果

ACS コンポーネント	結果
ACS	ACS とサブコンポーネントがオフライン保留中になり、次にオフラインになります。LSM は ACS がオフラインになる前にオフラインになる必要があります。

ACS コンポーネント 結果

LSM	LSM はオフライン保留中になってから、オフラインになります。
CAP	CAP はオフライン保留中になってから、オフラインになります。
ドライブ	使用可能な場合、ドライブはただちにオフラインになります。使用中の場合、ドライブはオンラインのままになります。
ポート	ポートは次の場合にただちにオフラインになります。 <ul style="list-style-type: none"> • ACS にほかのオンラインのポートがあるか、または • ACS はオフラインです。 <p>それ以外の場合、ポートはオンラインのままになります。</p>

• *vary offline force*

次の表に、各 ACS コンポーネントの *vary offline force* の結果を示します。このオプションは、デバイスが拡張アクティビティーに参与している間 (監査時など) に、それをオフライン状態にする必要がある場合に役立ちます。

表13.9 vary offline force の結果

ACS コンポーネント 結果

ACS	現在のロボットリクエストのみが完了し、ACS およびサブコンポーネントはただちにオフラインになります。保留中のリクエストは破棄され、新しいリクエストは拒否されます。LSM は ACS がオフラインになる前にオフラインになる必要があります。
LSM	現在のロボットリクエストのみが完了し、LSM はただちにオフラインになります。保留中のリクエストは破棄され、新しいリクエストは拒否されます。
CAP	現在のロボットのリクエストのみが完了し、そのあとすぐに CAP がオフラインになります。保留中のリクエストは破棄され、新しいリクエストは拒否されます。
ドライブ	現在のロボットリクエストのみが完了し、ドライブはただちにオフラインになります。保留中のリクエストは破棄され、新しいリクエストは拒否されます。
ポート	無効です。

• *vary diagnostic*

vary diagnostic リクエストは、指定されたコンポーネントを診断状態 (*cmd_proc* に対してのみオンライン) にします。ACSL S はコンポーネントに対するすべてのアクティブおよび保留中のリクエストを処理し、新しいクライアントアプリケーションリクエストを拒否して、コンポーネントを診断状態に変更します。ACS の場合、そのすべての下位 LSM も診断に変更されます。

- *vary online*

次の表に、各 ACS コンポーネントの *vary online* の結果を示します。コンポーネントをオンラインに変更 (*vary*) し、それがまず回復状態になると、ACSL S はコンポーネントに対するすべてのアクティブおよび保留中のリクエストを処理し、新しいリクエストを拒否して、コンポーネントをオンラインに変更します。コンポーネントがオンラインになると、ACSL S はコンポーネントに対するすべてのリクエストを処理します。

表13.10 vary online の結果

ACS コンポーネント	結果
ACS	ACS がオフラインの場合、ACS と LSM は回復になってから、オンラインになります。 ACS が診断状態である場合、ACS と LSM はただちにオンラインになります。
LSM	LSM は回復になってから、オンラインになります。LSM がオフラインの ACS に接続されている場合、それをオンラインに変更することはできません。
CAP	CAP は回復に移行してから、オンラインになります。
ドライブ	ドライブは回復に移行してから、オンラインになります。
ポート	ポートはただちにオンラインになります。

例

- ドライブ 0,0,9,3 をオフラインに変更 (*vary*) するには:

```
vary drive 0,0,9,3 offline
```

- CAP 0,0,0 を診断に変更 (*vary*) するには:

```
vary cap 0,0,0 diagnostic
```

- LSM 0,1 を強制的にオフラインにする (*force lsm*) には:

```
vary lsm 0,1 offline force
```

注:

システムの IPL 中はこれらのコンポーネントの状態が変更されません。ACSLs のインストールまたは再構成によって、可能な限り、すべてのコンポーネントがオンライン状態になります。

同じライブラリの複数のパーティションが ACSLS によって管理されている場合、個別の *vary* コマンドを使用して、同じ CAP (異なる ACS 内にあると識別される) を変更します。

例: ACS 0 と ACS 1 が同じライブラリの 2 つのパーティションである場合、パーティションごと (および ACS ごと) に独立して CAP をオンラインまたはオフラインに変更 (*vary*) します。同じ CAP を 1 つのパーティション (ACS) に対してオンラインに変更し、別のパーティション (ACS) に対してオフラインに変更できます。

関連項目:

情報の内容	参照先
ライブラリコンポーネントのステータスの表示	「query コマンド」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- *Vary: type identifier Varied state*

説明: ACSLS は指定されたライブラリコンポーネントの状態を変更しました。

変数:

- *type* はライブラリコンポーネントタイプです。
- *identifier* はライブラリコンポーネント識別子です。
- *state* はライブラリコンポーネントの新しい状態です。

中間メッセージ

なし。

エラーメッセージ

- *In-transit volume recovery incomplete.*

説明: LSM またはその ACS がオンラインに変更される間に、LSM ロボットは移動中のボリュームを廃棄できません。ACS をオンラインに変更する場合は、ACS の状態はただちにオンラインに変更されますが、移動中のボリュームの回復を完了できないすべての LSM はオフラインのままになります。

- *Vary: Vary type identifier failed, Drive in use.*

説明: コマンドが使用中のドライブに関与していたため、ACSLs は指定されたライブラリコンポーネントの状態を変更できません。

変数:

- *type* はライブラリコンポーネントタイプです。
- *identifier* はライブラリコンポーネント識別子です。

- *Vary: Vary type identifier failed, Incorrect lockid*

説明: *drive lock_id* が *lock_id* と一致しないため、ACSLs は指定されたライブラリコンポーネントの状態を変更できません。

変数:

- *type* はライブラリコンポーネントタイプです。
- *identifier* はライブラリコンポーネント識別子です。

- *Vary: Vary type identifier failed, State unchanged.*

説明: コンポーネントがすでにリクエストされた状態であるため、ACSLs は指定されたライブラリコンポーネントの状態を変更できません。

変数:

- *type* はライブラリコンポーネントタイプです。
- *identifier* はライブラリコンポーネント識別子です。

- *Vary: Vary type identifier failed, Vary disallowed.*

説明: *vary diagnostic* リクエストはクライアントアプリケーションから発行されたため、ACSLs は指定されたライブラリコンポーネントの状態を変更できません。

変数:

- *type* はライブラリコンポーネントタイプです。
- *identifier* はライブラリコンポーネント識別子です。

- *Vary: Vary type identifier failed, Incorrect state.*

説明: 次を変更 (*vary*) しようとしたため、ACSLs は指定されたライブラリコンポーネントの状態を変更できません。

- LSM または ACS が診断またはオフライン状態の場合に、CAP をオンラインに。
- LSM または ACS がオフラインの場合に CAP を診断に。
- ACS が診断またはオフライン状態の場合に、LSM をオンラインに。
- ACS がオフラインの場合に LSM を診断に。

変数:

- *type* はライブラリコンポーネントタイプです。
- *identifier* はライブラリコンポーネント識別子です。
- *Vary: Vary type identifier failed, Vary in progress.*

説明: ACSLS は次の理由で、指定されたライブラリコンポーネントの状態を変更できません。

- CAP および CAP またはその LSM または ACS が一時的な (回復またはオフライン保留中) 状態にあります。
- LSM および LSM、その ACS、または下位 CAP が一時的な状態にあります。
- ACS および ACS、下位 LSM、または下位 CAP が一時的な状態にあります。
- ACS、LSM、または CAP の診断またはオンライン状態への変更で、*vary offline/force* リクエストによって試みが上書きされました。
- 回復の移行状態のドライブ。

変数:

- *type* はライブラリコンポーネントタイプです。
- *identifier* はライブラリコンポーネント識別子です。
- *Vary: Vary type identifier failed, State unchanged.*

説明: ACS、LSM、または CAP をオフラインに変更 (*vary*) しようとし、*vary offline/force* リクエストによって試行が上書きされたため、ACSLs は指定されたライブラリコンポーネントの状態を変更できません。

変数:

- *type* はライブラリコンポーネントタイプです。
- *identifier* はライブラリコンポーネント識別子です。
- *Vary: Vary type identifier failed, ACS online.*

説明: オンライン ACS 内の最後のポートをオフラインに変更 (*vary*) しようとしたため、ACSL S は指定されたライブラリコンポーネントの状態を変更できません。

変数:

- *type* はラインブラリコンポーネントタイプです。
- *identifier* はライブラリコンポーネント識別子です。
- *Vary: Vary type identifier failed, No ports online.*

説明: ACS に対してオンラインのポートがないため、ACSL S は指定されたライブラリコンポーネントの状態を変更できません。

変数:

- *type* はラインブラリコンポーネントタイプです。
- *identifier* はライブラリコンポーネント識別子です。

表示領域メッセージ

ACS、LSM、ドライブ、またはポートの状態が変わると次のいずれかのメッセージが表示されます。

- *ACS acs_id: state*
- *LSM lsm_id: state*
- *Drive drive_id: state*
- *Port port_id: state*

移動中のボリュームの回復時、カートリッジラベルが読み取りできないかまたは重複している場合、次のメッセージが表示されます。カートリッジは CAP に移動されます。

CAP cap_id.Remove volumes from CAP.

- *cap_id*

カートリッジを格納する CAP です。

venter

venter コマンドはラベルのないカートリッジをライブラリに挿入するために CAP を準備します。

注:

LS500 はメディアラベルのないカートリッジをサポートしていません。SL500、SL3000、および SL8500 ライブラリはラベルのないカートリッジをサポートしていません (*venter*)。その理由は、これらのライブラリが、*volser* ラベルのないメディアラベルを読み取れず、ACSL S が不明なメディアタイプのカートリッジを挿入 (*venter*) しないためです。

形式

venter cap_id vol_id

オプション

- *cap_id*

カートリッジを挿入する CAP を指定します。

- *vol_id*

挿入するカートリッジの仮想ラベルを指定します。

使用方法

venter コマンドは、見つからないか読み取れないラベルを持つカートリッジをライブラリに挿入するように CAP を準備するために使用します。*venter* コマンドで仮想ラベル (ボリューム ID) を指定し、これによってカートリッジ情報を ACSLS データベースに追加します。1 から 42 個のカートリッジを一度に挿入できます。

メディアタイプが「3480」でないかぎり、個別のメディアタイプラベルがないカートリッジを挿入 (*venter*) できません。

注意:

挿入されるカートリッジにメディアラベルがあり、かつカートリッジのメディアラベルが SCSI ライブラリに接続されたドライブと互換性がある場合に、ACSL S は、SL500 より前の SCSI/ファイバ接続された LSM に対してのみ *venter* コマンドをサポートしています。カートリッジのメディアタイプが 3480 でない限り、個別のメディアラベルのないカートリッジを挿入 (*venter*) できません。

venter コマンドは、見つからないか読み取りできないラベルを持つカートリッジを挿入するために使用します。ラベルがない、または読み取れないカートリッジは、ACSL S が管理できないため、LSM のドアを開いてこれらのカートリッジをストレージセル内に配置しないでください。*audit* 中に、ACSL S は見つからないか、読み取れないラベルがあり、仮想ラベルのないカートリッジを取り出します。

`venter` コマンドは、挿入するカートリッジのメディアタイプを指定するオプションを提供しません。混在メディア環境では、ACSL5 は事実上挿入されたカートリッジのドライブ/メディアの非互換性を防止できません。

例

- 仮想ラベルをボリューム `MAINT1` および `MAINT2` に割り当てるには、`CAP0,2,2` を介してこれらのボリュームを挿入 (`enter`) します。

```
venter 0,2,2 MAINT1 MAINT2
```

関連項目:

情報の内容	参照先
コマンドの取り消し	「cancel」
スクラッチプールの作成または変更	「define pool」
ライブラリからのカートリッジの取り出し	「eject」
ラベル付きカートリッジをライブラリに挿入するための CAP (手動モード) の準備	「enter」
ドライブへのスクラッチカートリッジのマウント	「mount *」
ドライブ上のデータカートリッジのマウント	「mount」
CAP ステータスの表示	「query cap」
CAP モード (手動または自動) の設定	「set cap mode」
CAP 選択の優先度の設定	「set cap priority」

コマンド領域メッセージ

このセクションでは、コマンド領域メッセージについて説明します。

成功メッセージ

- *Venter: Enter complete, nn cartridges entered.*

説明: ACSLS は表示された数のカートリッジを挿入しました。

変数: *nn* は挿入されたカートリッジの合計数です。

- *Venter: vol_id Entered through cap_id*

説明: ACSLS は指定された仮想カートリッジを挿入しました。

変数:

- *vol_id* は仮想カートリッジラベルです。
- *cap_id* はカートリッジが挿入された CAP です。

中間メッセージ

なし。

エラーメッセージ

- *Venter: vol_id Enter failed, status*

説明: ACSLS は指定された仮想カートリッジを挿入しませんでした。

変数:

- *vol_id* は仮想カートリッジラベルです。
- *status* は ACSLS によって返されたリクエストの変換された完了ステータスです。

表示領域メッセージ

- *CAP cap_id: Place cartridges in CAP.*

説明: CAP はカートリッジを挿入する準備ができています。CAP を開き、カートリッジを挿入 (*enter*) します。

変数: *cap_id* は、カートリッジの挿入に使用される CAP です。

- *CAP cap_id: Unknown media type label.*

説明: メディアタイプが不明なため、1 つ以上のカートリッジを挿入できません。

変数: *cap_id* は、カートリッジの挿入 (*enter*) に使用される CAP です。CAP を開き、カートリッジを取り外します。

第14章 display コマンドリファレンス

この章では、*display* コマンドを使用する方法について学習します。

SQL の観点から考え、オプションに目を向けると、*display* コマンドの能力と柔軟性を明確に理解できます。*display* コマンドは、ACSL S データベースからの情報を報告するための強力なツールです。SQL SELECT 文と同様に、*display* では次を指定できます。

- 表示するデータベーステーブル (ACSL S オブジェクト)。
- ワイルドカード。
- オブジェクトを選択するために満たす必要のある条件。
- 報告するフィールドとそれらを報告する順序。
- 出力が表示されるソート順序。
- 選択条件を満たすオブジェクトの数のカウントのみを表示すること。

この結果は、表示するものとその表示方法のカスタマイズに使用できる *query* 機能です。*display* コマンドを使用すると、ACSL S *query* コマンドを使用して入手できるものよりもはるかに多くの情報を入手できます。たとえば、*display volume* ではカスタムの *volrpt*、*display lsm* を使用しないと入手できなかったボリューム情報にアクセスでき、*display drive* では LSM とテープドライブのシリアル番号が提供されます。

display コマンドは、

- `[display cap]`

選択されたオプションに基づいて特定の CAP 情報を表示します。

- `[display cell]`

選択されたオプションに基づいて特定のセル情報を表示します。

- `[display drive]`

選択されたオプションに基づいて特定のドライブ情報を表示します。

- `display lock`

ユーザー ID に基づいて特定のロック情報を表示します。

- `display lsm`

選択されたオプションに基づいて特定の LSM 情報を表示します。

- `display panel`

選択されたオプションに基づいて特定のライブラリパネル情報を表示します。

- `display pool`

選択されたオプションに基づいて特定のスクラッチプール情報を表示します。

- `display port`

選択されたオプションに基づいて特定のポート情報を表示します。

- `display volume`

選択されたオプションに基づいて特定のボリューム情報を表示します。

display コマンドオプションの使用

`display` コマンドは、さまざまなライブラリコンポーネントのさまざまなフィールドの情報を表示します。`display` コマンドで、出力を表示する順序と形式を選択できます。

表示される情報には複数の列と長い行を含めることができます。行が折り返されないようにするために、表示に使用する端末ウィンドウのサイズを変更できます。`cmd_proc -lq` ウィンドウはおそらく最適な選択肢になります。

`display` コマンドと一緒にワイルドカード、数値または英字範囲、および選択リストを使用できます。

ワイルドカード文字の使用

- ワイルドカード文字、* は、特定の `display` オペランドに対して、複合 `display` オペランド内の 1 つまたは複数のサブフィールドに対して、またはボリューム ID の先頭または末尾で、代入のために使用できます。1 つまたは複数の文字に一致する必要があります。
- 多くの ACSLS ライブラリコンポーネントには、複数のフィールドで構成された主キーが含まれます。

たとえば、主キーは次のとおりです。

- LSM では *acs*、*lsm*
- ドライブでは *acs*、*lsm*、*panel*、*drive*
- 複数のフィールドが含まれる主キーにワイルドカードを使用する場合は、キーのすべてのフィールドを1つのアスタリスク (*) として指定することも、個々のキーフィールドにワイルドカードを指定することもできます。いくつかのキーフィールドに (*)、その他にワイルドカードを指定する場合は、キーフィールドのすべてを指定してください。
- *vol_id* にはアスタリスク (*) を使用しないでください。*vol_id* 内の1文字に一致させるには、アンダースコア (_) を使用します。アンダースコアは位置指定であり、1つの文字を表します。これは、複数の文字を表すために *vol_id* で複数回使用されることがあります。

例

* 文字を使用した有効なワイルドカードの例:

- *display drive ** (すべてのドライブを表示します)
- *display drive 0,1,*,** (LSM 0,1 のすべてのドライブを表示します)
- *display volume *100* (100 で終わるすべてのボリュームを表示します)
- *display drive * -volume RB0001* (ボリューム RB0001 が含まれているドライブを表示します)

* 文字を使用した無効なワイルドカードの例:

- *display drive 0,1,** (LSM 0,1 のすべてのドライブを選択するときは、パネルとドライブの両方の位置に対してアスタリスクを指定する必要があります)
- *display cap *** (すべての CAP を選択するときは、1つのアスタリスクのみ許可されます)
- *display volume 1*111* (埋め込みアスタリスクは許可されません。*display volume 1_111* を使用してください)。

範囲の使用

- 範囲は「-」(ハイフンまたはダッシュ) で示されます。
- 範囲の使用ルールは次のとおりです。
 - ボリュームホーム LSM を除き、負の数は許可されません。
 - <n および >n は許可されます。<n は n より小さいすべての数値です。>n は n より大きいすべての数値です。
 - 複数の数値または範囲は、空白文字で区切って使用できます。

- 。 ボリューム範囲には英数字を指定できます。

例

有効な範囲の例は次のとおりです。

- `display volume * -drive 0,1,2,2-5` (acs 0、lsm 1、パネル 2、ドライブ 2 から 5 のすべてのボリュームを表示します)
- `display lsm * -state online -free_cells >50 -type 9730` (50 個より多くの空きセルがあるオンライン状態のすべての 9730 lsm を表示します)

無効な範囲の例は次のとおりです。

- `display drive 0,1,1,1-*` (* は範囲では有効ではありません)
- `display volume * -drive 0,1,?,1-5` (? は範囲では有効ではありません)

形式

```
display type arg ... [ -selection ... ] [ [ -c ] | [ -f field ... ]  
[ -s sort_field ... ] [ -n n ] ]
```

オプション

- `type`

表示するオブジェクトを指定します。有効なタイプは、cap、cell、drive、lock、LSM、panel、pool、port、および volume です。

- `arg`

(引数) オブジェクトタイプの識別子を指定します。1 つ以上の識別子と一致するオブジェクトのみが返されます。

- `-selection`

追加の選択基準を指定します。選択するには、コンポーネントが各選択条件の少なくとも 1 つの選択識別子に一致する必要があります。

- `-c`

(カウント) このリクエストで `arg` および `selection` 条件を満たすオブジェクトの数のみを表示します。このオプションは、`-f field`、`-s sort_field`、および `-n n` オプションと同時に指定できません。

- *-f field*

返される情報のフィールドと、それらが選択された各オブジェクトに対して返される順序を指定します。それぞれの `display` タイプには、使用可能、キー、およびデフォルトのフィールドのリストがあります。`-f` を指定する場合は、キーフィールドであるかすでに指定されている場合を除いて、デフォルトのフィールドは表示されません。`-f` を指定しない場合は、デフォルトのフィールドが返されます。各タイプのキーフィールドは、通常は最初に表示されます。ただし、`-f` オプションのあとに指定することによって、キーフィールドが表示される順序を変更できます。

フィールドのリストは空白文字で区切ります。

- *-s sort_field*

選択したフィールドによって返されたオブジェクトを指定した順序でソートします。各 `display` コマンドのデフォルトのソート順序については、各コマンドの下で説明します。ソートは内部データベースの値に基づいていて、常に英数字順で表示されるとは限りません。

タイプに対して返されるすべてのフィールドが有効な `sort_field` です。

- *-n n*

表示するオブジェクトの最大数を指定します。

80 文字より長い行を表示するときに行モードで `cmd_proc` を起動する

`curses` モードで起動した `cmd_proc` コマンドでは 80 文字より長い行を表示できず、80 文字より長い行を表示しようとする `cmd_proc` ウィンドウがハングします。

このため、多数のオプションのフィールドを表示するときは、`cmd_proc` を行モードで起動する (`-1` オプションを指定して) ことをお勧めします。例: `cmd_proc -1` として起動された `cmd_proc` を使用した `display drive * -f volume type state serial_num wwn`。

コマンド

このセクションでは、`display` コマンドについて説明します。

`display cap`

`display cap` コマンドは、選択されたオプションに基づいて特定の CAP 情報を表示します。

注:

SL3000 および SL8500 (4.70 以降のファームウェア) の場合。SL3000 に追加モジュールが追加されたり追加の CAP がインストールされたりするときに、CAP ID が変更されないようにするため、ライブラリはその構成情報内で可能性のあるすべての CAP の位置を返します。インストールされていない CAP は、CAP ステータスに「インストールされていない」として報告されます。

ACSLs は、実際に CAP がインストールされている CAP 位置のデータベースレコードのみを追加します。

注:

display の選択条件の指定と表示するデータの選択についての詳細は、「[display コマンドオプションの使用](#)」を参照してください。

形式

```
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 ] ]
```

フィールド

display cap のフィールドは次のとおりです。

- キーフィールド: *acs*、*lsm*、*cap*
- デフォルトのフィールド
ド:*acs*、*lsm*、*cap*、*status*、*priority*、*state*、*mode*、*size*、*availability*
- 使用可能なフィールド:*acs*、*lsm*、*cap*、*status*、*priority*、*state*、*desired_state*、*mode*、*size*、*condition*、*availability*

オプション

- *cap_id*

表示する CAP を指定します。*cap_id* の形式は、*acs*、*lsm*、*cap* です。

ワイルドカード文字、* は、任意の *cap_id* サブフィールドで使用することも、すべてのサブフィールドを表すために使用することもできます。ただし、すべての *cap_ids* (例: `display cap *` または `display cap *, *, *`) を表す場合、追加の *cap_ids* (数値または * のいずれか) は許可されません。数値の範囲は *cap_id* のすべてのサブフィールドに適用されます。

- *availability cap_availability*

追加の選択条件として1つまたは複数の *cap* の可用性を指定します。*CAP* の可用性は、*lib_not_partn*、*cap_shared*、*cap_dedicated*、*dedicated_other*、*not_installed* です。

dedicated_other または *not_installed* の *CAP* は *ACSL*S データベースに定義されません。これらの値は、*CAP* の可用性が変更されている (構成後に) 場合にのみ表示されます。再構成すると、これらの *CAP* が削除されます。

- *-status cap_status*

追加の選択条件として1つまたは複数の *CAP* のステータスを指定します。*CAP* のステータスは、*audit*、*available*、*eject*、*enter*、*insert_magazines* です。

- *-priority cap_priority*

1つまたは複数の *CAP* の優先順位を指定します。数値範囲ルールが適用されます。*CAP* の数値範囲は、0-16 です。

- *-state cap_state*

1つまたは複数の *CAP* の状態を指定します。*CAP* の状態は、*diagnostic*、*offline*、*offline_pending*、*online*、*recovery* です。

- *-manual*

手動モードの *CAP* を選択します。

- *-automatic*

自動モードの *CAP* を選択します。

- *-condition cap_condition*

1つまたは複数の選択する *CAP* の状況を指定します。有効な状況は、*inoperative*、*maint_required*、または *operative* です。

注:

inoperative または *maint_required* の状況は、ドライブ、*LSM*、または *ACS* がオンラインで変更されたときにのみクリアされます。そのため、*inoperative* または *maint_required* の *CAP* の状況は正確ではないことがあります。

- *-c*

(カウント) このリクエストで `arg` および `selection` 条件を満たすオブジェクトの数のみを表示します。このオプションは、`-f field`、`-s sort_field`、および `-n n` オプションと同時に指定できません。

- `-f field`

返される情報のフィールドと、それらが選択された各オブジェクトに対して返される順序を指定します。それぞれの `display` タイプには、使用可能、キー、およびデフォルトのフィールドのリストがあります。`-f` を指定する場合は、キーフィールドであるかすでに指定されている場合を除いて、デフォルトのフィールドは表示されません。`-f` を指定しない場合は、デフォルトのフィールドが返されます。各タイプのキーフィールドは、通常は最初に表示されます。ただし、`-f` オプションのあとに指定することによって、キーフィールドが表示される順序を変更できます。

フィールドのリストは空白文字で区切ります。

次は、`query cap` では報告されない新しい CAP のフィールドです。

- `desired_state`

ACS、ポート、LSM、ドライブ、または CAP の `desired state` は、コンポーネントに必要な可用性です。ACSL S は、明示的な `vary` 操作が実行されるときに目的の状態を設定します。(これは `cmd_proc` または `ACSAPI client` コマンドからの `vary` です。ライブライブラリのステータス変更によって ACSLS が内部的に生成する `vary` ではありません。)

ACS、ポート、LSM、ドライブ、または CAP の現在の状態 (「state」として指定されています) は、目的の状態によって制限されるコンポーネントの現在の可用性です。ドライブがオンラインで変更されたときに、その目的の状態がオンラインになります。ただし、ドライブが作動不能か、オフラインまたは準備できていない LSM 内にあるために、その現在の状態がオフラインである可能性もあります。ACS、ポート、LSM、ドライブ、または CAP の現在の状態は、`query` および `display` コマンドの結果でコンポーネントの「state」としても参照されます。

可用性に基づいた CAP の現在の状態は、現時点では維持されません。

論理ライブラリおよび論理ライブラリ内のテープドライブの可用性は、基となる物理ライブラリと論理ライブラリの両方に設定した目的の状態によっても左右されます。物理ライブラリと論理ライブラリの両方の目的の状態がオンラインの場合、論理ライブラリと論理テープドライブの現在の状態は、基となる物理ライブラリとドライブの現在の情報を反映します。

- `-s sort_field`

選択したフィールドによって返されるオブジェクトを、指定した順序でソートします。ソートは内部データベースの値に基づいていて、常に英数字順で表示されるとは限りません。

タイプに対して返されるすべてのフィールドが有効な `sort_field` です。

- `-n n`

表示するオブジェクトの最大数を指定します。

例

- すべての CAP データを表示 (`display`) するには:

```
display cap *
```

- ACS 1、LSM 1、CAP 1 の CAP データを表示 (`display`) するには:

```
display cap 1,1,1
```

- LSM 1、0 にすべての手動モードの CAP を表示 (`display`) するには:

```
display cap 1,0,* -manual
```

- オフラインの保留状態のすべての CAP を表示 (`display`) するには:

```
display cap * -state offline_pending
```

display cell

`display cell` コマンドは、選択されたオプションに基づいてセルに関する特定の情報を表示します。

注:

`display` の選択条件の指定と表示するデータの選択についての詳細は、[「display コマンドオプションの使用」](#)を参照してください。

形式

```
display cell cell_loc ... [ -status cell_status ... ] [ [ -c ] | [ -f field ... ] [ -s sort_field ... ] [ -nn ] ]
```

フィールド

`display cell` のフィールドは次のとおりです。

- キーフィールド: *acs*, *lsm*, *panel*, *row*, *column*、
- デフォルトのフィールド: *acs*, *lsm*, *panel*, *row*, *column*, *status*
- 使用可能なフィールド: *acs*, *lsm*, *panel*, *row*, *column*, *status*

オプション

- *-status cell_status*

追加の選択条件として1つまたは複数のセルのステータスを指定します。有効なセルのステータスは、*empty*、*full*、*inaccessible*、*reserved*です。

ワイルドカード文字、* は、任意の *cell_id* サブフィールドで使用することも、すべてのサブフィールドを表すために使用することもできます。ただし、すべての *cell_id* を表す場合 (例: *display cell ** または *display cell_id *,*,**)、追加の *cell_id* (数値または*のいずれか) は許可されません。数値の範囲は *cap_id* のすべてのサブフィールドに適用されます。

- *-c*

(カウント) このリクエストで *arg* および *selection* 条件を満たすオブジェクトの数のみを表示します。このオプションは、*-f field*、*-s sort_field*、および *-n n* オプションと同時に指定できません。

- *-f field*

情報のフィールドと、それらが選択された各オブジェクトに対して返される順序を指定します。それぞれの *display* タイプには、使用可能、キー、およびデフォルトのフィールドのリストがあります。*-f* を指定する場合は、キーフィールドであるかすでに指定されている場合を除いて、デフォルトのフィールドは表示されません。*-f* を指定しない場合は、デフォルトのフィールドが返されます。各タイプのキーフィールドは、通常は最初に表示されます。ただし、*-f* オプションのあとに指定することによって、キーフィールドが表示される順序を変更できます。

フィールドのリストは空白文字で区切ります。

- *-s sort_field*

選択したフィールドによって返されるオブジェクトを、指定した順序でソートします。ソートは内部データベースの値に基づいていて、常に英数字順で表示されるとは限りません。

タイプに対して返されるすべてのフィールドが有効な *sort_field* です。

- *-n n*

表示するオブジェクトの最大数を指定します。

例

予約済みステータスのセルの情報を表示するには:

```
display cell * -status reserved
```

display drive

`display drive` コマンドは、選択されたオプションに基づいて特定のドライブ情報を表示します。

注:

`display` の選択条件の指定と表示するデータの選択についての詳細は、「[display コマンドオプションの使用](#)」を参照してください。

形式

```
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 ] ]
```

フィールド

`display drive` のフィールドは次のとおりです。

- キーフィールド: `acs`、`lsm`、`panel`、`drive`

デフォルトのフィールド: `acs`、`lsm`、`panel`、`drive`、`status`、`state`、`volume`、`type`

- 使用可能なフィールド:

`acs`、`lsm`、`panel`、`drive`、`status`、`state`、`volume`、`type`、`lock`、`desired_state`、`serial_num`、`condition`、`wwn`、`last_dismount_time`、`error`

オプション

- `drive_id`

`drive_id` を次の形式で表示します: `acs`、`lsm`、`panel`、`drive`。

ワイルドカード文字、* は、任意の `drive_id` サブフィールドで使用することも、すべてのサブフィールドを表すために使用することもできます。ただし、すべての `drive_id` (例: `display drive *` または `display drive_id *,*,*`) を表す場合、追加の `drive_id` (数値または * のいずれか) は許可されません。数値の範囲は `drive_id` のすべてのサブフィールドに適用されます。

- `-status drive_status`

追加の選択条件として、次のステータスのいずれかを表示します: `available`、`in_use`、`reserved`。

考えられるシナリオ: `query drive all` を実行し、ドライブが使用中であるというメッセージを取得します。次に、`display drive *` を実行し、ドライブが予約されているというメッセージを取得します。つまり、ドライブの予約済みステータスには、マウントリクエストが進行中であること、およびボリュームがドライブに入る途中であるという意味が含まれています。同時に、ドライブは使用中であると見なされます。

- `-state drive_state`

1 つまたは複数のドライブの次の状態を表示します: `diagnostic`、`online`、`offline`、`recovery`。

- `-type drive_type`

トランスポートタイプ別にドライブを表示します。

- `-volume vol_id`

選択したドライブにマウントされているボリュームを表示します。`vol_id` には、テープボリュームまたはボリューム ID の範囲を指定する、6 文字の英数字文字列またはワイルドカード文字を指定できます。

- `-lock lock_id`

ロックされているドライブのドライブ情報をロック ID 別に表示します。

- `-serial drive_serial_num`

ドライブのシリアル番号を指定します。

- `-condition drive_condition`

指定されたドライブの状況を表示します。有効な状況は、`operative`、`inoperative`、または `maint_required` です。

注:

inoperative または *maint_required* の状況は、ドライブ、LSM、または ACS がオンラインで変更されたときのみクリアされます。そのため、inoperative または *maint_required* のドライブ状況は正確ではないことがあります。

- **-c**

(カウント) このリクエストで *arg* および *selection* 条件を満たすオブジェクトの数のみを表示します。このオプションは、*-f field*、*-s sort_field*、および *-n n* オプションと同時に指定できません。

- **-s *sort_field***

選択したフィールドによって返されたオブジェクトを指定した順序でソートします。ソートは内部データベースの値に基づいていて、常に英数字順で表示されるとは限りません。

タイプに対して返されるすべてのフィールドが有効な *sort_field* です。

- **-f *field***

情報のフィールドと、それらが選択された各オブジェクトに対して返される順序を指定します。それぞれの *display* タイプには、使用可能、キー、およびデフォルトのフィールドのリストがあります。*-f* を指定する場合は、キーフィールドであるかすでに指定されている場合を除いて、デフォルトのフィールドは表示されません。*-f* を指定しない場合は、デフォルトのフィールドが返されます。各タイプのキーフィールドは、通常は最初に表示されます。ただし、*-f* オプションのあとに指定することによって、キーフィールドが表示される順序を変更できます。

フィールドのリストは空白文字で区切ります。次は、*query drive* では報告されない新しいテープドライブフィールドの説明のリストです。

- ***desired_state***

ACS、ポート、LSM、ドライブ、または CAP の目的の状態は、コンポーネントに必要な可用性です。ACSLs は、明示的な *vary* 操作が実行されるときに目的の状態を設定します。(これは *cmd_proc* または *ACSAPI client* コマンドからの *vary* です。ライブラリのステータス変更によって ACSLS が内部的に生成する *vary* ではありません。)

ACS、ポート、LSM、ドライブ、または CAP の現在の状態 (「*state*」として指定されています) は、目的の状態によって制限されるコンポーネントの現在の可用性です。ドライブがオンラインで変更されたときに、その目的の状態がオンラインになります。ただし、ドライブが作動不能か、オフラインまたは準備できていない LSM 内にあるために、その現在

の状態がオフラインである可能性もあります。ACS、ポート、LSM、ドライブ、または CAP の現在の状態は、*query* および *display* コマンドの結果でコンポーネントの「state」としても参照されます。

論理ライブラリおよび論理ライブラリ内のテープドライブの可用性は、基となる物理ライブラリと論理ライブラリの両方に設定した目的の状態によっても左右されます。物理ライブラリと論理ライブラリの両方の目的の状態がオンラインの場合、論理ライブラリと論理テープドライブの現在の状態は、基となる物理ライブラリとドライブの現在の情報を反映します。

- *serial_num*

ACSL S がドライブのシリアル番号を報告できるのは、ドライブがそのシリアル番号をライブラリに報告し、ライブラリがそのシリアル番号を ACSLS に報告する場合のみです。最新のライブラリとドライブのみが、ACSL S にドライブのシリアル番号を報告します。これには、T9840、T9940、T10000、LTO、DLT 7000、SDLT、および以降のドライブが含まれます。SCSI 接続のライブラリ、L5500、T10000 ドライブをサポートする 9310、およびそれ以降のライブラリは、ACSL S にシリアル番号を報告します。

ACSL S は、ドライブトレイ上のシリアル番号ではなく、テープドライブの「ブリック」にあるシリアル番号を報告します。

ドライブのブリックにあるシリアル番号は、ドライブがライブラリと、SL コンソールと ACSLS を使用するユーザーに報告するシリアル番号です。これは SCSI/ファイバデータパス経由で (バックアップアプリケーションなどに) 報告されるシリアル番号でもありません。

ドライブトレイには独自のシリアル番号のラベルが付けられています。ドライブトレイのラベルにあるシリアル番号により、StorageTek の保守などのライセンスが提供されます。ドライブトレイにある番号はアクティブ化に使用されるため、ドライブ (ドライブの「ブリック」) はアクティブ化されたシリアル番号を更新しなくても交換できます。

ドライブトレイとドライブのブリックの間のやり取りは最小限です。これは、一部準備完了/まだ準備できていないというインジケータによって表示されます。ドライブトレイは、基本的には高機能を持たないドライブの電源装置です。

- *wwn*

ドライブの World Wide Name。WWN は 8 バイト (バイナリ) です。これは、16 進文字 (バイト) のペアの間にドット区切り文字のある、16 文字の 16 進文字 (0-9 および A-F) として表示されます。

表示される WWN はノードの WWN です。最初のポート (ポート A) の WWN は、1 つ大きい数字です。ドライブに 2 つのポートがある場合、ポート B の WWN は 2 つ大きい数字です。

ドライブの WWN は、SL3000 ライブラリおよび SL8500 ライブラリ (3.94 以降のファームウェア) で使用できます。

- *last_dismount_time*

カートリッジがドライブから最後にマウント解除された日時。ACSL5 はこれを使用して、*query mount* に報告するドライブを順序付けます。ドライブは、第一にマウントされるボリュームとの距離が近い (パススルーがもっとも少ない) LSM 順に、第二にもっとも早い *last_dismount_time* 順にリストされます。*earliest last_dismount_time* によってドライブを順序付けすると、ドライブの使用が均一になります。

これらの統計情報は、カートリッジがドライブからマウント解除されたときに ACSLS に報告されます。次の統計フィールドは、これらのライブラリとドライブの場合のみ報告されます。

ライブラリ:

- › すべての SL3000。
- › SL8500 (4.13 以降のファームウェア)。

テープドライブ:

- › T9840A、T9840C、および T9840D (1.42 以降のドライブファームウェア)。(メモリーの制限のため、T9840B はサポートされていません。)
 - › T9940A および T9940B (1.42 以降のドライブファームウェア)。
 - › T10000A および T10000B (1.38 以降のドライブファームウェア)。
- *error* - (ドライブの保守が必要)

これは、現時点では T10000 ドライブの場合のみ報告されます。値は次のとおりです。

- › *none* - ドライブエラーは報告されません。
- › *maint_reqd* - テープドライブに保守が必要です。

これは、テープドライブが、ドライブを避けるべきでありオフラインにする必要があると判断した場合のみ報告されます。修理を依頼してください。

主なドライブエラーのシナリオ:

これらのシナリオでは、ドライブとライブラリは手動介入を待機します。

表14.1 手動介入を待機するドライブとライブラリ

アンロードの障害	カートリッジがドライブ内でスタックしている
ドライブリーダーのオフフック障害	メディアが引き出されるまで、ドライブはドライブエラーを示しません
ドライブ IPL 障害	ドライブはライブラリとの通信を停止します。

- `-s sort_field`

選択したフィールドによって返されたオブジェクトを指定した順序でソートします。ソートは内部データベースの値に基づいていて、常に英数字順で表示されるとは限りません。たとえば、`drive type` は内部の数値のドライブタイプでソートされ、`display` であるリテラルではソートされません。

タイプに対して返されるすべてのフィールドが有効な `sort_field` です。

- `-n n`

表示するオブジェクトの最大数を指定します。

例

- パネル 0,1,1 のすべてのドライブを表示するには:

```
display drive 0,1,1, *
```

- LSM 0,1 のすべてのドライブを表示するには:

```
display drive 0,1, *, *
```

- パネルとドライブでソートされた、ACS1 内のすべてのドライブの最大 56 行を表示するには:

```
display drive 1, *, *, * -s panel drive -n 56
```

- ステータスが使用可能の、ACS 1、LSM 1 のドライブデータを表示するには:

```
display drive 1,1, *, * -status available
```

- すべてのドライブのドライブタイプ、シリアル番号、および World Wide Name (wwn) を表示するには:

```
display drive * -f type serial_num wwn
```

- すべてのドライブの目的の状態を表示するには:

```
display drive * -f state desired_state
```

display lock

`display lock` コマンドは、選択されたオプションに基づいて、特定のロック情報を `lock_id` 別に表示します。

注:

`display` の選択条件の指定と表示するデータの選択についての詳細は、「[display コマンドオプションの使用](#)」を参照してください。

形式

```
display lock lock_id ... [ -user user_id ... ] [ [ -c ] | [ -f field.
.. ] [ -s sort_field ... ] [ -n n ] ]
```

フィールド

`display lock` のフィールドは次のとおりです。

- キーフィールド: `lock_id`
- デフォルトのフィールド: `lock_id, user_id`
- 使用可能なフィールド: `lock_id, user_id`

オプション

- `lock_id`

数値の `lock_id` を表示します。

ワイルドカード文字、* は、任意の `lock_id` サブフィールドで使用することも、すべてのサブフィールドを表すために使用することもできます。ただし、すべての `lock_id` (例: `display lock *` または `display lock_id *,*,*`) を表す場合、追加の `lock_id` (数値または * のいずれか) は許可されません。数値の範囲は `lock_id` のすべてのサブフィールドに適用されます。

- `user_id`

ボリュームまたはトランスポートがロックされているユーザー名 (例: `acsss`) を指定します。`user_id` は、80 文字を超えると折り返されます。

- `-c`

(カウント) このリクエストで `arg` および `selection` 条件を満たすオブジェクトの数のみを表示します。このオプションは、`-f field`、`-s sort_field`、および `-n n` オプションと同時に指定できません。

- `-f field`

情報のフィールドと、それらが選択された各オブジェクトに対して返される順序を指定します。それぞれの `display` タイプには、使用可能、キー、およびデフォルトのフィールドのリストがあります。`-f` を指定する場合は、キーフィールドであるかすでに指定されている場合を除いて、デフォルトのフィールドは表示されません。`-f` を指定しない場合は、デフォルトのフィールドが返されます。各タイプのキーフィールドは、通常は最初に表示されます。ただし、`-f` オプションのあとに指定することによって、キーフィールドが表示される順序を変更できます。

フィールドのリストは空白文字で区切ります。

- `-s sort_field`

選択したフィールドによって返されたオブジェクトを指定した順序でソートします。ソートは内部データベースの値に基づいていて、常に英数字順で表示されるとは限りません。

タイプに対して返されるすべてのフィールドが有効なソートフィールドです。

- `-n n`

表示するオブジェクトの最大数を指定します。

例

- `lock_id 2` のロック情報を表示するには:

```
display lock 2
```

display lsm

`display lsm` コマンドは、選択されたオプションに基づいて特定の LSM 情報を表示します。

注:

`display` の選択条件の指定と表示するデータの選択についての詳細は、[「display コマンドオプションの使用」](#)を参照してください。

形式

```
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 ] ]
```

フィールド

display lsm のフィールドは次のとおりです。

- キーフィールド: *acs*、*lsm*
- デフォルトのフィールド: *acs*、*lsm*、*status*、*state*、*free_cells*
- 使用可能なフィールド: *acs*、*lsm*、*status*、*state*、*desired_state*、*free_cells*、*type*、*serial_num*、*activity*、*condition*、*door_status*
- *lsm_id*

lsm id を次の形式で表示します: *acs*、*lsm*。

ワイルドカード文字、* は、任意の *lsm_id* サブフィールドで使用することも、すべてのサブフィールドを表すために使用することもできます。ただし、すべての *lsm_id* (例: *display lsm ** または *display lsm_id *, **) を表す場合、追加の *lsm_ids* (数値または * のいずれか) は許可されません。数値の範囲は *lsm_id* のすべてのサブフィールドに適用されます。

- *-status lsm_status*

1 つまたは複数の選択する LSM のステータスを指定します。有効なステータスは、*audit*、*normal* です

- *-state lsm_state*

1 つまたは複数の選択する LSM の状態を指定します。有効な状態は、*diagnostic*、*online*、*offline*、*offline_pending*、*recovery* です。

- *-free_cells cell_count*

空きセルの数によって LSM を選択します。

- *-type lsm_type*

LSM をタイプ別に表示します。

- *-serial lsm_serial_num*

選択された LSM のシリアル番号を表示します。

SL8500 6.0+ および SL3000 3.0+ ファームウェアでは、ライブラリはライセンス所有のシリアル番号を報告します。(SL8500 ライブラリ番号を取得するには、LSM 番号に 1 を足して 4 で割ります。)

- *-condition lsm_condition*

選択された LSM の状況を表示します。有効な状況は、*operative*、*inoperative*、または *maint_required* です。

注:

inoperative または *maint_required* の状況は、ドライブ、LSM、または ACS がオンラインで変更されたときにのみクリアされます。そのため、*inoperative* または *maint_required* の LSM の状況は正確ではないことがあります。

- *-door_open*

ドアが開いている LSM を表示します。

- *-door_closed*

ドアが閉じている LSM を表示します。

- *-c*

(カウント) このリクエストで *arg* および *selection* 条件を満たすオブジェクトの数のみを表示します。このオプションは、*-f field*、*-s sort_field*、および *-n n* オプションと同時に指定できません。

- *-f field*

情報のフィールドと、それらが選択された各オブジェクトに対して返される順序を指定します。それぞれの *display* タイプには、使用可能、キー、およびデフォルトのフィールドのリストがあります。*-f* を指定する場合は、キーフィールドであるかすでに指定されている場合を除いて、デフォルトのフィールドは表示されません。*-f* を指定しない場合は、デフォルトのフィールドが返されます。各タイプのキーフィールドは、通常は最初に表示されます。ただし、*-f* オプションのあとに指定することによって、キーフィールドが表示される順序を変更できます。

フィールドのリストは空白文字で区切ります。次は、*query lsm* では報告されない新しい LSM フィールドです。

- *desired_state*

ACS、ポート、LSM、ドライブ、または CAP の目的の状態は、コンポーネントに必要な可用性です。ACSLs は、明示的な *vary* 操作が実行されるときに目的の状態を設定します。(これは *cmd_proc* または *ACSAPI client* コマンドからの (*vary*) です。ライブラリのステータス変更によって ACSLS が内部的に生成する *vary* ではありません。)

ACS、ポート、LSM、ドライブ、または CAP の現在の状態 (「state」として指定されています) は、目的の状態によって制限されるコンポーネントの現在の可用性です。ドライブがオンラインで変更されたときに、その目的の状態がオンラインになります。ただし、ドライブが作動不能か、オフラインまたは準備できていない LSM 内にあるために、その現在の状態がオフラインである可能性もあります。ACS、ポート、LSM、ドライブ、または CAP の現在の状態は、*query* および *display* コマンドの結果でコンポーネントの「state」としても参照されます。

論理ライブラリおよび論理ライブラリ内のテープドライブの可用性は、基となる物理ライブラリと論理ライブラリの両方に設定した目的の状態によっても左右されます。物理ライブラリと論理ライブラリの両方の目的の状態がオンラインの場合、論理ライブラリと論理テープドライブの現在の状態は、基となる物理ライブラリとドライブの現在の情報を反映します。

- *serial_num*

ACSLs が報告する LSM のシリアル番号は、SL3000 および SL8500 ライブラリ (SL8500 6.00 ファームウェアまたは SL3000 3.0 ファームウェアより前) の HBC カードに記録されているライブラリのシリアル番号です。以降のファームウェアの SL8500 および SL3000 ライブラリは、アクティブ化されたシリアル番号であるフレームのシリアル番号を報告します。

SL8500 および SL3000 ライブラリは、基本モジュールフレームのシリアル番号を使用してアクティブ化されます。このシリアル番号は、フレームの UL ラベル上に記載され、ライブラリの HBK カードに記録されています。(製造で交換用 HBK カードを再生成できます。)これらのシリアル番号は、SL8500 の場合は 516、SL3000 の場合は 571 で始まります。フレームのシリアル番号は、SL コンソールを使用して表示できます (「System Detail」->「Properties」->「General」)。

今後は、ライブラリファームウェアがこのフレームのシリアル番号を ACSLS に報告するようになります。LSM シリアル番号が変更されたときは常に、LSM がオンラインで変更されたときに ACSLS が自動的にシリアル番号を更新します。

- *-s sort_field*

選択したフィールドによって返されたオブジェクトを指定した順序でソートします。ソートは内部データベースの値に基づいていて、常に英数字順で表示されるとは限りません。

タイプに対して返されるすべてのフィールドが有効なソートフィールドです。

- `-n n`

表示するオブジェクトの最大数を指定します。

例

- タイプ 9714 のライブラリ内のすべての LSM についての情報を表示するには:

```
display lsm * -type 9714
```

- すべての LSM タイプを表示するには:

```
display lsm * -f type
```

- ライブラリのシリアル番号を表示するには:

```
display lsm * -f serial_num
```

- すべての LSM の状態と `desired_state` を表示するには:

```
display lsm * -f state desired_state
```

display panel

`display panel` コマンドは、選択されたオプションに基づいて特定のライブラリパネル情報を表示します。

注:

`display` の選択条件の指定と表示するデータの選択についての詳細は、「[display コマンドオプションの使用](#)」を参照してください。

形式

```
display panel panel_id ... [ -type panel_type ... ] [ [ -c ] |[ -f field ... ] [ -s sort_field ... ] [ -n n ] ]
```

フィールド

`display panel` のフィールドは次のとおりです。

- キーフィールド: `acs`、`lsm`、`panel`

- デフォルトのフィールド: *acs*、*lsm*、*panel*、*type*
- 使用可能なフィールド: *acs*、*lsm*、*panel*、*type*

オプション

- *panel_id*

パネル ID を次の形式で表示します: *acs*、*lsm*、*panel*。

ワイルドカード文字、* は、任意の *panel_id* サブフィールドで使用することも、すべてのサブフィールドを表すために使用することもできます。ただし、すべての *panel_id* (例: *display panel ** または *display panel_id *, *, **) を表す場合、追加の *panel_id* (数値または * のいずれか) は許可されません。数値の範囲は *panel_id* のすべてのサブフィールドに適用されます。

- *-type panel_type*

1 つまたは複数の有効なパネルタイプを指定します。

- *-c*

(カウント) このリクエストで *arg* および *selection* 条件を満たすオブジェクトの数のみを表示します。このオプションは、*-f field*、*-s sort_field*、および *-n n* オプションと同時に指定できません。

- *-f field*

情報のフィールドと、それらが選択された各オブジェクトに対して返される順序を指定します。それぞれの *display* タイプには、使用可能、キー、およびデフォルトのフィールドのリストがあります。*-f* を指定する場合は、キーフィールドであるかすでに指定されている場合を除いて、デフォルトのフィールドは表示されません。*-f* を指定しない場合は、デフォルトのフィールドが返されます。各タイプのキーフィールドは、通常は最初に表示されます。ただし、*-f* オプションのあとに指定することによって、キーフィールドが表示される順序を変更できます。

フィールドのリストは空白文字で区切ります。

- *-s sort_field*

選択したフィールドによって返されたオブジェクトを指定した順序でソートします。ソートは内部データベースの値に基づいていて、常に数値順で表示されるとは限りません。

タイプに対して返されるすべてのフィールドが有効なソートフィールドです。

- `-n n`

表示するオブジェクトの最大数を指定します。

例

すべてのパネルのパネルデータを表示します。

```
display panel *
```

display pool

`display pool` コマンドは、選択されたオプションに基づいて特定のスクラッチプール情報を表示します。

注:

`display` の選択条件の指定と表示するデータの選択についての詳細は、[「display コマンドオプションの使用」](#)を参照してください。

形式

```
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 ] ]
```

フィールド

`display pool` のフィールドは次のとおりです。

- キーフィールド: `pool_id`
- デフォルトのフィールド: `pool_id`、`low_water`、`high_water`、`overflow`
- 使用可能なフィールド: `pool_id`、`low_water`、`high_water`、`overflow`

オプション

- `pool_id`

スクラッチプールのプール ID を表示します。

ワイルドカード文字、* は、任意の `pool_id` サブフィールドで使用することも、すべてのサブフィールドを表すために使用することもできます。ただし、すべての `pool_id` (例:

`display pool *` または `display pool_id *, *, *` を表す場合、追加の `pool_id` (数値または * のいずれか) は許可されません。数値の範囲は `pool_id` のすべてのサブフィールドに適用されます。

- `-low_water low_water_mark`

プール内のスクラッチカートリッジの最小数を指定します。`low_water` マークにハイフン (-) が付加されて表示される場合、選択されたスクラッチプールのライブラリ内のカートリッジの数は指定された `low_water` マーク内に示されています。

- `-high_water high_water_mark`

プール内のスクラッチカートリッジの最大数を指定します。`high_water` マークにハイフン (-) が付加されて表示される場合、選択されたスクラッチプールのライブラリ内のカートリッジの数は指定された `high_water` マーク内に示されています。

- `-overflow`

スクラッチマウントがリクエストされた場合と、プールが空の場合に、共通プールがリクエストを満たすプールを選択します。

- `-no_overflow`

スクラッチマウントがリクエストされた場合と、プールが空の場合に、共通プールで使用可能なテープをチェックせずにマウントが失敗するプールを選択します。

- `-c`

(カウント) このリクエストで `arg` および `selection` 条件を満たすオブジェクトの数のみを表示します。このオプションは、`-f field`、`-s sort_field`、および `-n n` オプションと同時に指定できません。

- `-f field`

情報のフィールドと、それらが選択された各オブジェクトに対して返される順序を指定します。それぞれの `display` タイプには、使用可能、キー、およびデフォルトのフィールドのリストがあります。`-f` を指定する場合は、キーフィールドであるかすでに指定されている場合を除いて、デフォルトのフィールドは表示されません。`-f` を指定しない場合は、デフォルトのフィールドが返されます。各タイプのキーフィールドは、通常は最初に表示されます。ただし、`-f` オプションのあとに指定することによって、キーフィールドが表示される順序を変更できます。

フィールドのリストは空白文字で区切ります。

- `-s sort_field`

選択したフィールドによって返されたオブジェクトを指定した順序でソートします。ソートは内部データベースの値に基づいていて、常に数値順で表示されるとは限りません。

タイプに対して返されるすべてのフィールドが有効なソートフィールドです。

- `-n n`

表示するオブジェクトの最大数を指定します。

例

- プール 5 のすべての情報を表示するには:

```
display pool 5
```

- オーバーフローのあるすべてのプールを表示するには:

```
display pool * -overflow
```

display port

`display port` コマンドは、選択されたオプションに基づいて特定のポート情報を表示します。

注:

`display` の選択条件の指定と表示するデータの選択についての詳細は、[「display コマンドオプションの使用」](#)を参照してください。

形式

```
display port port_id ... [ -online | -offline ][ -name port_name .  
.. ] [ [ -c ] | [ -f field ... ] [ -s sort_field ... ] [ -n n ] ]
```

フィールド

`display port` のフィールドは次のとおりです。

- キーフィールド: `acs`、`port`
- デフォルトのフィールド: `acs`、`port`、`name`、`state`
- 使用可能なフィールド: `acs`、`port`、`name`、`state`、`desired_state`

オプション

- `port_id`

port id を次の形式で表示します: *acs*、*port*。

ワイルドカード文字、* は、任意の *port_id* サブフィールドで使用することも、すべてのサブフィールドを表すために使用することもできます。ただし、すべての *port_id* (例: *display port ** または *display port_id *, *, **) を表す場合、追加の *port_id* (数値または * のいずれか) は許可されません。数値の範囲は *port_id* のすべてのサブフィールドに適用されます。

- *online*

オンライン状態のポートを選択して表示します。

- *offline*

オフライン状態のポートを選択して表示します。

- *name port_name*

ポート名を指定します。

- *-c*

(カウント) このリクエストで *arg* および *selection* 条件を満たすオブジェクトの数のみを表示します。このオプションは、*-f field*、

-s sort_field、および *-n n* オプションと同時に指定できません。

- *-f field*

情報のフィールドと、それらが選択された各オブジェクトに対して返される順序を指定します。それぞれの *display* タイプには、使用可能、キー、およびデフォルトのフィールドのリストがあります。*-f* を指定する場合は、キーフィールドであるかすでに指定されている場合を除いて、デフォルトのフィールドは表示されません。*-f* を指定しない場合は、デフォルトのフィールドが返されます。各タイプのキーフィールドは、通常は最初に表示されます。ただし、*-f* オプションのあとに指定することによって、キーフィールドが表示される順序を変更できます。

フィールドのリストは空白文字で区切ります。次は、*query port* では報告されない新しいポートフィールドです。

- *desired_state*

ACS、ポート、LSM、ドライブ、または CAP の目的の状態は、コンポーネントに必要な可用性です。ACSLs は、明示的な *vary* 操作が実行されるときに目的の状態を設定しま

す。(これは `cmd_proc` または `ACSAPI client` コマンドからの (`vary`) です。ライブラリのステータス変更によって ACSLS が内部的に生成する `vary` ではありません。)

ACS、ポート、LSM、ドライブ、または CAP の現在の状態は、目的の状態によって制限されるコンポーネントの現在の可用性です。ドライブがオンラインで変更されたときに、その目的の状態がオンラインになります。ただし、ドライブが作動不能か、オフラインまたは準備できていない LSM 内にあるために、その現在の状態がオフラインである可能性もあります。ACS、ポート、LSM、ドライブ、または CAP の現在の状態は、`query` および `display` コマンドの結果でコンポーネントの「state」としても参照されます。

論理ライブラリおよび論理ライブラリ内のテープドライブの可用性は、基となる物理ライブラリと論理ライブラリの両方に設定した目的の状態によっても左右されます。物理ライブラリと論理ライブラリの両方の目的の状態がオンラインの場合、論理ライブラリと論理テープドライブの現在の状態は、基となる物理ライブラリとドライブの現在の情報を反映します。

- `-s sort_field`

選択したフィールドによって返されたオブジェクトを指定した順序でソートします。ソートは内部データベースの値に基づいていて、常に英数字順で表示されるとは限りません

タイプに対して返されるすべてのフィールドが有効なソートフィールドです。

- `-n n`

表示するオブジェクトの最大数を指定します。

例

- オンライン状態のすべてのポートを表示するには:

```
display port * -online
```

display volume

`display volume` コマンドは、選択されたオプションに基づいて特定のボリューム情報を表示します。

注:

`display` の選択条件の指定と表示するデータの選択についての詳細は、[「display コマンドオプションの使用」](#)を参照してください。

形式

```
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 ]
```

フィールド

`display volume` のフィールドは、次のとおりです。

- キーフィールド: `vol_id`
- デフォルトのフィールド: `vol_id`, `acs`, `lsm`, `panel`, `row`, `column`, `pool`, `status`, `media`, `type`
- 使用可能なフィールド: `vol_id`; `acs`; `lsm`; `panel`; `row`; `column`; `drive_lsm`, `drive_panel`, `drive` (カートリッジがマウントされている場所を表示するには、3 つすべてを表示します); `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`; `vol_safe_status`; `media_status`; `warranty_life`; `end_of_life`; `load_limit_alert`

オプション

- `vol_id`

選択されたボリュームを表示します。

`vol_id` には、テープカートリッジを指定する 1 つ以上の 1-16 の英数字文字列を指定できます。

ワイルドカード文字、* は、すべての `vol_id` (例: `display vol*`) を表すため、またはカートリッジのグループを表すために使用できます (例: `display vol_id VAP*` は VAP で始まるすべてのカートリッジを表示します)。

`vol_id` には、任意の有効なボリューム ID と * を含めることもできます。

- `home home_loc`

ボリューム情報の取得元のボリュームのホーム位置を、次の形式で指定します:

acs, lsm, panel, row, column。これは
-drive と一緒には使用できません。

home_loc サブフィールドでは数値範囲を使用できます。すべての *home_loc* サブフィールドを指定する必要があります。ワイルドカード文字、* は、1 つまたは複数の *drive_loc* サブフィールドの代わりに使用できます。サブフィールドまたは *drive_loc* に数値範囲ルールが適用されます。

- *acs <acs_id>*

ボリューム情報を取得する必要がある ACS を指定します。

- *drive drive_loc*

ボリューム情報の取得元のドライブ位置を次の形式で指定します:

acs, lsm, panel, drive。これは -home と一緒には使用できません

drive_loc サブフィールドでは数値範囲を使用できます。すべての *home_loc* サブフィールドを指定する必要があります。ワイルドカード文字、* は、1 つまたは複数の *drive_loc* サブフィールドの代わりに使用できます。数値範囲ルールはサブフィールドまたは *drive_loc* に適用されます。

- *data*

データカートリッジを選択します。

- *scratch*

スクラッチモードのカートリッジを選択します。

- *clean*

クリーニングカートリッジを選択します。

- *spent_clean*

ドライブタイプ別に使用済み (使い果たされた) として報告されたクリーニングカートリッジを選択します。

- *media media_type*

指定されたメディアタイプのカートリッジのみを選択します。

- *pool pool_id*

スクラッチプールからのカートリッジを表示および選択します。カートリッジは、データカートリッジに変更されたあとも、スクラッチプールの ID を保持します。

- *standard*

スキャン可能なボリューム ID *label_type* のテープを選択します。これは *-virtual* と一緒には使用できません。

- *virtual*

venter コマンドによって挿入された外部ボリューム ID *label_type* のないテープを選択します。これは *-standard* と一緒には使用できません。

- *status vol_status*

有効なエントリ

は、*dismount*、*eject*、*enter*、*mount*、*home*、*in_drive*、*move*、*missing*、*absent*、*ejected* です。

- *entry entry_date*

カートリッジがライブラリに挿入された日時を指定します。これは ISO 標準形式 (yyyy-mm-dd) で入力および表示されます。

- *access access_date*

カートリッジが最後に使用された日時を指定します。これは ISO 標準形式 (yyyy-mm-dd) で入力および表示されます。

- *lock lock_id*

ボリュームのロック ID を指定します。

- *max_use max_use*

クリーニングカートリッジの最大使用回数を指定します。クリーニングカートリッジではないカートリッジの *max_use* 値はゼロです。

- *lock_time lock_time*

ロックが設定された日時を指定します。これは、システムのデフォルトの形式に従って表示されます。

- *recording_format_family drive_family -recording_format_model drive_model*

記録形式のドライブファミリーとドライブモデルを指定します。例: T10000 および T10000C

- *c*

(カウント) このリクエストで *arg* および *selection* 条件を満たすオブジェクトの数のみを表示します。このオプションは、*-f field*、*-s sort_field*、および *-n n* オプションと同時に指定できません。

- *-f field*

情報のフィールドと、それらが選択された各オブジェクトに対して返される順序を指定します。それぞれの *display* タイプには、使用可能、キー、およびデフォルトのフィールドのリストがあります。*-f* を指定する場合は、キーフィールドであるかすでに指定されている場合を除いて、デフォルトのフィールドは表示されません。*-f* を指定しない場合は、デフォルトのフィールドが返されます。各タイプのキーフィールドは、通常は最初に表示されます。ただし、*-f* オプションのあとに指定することによって、キーフィールドが表示される順序を変更できます。

フィールドのリストは空白文字で区切ります。次は、*query volume* では報告されない新しいボリュームフィールドのリストです。これらの統計フィールドは、これらのライブラリとドライブの場合のみ報告されます。

次の統計フィールドは、これらのライブラリとドライブの場合のみ報告されます。

ライブラリ:

- すべての SL3000。
- SL8500 (4.13 以降のファームウェア)。
- テープドライブ:
 - T9840A、T9840C、および T9840D (1.42 以降のドライブファームウェア)。(メモリーの制限のため、T9840B はサポートされていません。)
 - T9940A および T9940B (1.42 以降のドライブファームウェア)。
 - T10000A および T10000B (1.38 以降のドライブファームウェア)。
 - T10000C および T10000D、および以降のドライブ。

- *recording_format_family*

これは、このカートリッジの最後にデータに記録されたドライブファミリを示しています。

- *recording_format_model*

これは、このカートリッジの最後にデータに記録されたドライブモデルを示しています。この情報は次に役立ちます。

- カートリッジ上のデータを読み取ることができないドライブに、カートリッジがマウントされないようにします (T10000A が T10000B ドライブで記録されたデータを読み取ることができない場合など)、または
- カートリッジにデータを付加します (T10000B が T10000A によって書き込まれたデータをカートリッジに付加できない場合など)。

- *encrypt_status*

これは、既知の場合はカートリッジの暗号化ステータスを示します。

- *volSAFE_status*

これは、これが *volSAFE* カートリッジかどうかを示します。*volSAFE* は StorageTek の WORM (Write Once Read Many) カートリッジです。

- *media_status*

テープドライブからカートリッジが「問題あり」として報告された場合は、それを調べる必要があります。

- *warranty_life* および *end_of_life*

ボリュームの *warranty_life* と *end_of_life* のパーセンテージは、その *end_of_life* 設定のパーセンテージとしてのボリュームの使用量です。これらは 000.0% から 100.0% を超える値です。(*warranty_life* は *end_of_life* より小さい値です。)

この情報を使用して、新しいカートリッジにデータを移行したあとで廃棄する必要のあるカートリッジを特定できます。

- *load_limit_alert*

ボリュームの *load_limit_alert* はブール値であり、ここでは True=1 および False=0 です。*end_of_life* のパーセンテージを報告するボリュームが 100% に達すると、その *load_limit_alert* が 1 に設定されます。一部のドライブタイプでは *load_limit_alert* のみが報告されます。

- *-s sort_field*

選択したフィールドによって返されるオブジェクトを、指定した順序でソートします。ソートは内部データベースの値に基づいていて、常に英数字順で表示されるとは限りません。たとえば、*media type* は内部の数値のメディアタイプでソートされ、*display* であるリテラルではソートされません。

タイプに対して返されるすべてのフィールドが有効なソートフィールドです。

- `-n n`

表示するオブジェクトの最大数を指定します。

例

- 位置に基づいてボリュームを表示するには:

この例では、ホーム位置がパネル 0,1,9 にあるすべてのボリュームが表示されます。

```
display volume * -home 0,1,9,*,*
```

- すべての 9840 データボリュームを表示するには:

```
display volume * -media STK1R
```

- 9840 データボリュームとその記録形式 (既知の場合) を表示するには:

```
display volume * -media STK1R -f media recording_format_family  
recording_format_model
```

- ACS およびメディアタイプ別にクリーニングボリュームを表示するには:

```
display volume * -home acs,*,*,*,* -media media_type  
-f vol_id acs lsm media max_use access_count
```

- ボリュームを日付順に表示するには:

- 2011 年に挿入されたすべてのボリュームを表示するには:

```
display volume * -entry 2011
```

- 2011 年 1 月に挿入されたすべてのボリュームを表示するには

```
display volume * -entry 2011-01
```

- 2011 年 1 月 27 日に挿入されたすべてのボリュームを表示するには

```
display volume * -entry 2011-01-27
```

- 2011 年 1 月 27 日の午前 10 時から 11 時の間に挿入されたすべてのボリュームを表示するには

```
display volume * -entry 2011-01-27:10
```

- 2011 年 1 月から 2011 年 4 月までの間に挿入されたすべてのボリュームを表示するには:

```
display volume * -entry 2011-01-2011-04
```

- 4月10日の午前6時33分から6時57分45秒までに挿入されたすべてのボリュームを表示するには (たとえば、停電のあとにこれを実行します):

```
display volume * -entry 2011-04-10:06:33:00-2011-04-10:6:57:45
```

- 2011年2月25日以降に挿入されたすべてのボリュームを表示するには:

```
display volume * -entry >2011-02-25
```

- `end_of_life` 使用回数でソートされた T10000 データボリュームを表示するには:

```
display volume * -media T10000T1 T10000TS -f media end_of_life  
warranty_life -s end_of_life
```

- すべてのボリュームの保証期限と寿命を `end_of_life` 使用回数でソートして表示するには

```
display volume * -f warranty_life end_of_life -s end_of_life
```

- 使用済み (使い果たされた) クリーニングカートリッジを表示するには:

```
display volume * -spent_clean
```


第15章 lib_cmd

この章では、オプションのコマンド行インタフェース (CLI)、*lib_cmd* について説明します。このツールは主に ACSLS の論理ライブラリの管理に使用することを目的としていますが、一部のコマンドは、ある種の特定のコンテキストで ACSLS GUI または *cmd_proc* の代替として便利な場合があります。*lib_cmd* CLI は、長い (42 より大きい) ボリュームリストを指定する機能を含む、取り出し操作をサポートしています。

はじめに

ACSLS は、バージョン 8.0 で論理ライブラリの概念を導入しました。論理ライブラリのクライアント操作には、ファイバチャネル接続経由で SCSI Media Changer Emulation (SMCE) を介し、SCSI コマンドを使用してアクセスします。クライアントは、ライブラリがクライアントアプリケーションに直接接続されているかのように操作することで、テープのマウント (*mount*)、マウント解除 (*dismount*)、挿入 (*enter*)、および取り出し (*eject*) を行なったり、SMCE インタフェースを使用してその独自のテープインベントリを管理したりできます。

ACSLS 8.2 まで、論理ライブラリ管理設定機能は、ACSLS グラフィカルユーザーインタフェース (同様に ACSLS 8.0 で導入) を介して行われていました。そのようなタスクには、目的の数のストレージセルとテープドライブスロットを含む論理テープライブラリの作成、クライアントのイニシエーターとターゲットの関係の確立、ライブラリへの物理ドライブの割り当て、およびテープボリュームの割り当てが含まれます。

これらの管理タスクは、UNIX コマンド行インタフェース (CLI) を使用して実行することもできます。CLI は、クリティカルな管理機能へのより高速なパスを提供し、バッチ処理の UNIX シェルスクリプトを使用して論理ライブラリを管理する手段を提供することで、論理ライブラリの操作性を拡張します。

コマンド

lib_cmd

パス名

`$ACS_HOME/bin/lib_cmd`

概要

```

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 ]

```

lib_cmd の使用

lib_cmd コマンド行ユーティリティーは ACSLS GUI の代替機能であり、ACSLs 論理ライブラリのリソースを管理およびモニターできます。*lib_cmd* を使用して、ACSLs 管理者は、論理ライブラリの作成、編集、または削除、論理ライブラリへのボリュームまたはテープドライブの割り当てまたは割り当て解除、イニシエーターまたはターゲットのリフレッシュ、クライアント通信用のイニシエーターとターゲットのマッピングの作成、論理ライブラリのステータスの表示、構成されている物理ライブラリまたは論理ライブラリ内のボリュームやドライブのステータスの表示を行うことができます。

論理ライブラリの管理に加えて、lib_cmd は、cmd_proc の代わりにライブラリのコンテンツの表示または単一の CAP (SL3000 AEM など) への大量のボリュームの取り出しに使用できる、一般的なライブラリコマンドの小さいセットを備えています。

これらの操作は、ACSL5 グループのユーザー (*root* を含む) が実行できます。

lib_cmd は、対話型モードまたはバッチモードで操作できます。対話型モードは、単独でのコマンドによって、引数を指定せずに呼び出されます。

lib_cmd

対話型モードは、指定できるすべてのサブコマンドと引数をリストする「ヘルプ」機能をサポートしています。各サブコマンドのコンテキスト内で指定できる引数とオプションのリストは、リクエスト時に表示できます。指定できる有効な入力文字列がわからない場合は、<Enter> キーを押して、即時コンテキストで指定できるすべての引数のリストを取得できます。

バッチモードには、このユーティリティとユーザー生成のスクリプトとの併用をより適切にサポートする対話機能はありません。バッチモードでは、目的のオプションと引数のすべてをコマンドとともに発行する必要があります。そうでない場合は、エラーメッセージが返されます。

オプション

- *-f* (Command file input specification)

-f オプションがリクエストされると、ユーティリティは、指定された入力ファイル内で検出した一連のコマンドを実行します。入力ファイルには、完全形式のリクエストのみが含まれている必要があります。

サブコマンド

このセクションでは、サブコマンドについて説明します。

assign drive drive_id lib_id

指定したドライブを指定したライブラリに割り当てます。*drive_id* は、物理ライブラリ (*acs*, *lsm*, *panel*, *drive*) 内のドライブの標準的な表記で表されます。引数 *lib_id* は、論理ライブラリの割り当て済みの ID です。論理ライブラリ ID は 1001 から始まり、ライブラリが作成されると増分で増加します。

unassign drive lib_id logical_drive_id

指定した論理ドライブを論理ライブラリ構成から削除します。ドライブはその論理アドレス (アドレス 500 で始まる) によって指定されます。論理ライブラリ ID は 4 桁の数字 (1001 で始まる) です。

```
assign volume vol_id | vol-range lib_id
```

論理ライブラリ内での排他的な使用のためにボリュームを割り当てます。ボリュームは、標準の 6 またはバーコードラベル形式によって表されます。ライブラリは、その数値の論理ライブラリ ID によって表されます。ボリュームは「割り当て可能」である必要があります。つまり、まだ割り当て済みではなく、所有されていない、マウント済み、予約済み、転送中、あるいは使用中ではありません。

範囲は、低い ACSII 値文字列と高い ASCII 値文字列をダッシュで区切って指定します。ボリューム範囲がリクエストされると、*lib_cmd* が指定された範囲内のボリュームのバックアップ ACS を検索します。範囲内のすべての割り当て可能なボリュームが、指定された論理ライブラリに割り当てられます。

```
unassign volume vol_id | vol-range lib_id
```

論理ライブラリインベントリからボリュームを削除します。ボリュームはそのホームセル内に存在する必要があり、保留中の移動操作のためには予約できません。

範囲は、低い ACSII 値文字列と高い ASCII 値文字列をダッシュで区切って指定します。ボリューム範囲がリクエストされると、*lib_cmd* が指定された範囲内のボリュームの指定された論理ライブラリを検索します。範囲内の使用中ではない割り当て済みのボリュームは、ライブラリから割り当て解除されます。

```
create library lib_name backing_acs cell_capacity drive_capacity
```

新しい論理ライブラリを作成します。*create library* のあと、この順序の 4 つの引数が必要です: ライブラリ別名 (文字列)、バックアップ ACS ID (整数)、ストレージセルの数 (整数)、およびドライブスロットの数 (整数)。

```
create mapping initiator_id target_id library_id
```

指定したライブラリにイニシエーターとターゲット (クライアント - サーバー) の関係を構築します。

```
edit library lib_id [-n name ] [-c capacity ] [-d drive_slots ] [-f volume_label_format [6|8p|8s|all] [-x imp/exp_cell_count ]
```

既存の論理ライブラリの構成を変更します。特定のライブラリは、4 桁の数値 ID によって表されます。任意の 1 つのオプションを指定でき、その 1 つのフィールドのみが、このオプションで指定された引数を使用して変更されます。次のオプションがあります。

- *-n* ライブラリの別名
- *-c* 論理ストレージセルの容量
- *-d* 論理ドライブスロットの数
- *-f* ボリュームラベル形式
- *-x* 論理インポート/エクスポート (CAP) セルの数

ボリュームラベル形式は次のいずれかで表現できます。

- *6* 6 文字のレガシーボリュームラベル
- *8p* メディアタイプコードの接頭辞を付加した 8 文字
- *8s* メディアタイプコードの接尾辞を付加した 8 文字
- *all* すべてのラベル形式を受け入れます

refresh initiator

現在 ACSLS が認識しているすべてのイニシエーターを取得します。

refresh target

現在 ACSLS が認識しているすべてのターゲットを取得します。

delete initiator initiator_id (y/n)

指定したイニシエーターを構成から削除します。指定した削除を確定するには確認 (y) が必要です。

delete library lib_id (y/n)

指定した論理ライブラリを ACSLS 構成から削除します。これによりカスケード式の *delete* が作成され、割り当てられていたボリュームとドライブの、ライブラリとの関連付けが解除されます。ライブラリは、削除を試みる前にオフラインにする必要があります。ライブラリ内の論理ボリュームの 1 つ以上が現在使用中の場合、削除は失敗します。指定した削除を確定するには確認 (y) が必要です。

delete mapping lib_id (y/n)

指定した論理ライブラリからイニシエーターとターゲットのすべてのマッピングを削除します。この操作は、現在のクライアント接続をすべて無効化します。指定した削除を確認するには確認 (y) が必要です。

```
delete target target_id (y/n)
```

指定したターゲットを構成から削除します。指定した削除を確認するには確認 (y) が必要です。

```
display drive drive_id [-t drive_type ] [-p acs_id | all ] [-l lib_id | all ] [-u acs_id ]
```

指定したドライブまたは指定したライブラリ内のすべてのドライブのサマリーを生成します。

この表示には、ドライブの物理および論理アドレス、ドライブタイプ、ドライブの状態、ドライブステータスが生まれ、*drive_status* が *in_use* を示している場合は、マウントされているカートリッジの *VOLSER* が表示されます。

```
----- 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
-----
```

- オプションを指定せずに *display drive all* を発行すると、各物理 ACS に含まれているすべてのボリュームのリストが生成されます。
- *-t* オプションを指定すると、指定したドライブタイプのドライブのみが表示されます。
- *-p* オプションを指定すると、指定した物理 ACS 内のドライブのみが表示されます。
- *-l* オプションを含めると、指定した論理ライブラリに割り当てられているドライブのみが表示されます。*-l* オプションのあとに *all* を指定すると、構成されている論理ライブラリのそれぞれに関連付けられているすべてのドライブが表示に含まれます。
- *-u* オプションは、関連する ACS に含まれている未割り当てのドライブのみを表示します。このオプションのあとには、目的の ACS を指定する引数が続いている必要があります。

```
display initiator
```

ACSL が識別できるすべてのイニシエーターのリストを生成します。各イニシエーターは、World Wide 名と別名でリストされます。

```
display library [ -l lib_id | all ][ -p acs_id | all ]
```

リクエストされたライブラリ ID の属性を表示します。論理ライブラリの場合は `-l`、物理 ACS の場合は「`-p`」を指定します。

- `-l` を指定すると、ユーティリティーが次のように論理ライブラリ構成のサマリーを生成します。

数値 ID、名前、状態 (目的の状態)

バックアップ ACS ID、状態 (目的の状態)。

割り当て済みのインポート/エクスポートセルの数

割り当て済みのドライブスロットの数

割り当て済みのドライブの数

割り当て済みのストレージセルの数

空きセルの数

割り当て済みのボリュームの数 (これらのうちアクセス不能の数)

ボリュームラベル形式

```
-----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)
-----
```

アクセス不能なボリュームは、次のいずれかの理由で論理ライブラリからアクセスできない、割り当て済みのボリュームです。

- ボリュームが物理ライブラリから取り出し済みです。

- ボリュームがクライアントアプリケーションによって *eject* とマークされています。
- 割り当て済みのボリュームは物理ライブラリ内に存在しますが、論理ライブラリがそのストレージ容量を超えています。
- 数値 *lib_id* の代わりに引数 *all* を指定すると、構成されている論理ライブラリごとにこの表示が繰り返されます。バッチモードでは、引数を指定せずに *-1* を渡すと、引数 *all* を使用した場合と同じ結果になります。
- オプション *-p* を指定された ACS ID とともに指定すると、ユーティリティーはその ACS のライブラリ構成を要約する表示を生成します。

```
----- 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   Availability
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
-----
```

- *-p* オプションを指定して引数 *all* を渡すと、構成されている物理 ACS ごとにこの表示が繰り返されます。

all 引数は、バッチモードでのリクエストに対するデフォルトの表示です。引数を指定しない *-p* オプションは、*all* をリクエストした場合と同じ結果になります。

```
display mapping [ lib_id | all ]
```

各ライブラリ ID (または、すべてのライブラリ) のイニシエーターとターゲットのマッピングのリストを生成します。

```
display volume [ vol_id ] | vol-range ] [ -p [ acs_id | all ] [ -l
[ lib_id | all ] [ -u acs_id ]
```

指定したボリュームまたはボリュームセットのサマリー情報を生成します。

```
-----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
-----
```

- ボリュームがマウントされている場合、論理アドレスと物理的な位置には、ボリュームがマウントされているドライブのアドレスが反映されます。それ以外の場合は、ボリュームの論理および物理ホームアドレスがリストされます。論理アドレスは、論理ライブラリにボリュームが割り当てられている場合にのみ取り込まれます。
- ボリュームの範囲は、低い ACSII 値文字列と高い ASCII 値文字列をダッシュで区切って指定できます。この表示には、指定された範囲内の各ボリュームのステータスサマリーが表示されます。
- オプションを指定せずに *display volume all* を発行した場合、この表示には各物理 ACS に含まれているすべてのボリュームが表示されます。
- *-p* オプションを指定して物理 ACS ID を渡した場合、リストはその指定された物理 ACS に存在するボリュームに限定されます。*-p* を指定して *all* を発行すると、すべての物理 ACS に含まれているボリュームが表示されます。
- *-l* オプションを指定して論理ライブラリの数値 ID (*lib_id*) を渡すと、その特定の論理ライブラリに関連付けられたボリュームのセットのみが表示されます。*-l* オプションを指定して単語 *all* を渡すと、ユーティリティーは、構成されている各論理ライブラリに関連付けられているすべてのボリュームを表示します。
- *-u* オプションは、指定した物理 ACS に含まれる未割り当てのボリュームに限定された表示を生成します。

```
eject cap <cap_id> [-verbose] volume <vol_id...vol_id> | file <path
_to_volume_list>
```

lib_cmd eject は、長い (42 より大きい) ボリュームリストを指定する機能を含む、取り出し操作をサポートしています。ボリュームは、ユーザーの入力として、またはボリュームリスト

が含まれているテキストファイルへのパスを指定することによって入力できます。複数のボリュームは、大文字または小文字で 1 行に入力できます。

- `<cap_id>`
 - これは、ライブラリ構成内の特定の既存の CAP である必要があります。
 - ワイルドカードはサポートされていません。
 - CAP のリストが使用できます。
- `[-verbose | -v]`
 - `verbose` オプションは、すべてのボリュームを表示することを指定します。
 - デフォルトは `verbose`「ではなく」、最大 10 個プラス 1 のボリュームが表示されます。
- `<vol_id>`
 - 1 行に 1 つまたは複数のボリューム識別子を指定できます。
 - 対話型モードで実行されているときは、`<Return>` を押してリストを終了します。
- `<path_to_volume_list>`
 - テキストファイルの相対またはフルパスを指定します。
 - 1 行に複数のボリューム識別子を指定できます。
 - すべてのコメント行 (`#` で始まる) または空白行は無視されます。

どちらの形式でも指定できるボリュームの数に既知の制限はありません。1 つの取り出し操作は、いったんすべてのボリュームが指定されたあとで ACSLS によって実行され、可能性のあるすべてのボリュームが取り出されるまで、必要に応じて指定された CAP が入力されません。

出力の表示には、ボリューム数のレポート (指定したボリューム、取り出し済みのボリュームと取り出されていないボリューム、またはそのいずれかのボリューム) が含まれます。`verbose` オプションは、すべてのボリュームを示す出力を生成します (デフォルトでは、リストごとに 10 個のボリュームのプレビューが生成されます)。取り出されていないボリュームの場合は、ボリューム識別子と理由を示す行が表示されます。

```
vary drive lib_id drive_id [online|offline|diagnostic]
```

論理ドライブの目的の状態を、指定した (オンライン、オフライン、診断) 状態に変更します。

```
vary library lib_id [online|offline|diagnostic]
```

論理ライブラリの目的の状態を、指定した (オンライン、オフライン、診断) 状態に変更します。

バッチモードでの lib_cmd の使用

`lib_cmd` をバッチモードで使用すると、より複雑な管理タスクを実行できます。たとえば、複数のボリュームまたは複数のドライブを割り当てまたは割り当て解除する操作は、多くの場合バッチモードでもっとも効率的に実行されます。

この例では、SL8500 モジュールのレーン 2 のすべての 9840 ドライブを論理ライブラリ 1002 に割り当てます。

1. `lib_cmd` を使用して、acs-0 のすべての未割り当てのドライブをリストし、`grep` を使用して、レーン 2 (acs-0, lsm-1) 上に存在する 9840 ドライブのみをフィルタします。

```
$ lib_cmd display drive -u 0 | grep 9840 | grep 0,1,.,.
```

レーン 2 のすべての未割り当てのドライブが、5 つのフィールドの表示でリストされます。

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	

ここではこの表示からドライブ ID のみが必要なため、`awk` を使用して最初のフィールドのみを出力します。

```
$ 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. 同じ `awk print` 文に、ドライブを論理ライブラリ 1002 に割り当てするのに必要なコマンドを構築するテキストを追加できます。

```
$ 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. コマンドテキストに満足したら、ここで出力をテキストファイルに送信できます。

```
$ lib_cmd display drive -u 0 | grep 9840 | grep 0,1,.. | awk '{print "assign
drive "$1" 1002"}' > /tmp/assignDr
```

4. 最後に、コマンドファイルに加えた変更を編集してから、*lib_cmd* を使用してファイルを実行できます。

```
$ 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. *lib_cmd display* コマンドを使用して、割り当てを確認します。

```
$ 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
```

付録A ACSLS のバックアップおよび回復ツール

この付録では:

- 各ユーティリティー、その使用目的、およびそれらが重要である理由について説明します。
- 障害回復シナリオについて概要レベルで説明します。

ACSLS バックアップツール

ACSLS は、そのデータベースと ACSLS 制御ファイルの両方をバックアップするための、3つの堅牢な、明らかに異なる方法を提供します。各ユーティリティーは異なる機能を実行し、すべての方法が完全な障害回復計画における重要な役割を果たします。

自動バックアップ

ACSLS は自動データベース保護サービスを提供します。これらの自動保護サービスは、ACSLS データベースの日常の操作を、意図しない結果をもたらす可能性のある変更、またはデータベースの破損から保護します。

これらの自動バックアップ保護サービスの結果として、現在から保持期間の終わりまでのバックアップ時間までデータベースを復元できます。復元ツールについては、この付録のあとのほうで説明します。

このセクションでは、自動バックアップの方法とそれらを使用する理由について説明します。

- ACSLS のデフォルトのバックアップディレクトリ

ACSLS の初期インストール中に、バックアップに使用するディレクトリの名前を指定するよう求められています (デフォルトでは `/export/backup`)。このディレクトリ内でバックアップアクティビティーが行われます。

- 完全なデータベースバックアップが実行され、日付の命名規則を使用してこのディレクトリ内に格納されます。

```
/export/backup/yyyy-mm-dd-hh:mm:ss.tar.
```

日次バックアップを実行する時間は、`acsss_config` 内で「Automatic Backup Variables」を変更することによって変更できます。

デフォルトのバックアップ動作の変更の詳細については、6章「[ACSL S の動作を制御する変数の設定](#)」を参照してください。

- データベースの保持期間

自動バックアップに影響を与える ACSLS 内のもう 1 つの構成可能なパラメータは、データベースの保持期間です。これは、ACSL S がバックアップを保持する時間の量として定義されます。

保持期間のデフォルトは 8 日です。

デフォルトのバックアップ動作の変更の詳細については、6章「[ACSL S の動作を制御する変数の設定](#)」を参照してください。

保持期間は `acsss_config` を使用して変更することもできます。

手動バックアップ

ACSL S は、コマンド行を使用してクリティカルな ACSLS データをバックアップする、`bdb.acsss` と呼ばれるユーティリティを備えています。これは、環境が同じまたは同一のハードウェア、OS レベル、および ACSLS バージョンで構成される ACSLS データベースを復元するために使用する方法でもあります。「[bdb.acsss](#)」を参照してください。

コマンド行オプションなしで使用される `bdb.acsss` には、データベースのバックアップを作成し、それをデフォルトのバックアップディレクトリに格納する機能があります。すべてのクリティカルな ACSLS データベースと ACSLS 制御ファイルは、単一のファイルにバックアップされます。そのあとでこのファイルを使用して、内蔵ディスクまたはマザーボードの障害などのシナリオで、ACSL S を同じまたは同一のハードウェアでの前の状態に復元できます。

`rdb_acsss` ユーティリティでは、ファイルと場所 (`rdb.acsss -f /path/my_file`) またはテープデバイス (`-f /dev/rmt/0mn`) のいずれかを指定するために使用できる「`-f`」オプションを使用できます。テープデバイスの使用中は、テープデバイス上のファイル名を指定しません。

手動でのデータベースのエクスポート

ACSL S は、ACSL S データベース、ACSL S 制御ファイル、およびカスタマイズされた動的変数をエクスポートする、`db_export.sh` と呼ばれるユーティリティを備えています。`db_export.sh` ユーティリティは、カンマで区切られた ACSII ファイルへの ACSLS データベースのダンプ、ACSL S 制御ファイルのコピー、動的変数のコピーを行います。これは、新しいバージョンの ACSLS への移行に使用する方法であり、エクスポートを実行する前に

ACSLS とデータベースを両方とも停止する必要があるため、日次バックアップ操作には推奨されません。

`db_export.sh` コマンド行ユーティリティーは、異なるレベルのサーバーハードウェア、OS バージョン、および異なるリリースの ACSLS の間でのデータベースの移行に望ましい方法です。オプションを指定しない場合は、`/dev/0mn` などのローカルのデフォルトのテープデバイスで使用できます。そのあとでこのテープを任意の場所に移動でき、ACSLS およびその関連付けられた ACSLS 制御ファイルを任意の OS バージョンまたはレベルの ACSLS に復元できます。

注:

どのテープデバイスでも選択できますが、巻き戻しなしのデバイスを使用する必要があります。`db_export` ユーティリティーは 2 つのファイルを作成します。巻き戻しデバイスを選択した場合は、最初のファイル (データファイル) が 2 番目のファイルの作成時に上書きされることがあります。

`bdb.acsss` ユーティリティーと同様に、「`-f`」オプションを使用してシステムデフォルト以外のテープデバイスを指定できます。このオプションを使用するには、接続されている任意のテープデバイスに対して

`db_export.sh /dev/0mn` を実行するだけです。

`-f` では、名前を付けたファイルにデータベースをエクスポートすることもできます。この方法を使用すると、2 つのファイル (名前を付けた 1 つのファイルと、`misc` 拡張子を持つもう 1 つのファイル) が作成されることがわかります。インポートを確実に成功させるためには、インポートが行われるサーバーにファイルを両方とも転送する必要があります。

`-f` オプションを指定しても指定しなくても、`db_export.sh` ユーティリティーを実行すると、エクスポート先の ACSLS のバージョンを選択するよう求められます。

`db_export.sh` のメニュー選択項目は次のとおりです。

```
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
E: Exit
Please select by number (or E to exit):
```

ACSLS 回復ツール

ACSLS は、すべてのバックアップとエクスポートの復元に、2 つの異なる回復ツールを使用します。これらのツールは両方とも、メニュー方式のユーザーインターフェースと簡単に選択できるオプションを備えています。2 つのユーティリティーは次のとおりです。

- `rdb.acsss` - 自動と手動の両方のバックアップの回復ツール
- `db_import.sh` - 同じバージョンの ACSLS、別のバージョンの ACSLS、または別のハードウェアプラットフォームからエクスポートされた、データベースと ACSLS 制御ファイル、またはそのいずれかを復元します。このオプションは、カスタマイズした動的変数の回復にも使用できます。

`rdb.acsss` の使用

`rdb.acsss` ユーティリティは、自動バックアップ機能または `bdb.acsss` ユーティリティのいずれかによって作成されたバックアップを使用して、ACSLs データベースと ACSLS 制御ファイルを復元します。ACSLs 制御ファイルは `$ACLS_HOME/data` 内にあり、ACSLs 用のいくつかの異なる環境変数を定義します。これらは、アクセス制御設定、スラッシュ優先順位、拡張ストア LSM、カスタム `volrpt` 設定、およびボリューム属性 (`watch_vols` ユーティリティ用)などを指定します。

オプションと手順については、「`rdb.acsss`」を参照してください。

`db_import.sh` の使用

ACSLs は、同じバージョンの ACSLS や別のバージョンの ACSLS からエクスポートされたデータベース (別のハードウェアプラットフォームからでも可能です) を復元するための `db_import.sh` ユーティリティを備えています。`rdb.acsss` と同様に、実行するタスクを選択するための、メニュー方式の読みやすいユーザーインターフェースが備わっています。

`db_import.sh` ユーティリティをオプションを指定せずに使用することも、「`-f`」オプションに引数としてパスとファイル名を付けて指定することもできます。コマンド行からオプションを指定せずに `db_import.sh` を実行すると、ユーティリティはローカルテープデバイス上のエクスポートされたデータベースを探します。最初にエクスポートされたデータベースの有無をチェックし、有効なデータベースエクスポートファイルであることを確認してから、4つのオプションのあるメニューを表示します。

注:

`-f` オプションにデフォルト以外のデバイスのテープデバイスを含めて指定する (`-f /dev/rmt/0mn`) こともできます。任意の有効なテープデバイスを指定できますが、巻き戻しなしのデバイスを指定する必要があります。`db_import.sh` ユーティリティは 2 つのファイル (データ用に 1 つと制御ファイル用に 1 つ) を使用します。巻き戻しデバイスを使用すると、データファイルの回復後に、テープが巻き戻されて制御ファイルが失敗する場合があります。

パスとファイル名を付けて `-f` オプションを指定した場合、`db_import.sh` は、指定されたファイル名をエクスポートされたデータベースファイルとして使用します。ローカルのテープデ

バイスと同様に、最初にそのファイルが存在しているかどうかを確認するためのチェックを行い、指定されたファイル名がエクスポートされたデータベースファイルであることを検証します。指定されたファイルが有効なエクスポートの場合は、メニューが表示されます。メニューのオプションは次のとおりです。

- オプション 1 - エクスポートされたファイルのデータベーステーブル、制御ファイル、および動的変数をインポートします。

このオプションでは、ライブラリデータベースと、エクスポートされたバージョンから保存されたカスタマイズ済みの更新すべてがインポートされます。

- オプション 2 - エクスポートされたファイルからデータベーステーブルのみをインポートします。

このオプションでは、完全なライブラリ構成とボリュームデータセットがインポートされますが、エクスポートされたバージョンで行われたシステムのカスタマイズは適用されません。

- オプション 3 - エクスポートされたファイルの制御ファイルのみをインポートします。

このオプションでは現在のライブラリデータベースは変更されず、以前のバージョンからエクスポートされたカスタマイズのみがインポートされます。

- オプション 4 - エクスポートされたファイルからカスタマイズされた動的変数をマージします。

このオプションでは、エクスポートされたバージョンのカスタマイズされた設定を、現在のバージョンとマージします。[83 ページの6章「ACSLS の動作を制御する変数の設定」](#)を参照してください。

障害のシナリオ

このセクションでは、障害のシナリオについて説明します。

データベースが破損した

1. ユーザー `acsss` として、回復を実行する前に ACSLS を停止します

```
$ acsss db
$ rdb.acsss
```

2. オプション 2 を選択します。「`rdb.acsss`」を参照してください。
3. 回復が完了したら、ACSLS を起動します: `acsss enable`。

間違ったライブラリに対して `acsss_config` を実行した

1. オプション 2 を選択します。「[rdb.acsss](#)」を参照してください。
2. ACSLS を起動し、データベースのバックアップおよび復元手順に従ってテストします。

サーバー障害 – 新しいハードウェアでの同じサーバーの再構築

1. オペレーティングシステムをインストールします。
2. 前のサーバーの設定で新しいサーバーと OS を構成します。
3. ACSLS をインストールします。
4. バックアップテープまたは FTP バックアップファイルをサーバーに挿入します。
5. `rdb.acsss` ユーティリティーを起動します。
6. オプション 2 を選択します。「[rdb.acsss](#)」を参照してください。
7. `rdb.acsss` を終了します。
8. ACSLS を起動し、データベースのバックアップおよび復元手順に従ってテストします。

サーバー障害 – 新しいハードウェアでの別の ACSLS サーバーの再構築

1. オペレーティングシステムをインストールします。
2. ACSLS をインストールします。
3. ACSLS サーバー間のバックアップファイルを適切な場所に置きます。
4. `rdb.acsss` と入力します。「[rdb.acsss](#)」を参照してください
5. オプション 3 を選択します。
6. 回復ユーティリティーが完了したら、ACSLS を起動し、データベースのバックアップおよび復元手順に従ってテストします。

付録B エンタープライズライブラリの接続オプション

この章では、次の内容について説明します。

- 「概要」
- 「デュアル TCP/IP サポート」
- 「マルチ TCP/IP サポート」
- 「冗長電子装置」

概要

SL8500 および SL3000 ライブラリに ACSLS を接続するには、複数のオプションがあります。これらのオプションは、ACSLS と SL8500 または SL3000 間の通信に個別または一緒に使用できます。

連結された 1 組の SL8500 では、デュアル TCP/IP、マルチライブラリ TCP/IP、および冗長電子装置を単独で、または組み合わせて実装できます。

1 台の SL3000 または SL8500 では、デュアル TCP/IP または冗長電子装置 (RE)、あるいはその両方を実装できます。SL3000 または SL8500 には IPv4 を使用して接続できます。

接続オプションの概要は次のとおりです。

- デュアル TCP/IP

デュアル TCP/IP は、ACSLS とライブラリコントローラカード間に 2 つの別個かつ独立した TCP/IP 接続を提供します。これらの通信パスの 1 つに障害が発生した場合、ACSLS は自動的に 2 番目のパスを通信に使用します。

デュアル TCP/IP サポートを実装するには、*route* コマンドを使用して ACSLS サーバーとライブラリの両方でルーティングテーブルを定義および管理する必要があります。ACSLS サーバーとライブラリのポートペア間の通信では、これらのルーティングテーブルに定義されているネットワーク通信パスが強制的に使用されます。

SL8500 と SL3000 はどちらも、ライブラリとのデュアル TCP/IP 通信をサポートしています。

- マルチ TCP/IP サポート

マルチ TCP/IP サポートを使用すると、ACSL S サーバーは連結された 1 組の SL8500 内の複数の SL8500 ライブラリに接続できます。1 台のライブラリとの通信に障害が発生した場合、ACSL S はライブラリとの通信をほかのライブラリとの接続に自動的に送信します。ライブラリは自動的にほかのライブラリにメッセージを転送します。

マルチ TCP/IP 通信では、ACSL S サーバーまたは SL8500 ライブラリにルーティングテーブルを定義する必要がないため、構成と管理がデュアル TCP/IP 通信より簡単です。ただし、マルチ TCP/IP には連結された 1 組の SL8500 ライブラリが必要です。単一のスタンドアロン SL8500 または SL3000 ライブラリには適合しません。

- 冗長電子装置 (RE)

RE では、ライブラリコントローラカードの冗長セットを使用します。常に、1 つのセットがアクティブであり、もう一方のセットがスタンバイです。アクティブなライブラリコントローラは、ACSL S または SL コンソールからのコマンドに応答して、スタンバイにフェイルオーバーできます。ライブラリカードの障害が発生した場合は、ライブラリで自動フェイルオーバーを開始できます。

RE を使用すると、中断を最小限に抑えたライブラリファームウェア (マイクロコード) のダウンロードが可能になります。連結された 1 組の SL8500 内で、ライブラリごとに RE を実装できます。コンプレックス内のいずれかまたはすべてのライブラリに RE を実装できます。

ライブラリで RE をサポートするには、ACSL S 7.3.1 または 8.0.2 以降が必要です。

ACSL S とライブラリの通信ステータスの表示

ACSL S と管理対象ライブラリの通信のステータスを表示およびモニターするには、*query Imu* コマンドを使用します。*query Imu* コマンドは、ライブラリへの ACS およびポートの接続ステータスも表示します。

デュアル TCP/IP サポート

デュアル TCP/IP は、SL8500 および SL3000 ライブラリ (ここではライブラリと呼ぶ) に購入できるオプションです。ライブラリに 2 つの TCP/IP 接続を提供します。ただし、2 つの接続の 1 つのみを動作させてライブラリを使用することもできます。

デュアル TCP/IP の目的は、障害の発生した通信パスを自動的に認識し、回避することです。これは自動的に行われるため、動作不能な接続から手動で切り替える必要はありません。

ライブラリでデュアル TCP/IP サポートを使用するには、*route* コマンドを使用して ACSLS サーバーとライブラリの両方のルーティングテーブルを管理する必要があります。これにより、ライブラリ上に定義されたネットワークインタフェースへのルートが強制的に使用され、基本的には、インタフェース間に 1 対 1 の関係が作成されます。顧客システム管理者 (CSA) が ACS サーバー上のルーティングテーブルを変更し、顧客システムエンジニア (CSE) がライブラリ上のルーティングテーブルを更新します。UNIX の *route* コマンドの詳細は、ACSLS サーバーのマニュアルページを参照してください。

要件

- システム管理者とネットワーク管理者の両方と協力して、現在のネットワーク環境について理解し、必要なすべての IP アドレスを事前に識別します。
- システム管理者と協力して、ネットワークインタフェースを構成するか、またはそれが正しく構成されていることを検証します。

構成

ACSLS はすべてのアクティブな接続を使用するため、ACSLS からライブラリに対して 2 つの接続を開いた状態に保つことをお勧めします。1 つの接続が動作不能になった場合、ACSLS は残りの動作可能な接続を使用しながら、障害の発生した接続上で通信の再確立を試行し続けます。

デュアル TCP/IP 実装では、シナリオ 1 に示すように、ACSLS サーバーの 2 つのネットワークインタフェースを 2 つの別々のサブネット上に構成することが推奨されています。これにより、ネットワーク通信における最大限のスループットと最小限のリソース競合が実現するほか、予備の物理接続も提供されるため信頼性が向上します。

単一のライブラリに対する 2 つの TCP/IP 接続を構成するには、*acsss_config* ユーティリティまたは動的構成 (*config*) を使用します。ライブラリへの接続数 (2) およびネットワークデバイスの IP アドレスを入力します。SL3000 は、IPv4 接続をサポートしています。

次のシナリオは、ACSLS サーバーの構成例を示しています。ライブラリのデュアル TCP/IP 機能の構成手順については、該当するライブラリシステムのデュアル TCP/IP 機能に関するドキュメントを参照してください。

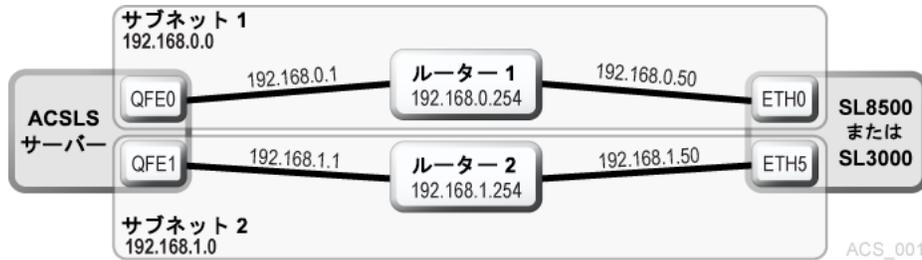
次のシナリオではプライベートサブネット IP アドレスを使用しており、実際の環境では異なります。これらのシナリオでは、ネットワークデバイスが構成されていて、正常に機能していることを前提としています。

シナリオ 1 - 推奨される構成

シナリオ 1 はデュアル TCP/IP 機能の推奨される構成です。

この構成では、ACSL S サーバーに搭載された 2 つのネットワークインタフェースが 2 つの別々のサブネット上に存在します。SL8500 または SL3000 の 2 つのネットワークインタフェースは、ACSL S サーバーと同じ 2 つのサブネット上にあります。

図B.1 推奨される構成



このシナリオにおいて、ライブラリでは、次のように ACSLS サーバー上のネットワークインタフェースとの 1 対 1 の関係が使用されます。

- ACSLS サーバー上の qfe0 インタフェースは、SL8500 または SL3000 上の eth0 インタフェースとのみ通信します。
- ACSLS 上の qfe1 インタフェースは、SL8500 または SL3000 上の eth5 とのみ通信します。

UNIX の `route` コマンドを使用して、この関係を強制的に適用します。

- Solaris の場合: `root` ユーザーとして、次のコマンドを入力します。

```
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
```

最初の `route` コマンドは、`192.168.0.50` とのすべての通信が ACSLS サーバー上の `qfe0` を経由してからルーター 1 を経由するようにルーティングします。

2 番目のコマンドは、`192.168.1.50` とのすべての通信が ACSLS サーバー上の `qfe1` を経由してからルーター 2 を経由するようにルーティングします。

次のように入力すると、ルートがルーティングテーブル内に存在することを検証できます。

```
# netstat -r
```

例B.1 IPv4 ルーティングテーブル

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

最初の2つのエントリは、先程追加したものです。192.168.0.50 とのすべての通信は QFE0 を経由し、192.168.1.50 との通信は QFE1 を経由します。

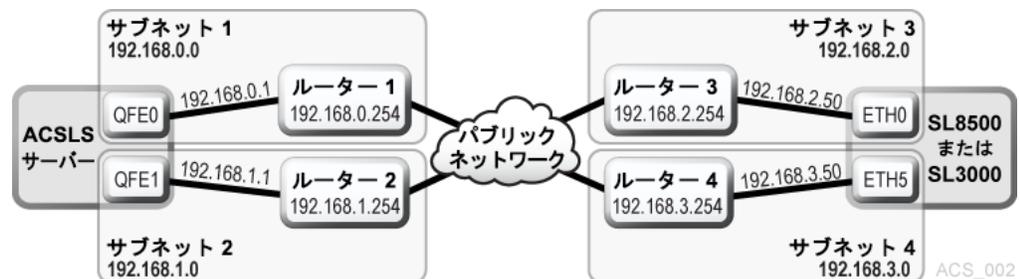
注意: StorageTek SL8500 モジュラーライブラリシステムのデュアル TCP/IP 機能に関するドキュメントの説明に従って、ライブラリのルーティングテーブルを構成してください。

シナリオ 2

シナリオ 2 は次のとおりです。

- ACSLS サーバーの2つのインタフェースが、ライブラリとは別のサブネット上にあります
- SL8500 または SL3000 ライブラリの2つのネットワークインタフェースが、ACSLS とは別のサブネット上にあります
- ACSLS と SL8500 または SL3000 は、両方ともパブリックネットワークを使用します

図B.2 ACSLS と SL8500 または SL3000 でパブリックネットワークを使用



UNIX の route コマンドを使用して、この関係を強制的に適用します。

- Solaris の場合: root ユーザーとして、次のコマンドを入力します。

```
#route add 192.168.2.50 -ifp qfe0 192.168.0.254
```

```
#route add 192.168.3.50 -ifp qfe1 192.168.1.254
```

ACSLS のデフォルトルートは同じままです。サブネット内のルートは、パブリック LAN 経由でライブラリに通信をルーティングする方法を認識します。また、インタフェース間の 1 対 1 の関係が引き続き強制的に適用されています。この場合も、次のコマンドを使用してこれを確認できます。

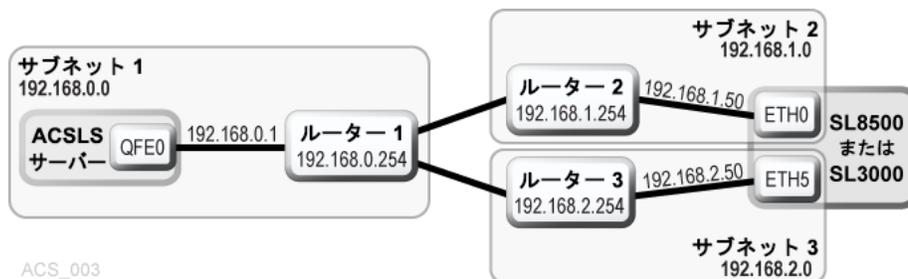
```
# netstat -r
```

注意: StorageTek SL8500 または SL3000 モジュールライブラリシステムのデュアル TCP/IP 機能に関するドキュメントの説明に従って、ライブラリのルーティングテーブルを構成してください。

シナリオ 3

このシナリオでは、1 台の ACSLS サーバーの 1 つのネットワークインタフェースが、別個のサブネット上にあります。SL8500 または SL3000 ライブラリの 2 つのネットワークインタフェースが、ACSLS サーバーとは別の 2 つのサブネット上にあります。

図B.3 2 つのネットワークインタフェースを備えた SL8500 または SL3000



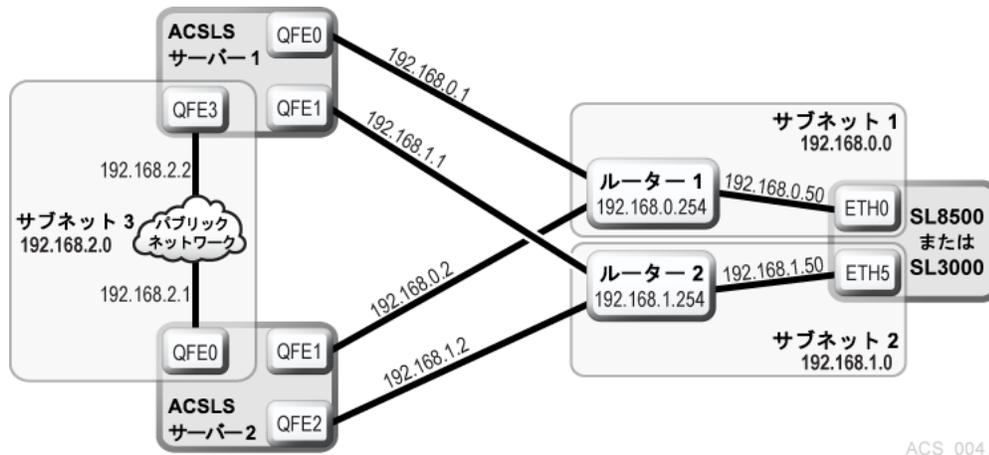
注意: StorageTek SL8500 または SL3000 モジュールライブラリシステムのデュアル TCP/IP 機能に関するドキュメントの説明に従って、ライブラリのルーティングテーブルを構成してください。

シナリオ 4

シナリオ 4 は次のとおりです。

- 2 台の高可用性 (ACSLS HA) サーバーが、どちらも 3 つのネットワークインタフェースを備え、SL8500 または SL3000 との間に 2 つの別々のプライベートサブネットを使用し、さらに 1 つのパブリックネットワークを使用します。
- 1 台の SL8500 または SL3000 ライブラリの 2 つのネットワークインタフェースが、ACSLS サーバーと同じ 2 つのプライベートサブネット上にあります。

図B.4 ACSLS HA



このシナリオで、ACSLS HA には、ネットワークインタフェースがそれぞれ異なる 2 つの別々のサーバーが使用されます。したがって、両方の ACSLS サーバーに、カスタムルートのエントリを追加する必要があります。

Solaris ユーザーの場合:

- ACSLS サーバー 1 では、次のように入力します。

```
route add 192.168.0.50 -ifp qfe0 192.168.0.254
```

```
route add 192.168.1.50 -ifp qfe1 192.168.1.254
```

- ACSLS サーバー 2 では、次のように入力します。

```
route add 192.168.0.50 -ifp qfe1 192.168.0.254
```

```
route add 192.168.1.50 -ifp qfe2 192.168.1.254
```

ライブラリの構成に両方のサーバーの IP アドレスを追加する必要があります。*StorageTek SL8500* または *SL3000* モジュラーライブラリシステムのデュアル TCP/IP 機能に関するドキュメントを参照してください。

ACSLS HA では、ライブラリのネットワークインタフェースを 2 つの別々のサブネットに分割することが重要です。高可用性環境の目的は、冗長性を組み込み、単一障害点を排除することです。

注意: *StorageTek SL8500* または *SL3000* モジュラーライブラリシステムのデュアル TCP/IP 機能に関するドキュメントの説明に従って、ライブラリのルーティングテーブルを構成してください

カスタマイズされたルーティングテーブルのエントリをリブート後も保持する

カスタマイズされたルーティングテーブルのエントリは、システムのリブート後に失われます。これはシステムのルーティングテーブルの特性であり、予期された動作です。

SL8500 または SL3000 でデュアル TCP/IP 機能をサポートするには、ACSL S サーバー上のルーティングテーブルにカスタムエントリを追加する必要があります。ACSL S サーバーのリブート時に、すべてのルーティングテーブルエントリがフラッシュされ、ライブラリへの必要なルートがすべて削除されます。これはオペレーティングシステムの特性なので、この状況に対処するには 2 つの方法があります。

スクリプトの作成

ブート時に初期化されるカスタムルートを追加するスクリプトを作成できます。手順については、「[ブート時に初期化されるカスタムルートの追加](#)」を参照してください。

その後、これらのスクリプトを `rc` ディレクトリ構造に配置し、ブート時に自動実行されるようにします。これを実装する最善の方法の詳細については、システムのドキュメントを参照してください。

ACSL S の起動スクリプトを使用して、ブート時にカスタムルーティングエントリを追加します。起動スクリプトは、カスタマイズされたルーティングテーブルのエントリを含むファイルを確認します。見つかったすべてのエントリが、UNIX の `route` コマンドを使用して自動的にルーティングテーブルに追加されます。スタンドアロンの ACSL S インストールの場合、ライブラリのサポートに必要なルートエントリを管理するには、これが望ましい方法です。

重要: 高可用性 ACSL S (ACSL S HA) 環境の ACSL S インストールでは、このソリューションは機能しません。

この場合は、最初の方法を使用してルーティングテーブルを管理する必要があります。

ACSL S HA は、クラスタ化されたリソースの管理に Solaris Cluster を利用するため、ブート時にシステムの RC メカニズムによって ACSL S を自動的に起動することはできず、スタンドアロンの ACSL S サーバーとは異なる方法でシステムの初期化を処理します。これは Solaris Cluster エージェントによってのみ処理され、S87ACSL S 起動スクリプトが使用されることはありません。適切な「`route add`」コマンドを含むスクリプトを追加し、それを `/etc/rc2.d` ディレクトリ構造に配置します。ACSL S HA 環境のユーザーは Oracle Advanced Customer Support を依頼することを強くお勧めします (ACSL S HA システムを最初に設置した同じコンサルタントが望ましいでしょう)。

ブート時に初期化されるカスタムルートの追加

カスタムルーティングエントリを追加するには:

1. 次のディレクトリに移動します。

```
$ACS_HOME/data/external/ custom_routing.
```

このディレクトリには、テンプレートファイル *custom_routing_tables.tpl* が含まれています。

2. このファイルをコピーし、ファイル名を *custom_routing_tables* に変更します。

```
# cp custom_routing_tables.tpl custom_routing_tables
```

3. *custom_routing_tables* ファイルを編集 (vi) してエントリを追加します。

このファイルには 3 つのフィールドが含まれています。

- SL8500 または SL3000 の IP アドレス。
- 1 対 1 の関係を確立したい ACSLS サーバー上のインタフェースの名前。
- サブネットのデフォルトルートの IP アドレス。

4. 形式については、*custom_routing_tables* のコメントセクションの説明に従います。

注:

空白行が存在しないことを確認してください。

サーバーがリブートすると、ACSLs は自動的に初期化され、カスタムルートがルーティングテーブルに追加されます。

5. 次のコマンドを使用して、ルーティングテーブル内のすべてのルートを確認します。

```
# netstat -r
```

route および *netstat* コマンドの詳細については、UNIX のマニュアルページを参照してください。

ルーティングコマンドの削除

以前の構成に不要になった、または誤って追加されている特殊なルーティングコマンドを削除するには、*route* コマンドを使用します。

例: *root* ユーザーとして、次のコマンドを入力します。

```
# route delete 192.168.0.50 192.168.0.254
```

これは、デフォルトルート `192.168.0.254` を使用している `192.168.0.50` (SL8500 または SL3000) へのルートを削除するように指示しています。ルートが削除されます。

マルチ TCP/IP サポート

SL8500 3.97 以降のファームウェアがインストールされている場合、ACSLs は ACS (ライブラリコンプレックス) 内の複数台の SL8500 に接続できます。

ACSLs は最大 15 の ACS への接続をサポートします。たとえば、4 台の SL8500 への 15 の接続、2 台の SL8500 それぞれへの 2 つの接続、1 台の SL8500 への 2 つの接続ともう 2 台の SL8500 への 2 つの接続、2 または 3 台のライブラリへの 3 つの接続などの場合が可能です。

ACSLs が複数のライブラリに接続する場合、冗長性を得られるように別々のサブネットで接続してください。1 つのサブネットで障害が発生しても、ACSLs とライブラリ間の通信はその他のサブネットを介して続きます。

ACSLs から 1 つの SL8500 HBC カードへの接続が 2 つある場合、「[デュアル TCP/IP サポート](#)」の説明に従って、SL8500 および ACSLS サーバーのルーティングテーブルを構成する必要があります。ACSLs サーバーと各 SL8500 HBC カード間の接続が 1 つしかない場合は、ACSLs および SL8500 のルーティングテーブルを構成する必要はありません。

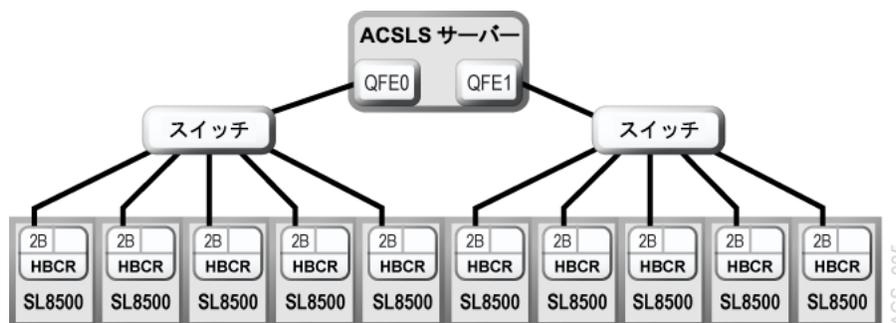
ライブラリのパフォーマンスを最適化し、SL8500 どちらのライブラリ間通信を最小限に抑えるには、最初の接続 (ポート 0) を、もっともアクティビティの多いライブラリに対して定義します。

マルチ TCP/IP 通信では、ACSLs サーバーまたは SL8500 ライブラリにルーティングテーブルを定義する必要がないため、構成と管理がデュアル TCP/IP 通信より簡単です。ただし、マルチ TCP/IP には連結された 1 組の SL8500 ライブラリが必要です。単一のスタンドアロン SL8500 または SL3000 ライブラリには適合しません。

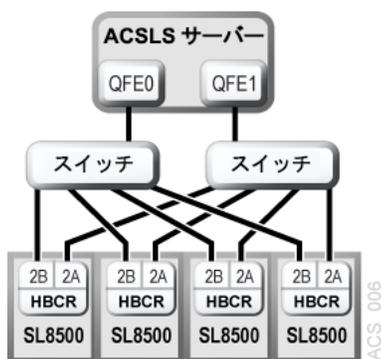
詳細については、*StorageTek SL8500 モジュラーライブラリシステム*のホストとライブラリの通信に関する技術資料を参照してください。

[図B.5「マルチ TCP/IP を使用した ACSLS」](#) はマルチ TCP/IP 構成を使用した ACSLS を示し、[図B.6「マルチ TCP/IP およびデュアル TCP/IP を使用した ACSLS」](#) はマルチ TCP/IP およびデュアル TCP/IP 構成を使用した ACSLS を示しています。

図B.5 マルチ TCP/IP を使用した ACSLS



図B.6 マルチ TCP/IP およびデュアル TCP/IP を使用した ACSLS



冗長電子装置

オプションの SL8500 または SL3000 冗長電子装置 (RE) 機能は、エンタープライズライブラリ内のフェイルオーバー保護を提供します。ライブラリコントローラでエラーが発生した場合は、動作が自動的に代替ライブラリコントローラに切り替わるため、ライブラリおよびホストの動作の中断が最小限になります。これにより、ライブラリでは通常の動作を継続しながら、Oracle サポート担当者が故障したカードを交換できるようになります。

さらに、RE を使用すると、ファームウェアアップグレード中のライブラリ操作の中断が最小限に抑えられます。

注:

ライブラリでは、ロボットや電源システムといったさまざまなコンポーネントの冗長性が提供されています。「冗長電子装置」という用語は、特にライブラリおよびドライブコントローラコンポーネントの冗長性を指します。

RE には、次のハードウェアコンポーネントがすべて必要です。

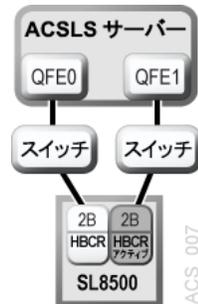
- アクティブなライブラリコントローラ (HBC または HBCR) とアクティブなドライブコントローラ (HBT) のペア

- スタンバイ HBC または HBCR とスタンバイ HBT のペア
- その他の冗長コンポーネント

詳細は、StorageTek SL8500 または SL3000 のユーザーズガイドを参照してください。

図B.7「RE を使用した ACSLS」は、単一のライブラリで RE を使用した ACSLS を示しています。

図B.7 RE を使用した ACSLS

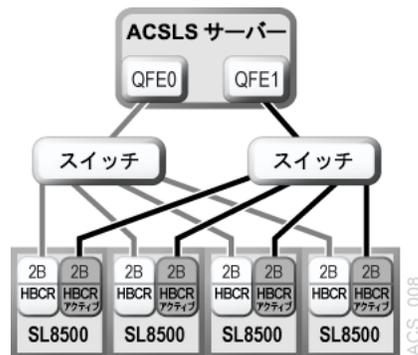


ACSLS での RE のサポート

ACSLS は、単一のライブラリコンプレックス (パススルーで接続されたライブラリの ACS) 内に混在している、アクティブおよびスタンバイの SL8500 ライブラリコントローラ (LC) カードを処理します。

図B.8「RE とマルチ TCP/IP を使用した ACSLS」に示すように、各 SL8500 のいずれかの HBCR カードをアクティブコントローラカードにすることができます。

図B.8 RE とマルチ TCP/IP を使用した ACSLS



連結された 1 組の SL8500 内で、各ライブラリが独自にライブラリコントローラの冗長ペアを持つことができるようになりました。1 つのライブラリコンプレックス内で、一部のライブラリ

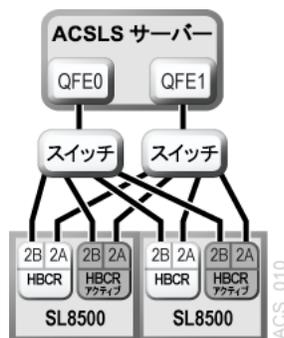
にはライブラリコントローラカードのペアを備えて RE を有効にする一方で、ほかのライブラリにはライブラリコントローラを 1 つだけ備えることができます。ACSL S は同時にすべてのアクティブ LC と通信できます。

ACSL S では、[図B.9「RE とデュアル TCP/IP を使用した ACSLS」](#)のような RE とデュアル TCP/IP の組み合わせ、または、[図B.10「RE とデュアル TCP/IP およびマルチ TCP/IP」](#)のような RE とデュアルおよびマルチ TCP/IP の組み合わせがサポートされています。

図B.9 RE とデュアル TCP/IP を使用した ACSLS



図B.10 RE とデュアル TCP/IP およびマルチ TCP/IP



マウントおよびマウント解除の照会と再試行

RE をサポートするために、ACSL S では、ライブラリおよびドライブが一時的に停止した際のマウントおよびマウント解除の照会と再試行が実装されました。詳細については、「[ライブラリが一時的に使用不可能になっている場合のマウントおよびマウント解除のキューへの追加と再試行](#)」を参照してください。

単一ライブラリのみの switch lmu

`switch lmu` コマンドは、1 台の SL3000 または単一の SL8500 ライブラリに含まれているライブラリコントローラ間で強制的に切り替えを行う場合に使用できます。`switch lmu` コマン

ドは、ライブラリコンプレックス内でほかの SL8500 に連結されている SL8500 の切り替えには使用できません。

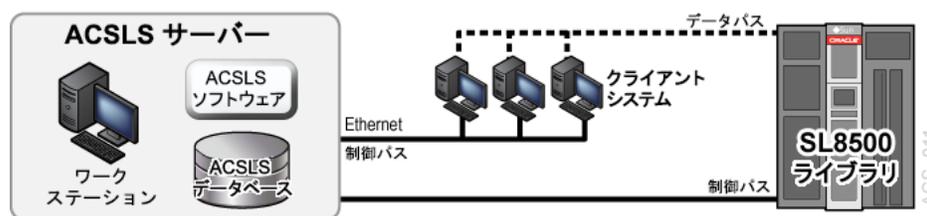
付録C SL8500 の ACSLS サポート

この付録は、次のトピックについて説明します。

- 「複数の SL8500 へのマルチ TCP/IP を使用した接続」
- 「SL8500 内部アドレスと ACSLS アドレスについて」
- 「SL コンソールを使用したアドレス変換」
- 「パーティションからセルを取り外す前のカートリッジの移動」
- 「SL8500 CAP」
- 「エレベータおよび PTP アクティビティを最小限に抑える」
- 「ワークロードをサポートするためのテープドライブの構成」
- 「カートリッジの場所の管理」
- 「見つからないカートリッジを見つける」
- 「SL8500 のオフラインへの変更」
- 「動的構成 (config) ユーティリティの使用」
- 「SL8500 の拡張」
- 「パススルーポートを使用した SL8500 の接続」

次の図は、ACSLS サーバーを伴った SL8500 ライブラリを示しています。

図C.1 SL8500 用に ACSLS を構成する前の ACSLS サーバーを伴う SL8500



SL8500 ライブラリは、Ethernet 物理インタフェースを介した TCP/IP プロトコルを使用して、ホストおよび ACSLS を管理しこれらと通信します。このインタフェースにより、ACSLS は SL8500 に接続し、通信できます。SL8500 用に ACSLS を構成する前に:

- 1 台以上の SL8500 を ACSLS に接続します
- SL8500 のすべてのコンポーネントが動作していることを確認します。

ACSLs は、ライブラリによって報告される情報から、そのライブラリ構成を構築します。SL8500 コンポーネントが動作していない場合は、ライブラリ情報が ACSLS に報告されない可能性があり、SL8500 の ACSLS 構成が不完全になります。

注:

ドライブや CAP などのコンポーネントが動作していない場合は、動的構成 (`config acs`、`config lsm` または `config drives`) を使用して、ACSLs が実行しライブラリがオンラインである間に、これらのコンポーネントを簡単に追加または更新できます。

複数の SL8500 へのマルチ TCP/IP を使用した接続

SL8500 3.97 以降のファームウェアがインストールされている場合、ACSLs は ACS (ライブラリコンプレックス) 内の複数台の SL8500 に接続できます。

ACSLs は最大 15 の ACS への接続をサポートします。たとえば、4 台の SL8500 への 15 の接続、2 台の SL8500 それぞれへの 2 つの接続、1 台の SL8500 への 2 つの接続ともう 2 台の SL8500 への 2 つの接続、2 または 3 台のライブラリへの 3 つの接続などの場合が可能です。

ACSLs が複数のライブラリに接続する場合、冗長性を得られるように別々のサブネットで接続してください。1 つのサブネットで障害が発生しても、ACSLs とライブラリ間の通信はその他のサブネットを介して続きます。

ACSLs から 2 つの接続で 1 枚の SL8500 HBC カードに接続している場合は、「[概要](#)」の説明に従って、SL8500 と ACSLS サーバーのルーティングテーブルを構成します。ACSLs サーバーと各 SL8500 ライブラリ間の接続が 1 つしかない場合は、ACSLs と SL8500 のルーティングテーブルの構成は必要ありません。

ライブラリのパフォーマンスを最適化し、SL8500 間のライブラリ間通信を最低限に抑えるには、もっともアクティビティの高いライブラリに接続します。`acsss_config` または `config acs` で指定する最初の接続を、もっともアクティビティの高い SL8500 への新しい接続にします。

詳細は、*SL8500 Modular Library System の技術的な概要 - ホストとライブラリ間の通信* を参照してください。

SL8500 のすべてのコンポーネントが動作していることの確認

SL8500 のすべてのコンポーネントが動作していることを確認するには:

1. StorageTek ライブラリコンソール (SL コンソール) にログオンします。

SL8500 のコンソールとリモートライブラリコンソールのどちらも使用できます。

2. 「Tools」 -> 「System Detail」を選択します。

- すべての SL8500 コンポーネントが緑色になっている必要があります。

例外: 黄色のドライブは、動的構成を使用して現時点でまたはあとから構成できます (「bdb.acsss」)。

- 見つからないコンポーネントは、動的構成 (`config acs` または `config lsm`) ユーティリティーを使用して追加できます。
- 重要: SL8500 を構成する前に、エレベータ (エレベータフォルダ) は緑色になっている必要があります。エレベータが緑色でない場合は、ACSLS に対して SL8500 を構成しないでください。エレベータは論理パススルーポート (PTP) です。PTP がない場合、SL8500 レールが接続されていることが ACSLS にはわかりません。

3. SL8500 コンポーネントが動作可能になったら、「CSI チューニング変数の設定」または「acsss マクロ」の説明に従って、ACSLS に対して SL8500 を構成します。

SL8500 内部アドレスと ACSLS アドレスについて

SL8500 の内部アドレスと、ACSLS および HSC でサポートされているその他のライブラリの内部アドレスとは、違いがあります。

- SL8500 は 1 を基準にしており、負の数を使用します。
- その他のライブラリは 0 を基準にしており、負の数は使用しません。
- SL8500 は、ライブラリ、レール、列、サイド、および行の 5 つのパラメータを使用します。
- レガシー StorageTek ライブラリ (9310 など) は、ACS、LSM、パネル、行、および列を使用します (HLI-PRC)。

表C.1 アドレス指定の説明

HLI-PRC	SL8500	説明
ACS	ライブラリ	ライブラリコンプレックス内の特定の SL8500 ライブラリの番号。ACS は SL8500 ライブラリコンプレックスです。ライブラリコンプレックスには複数の SL8500 を配置できます。
LSM	レール	SL8500 ライブラリには、HandBot が移動するレールが 4 つあり、上から下に 1-4 (1 を基準) の番号が付けられています。
LSM 0	レール 1	
LSM 1	レール 2	
LSM 2	レール 3	

LSM 3	レール 4	ACCLS では、それぞれのレールは個別の LSM とみなされ、上から下に 0-3 (0 を基準) の番号が付けられています。
パネル パネル 0 パネル 1 パネル 2-n	列 CAP ドライブ ストレージスロット	列は、ライブラリ内の水平方向の場所を示します。ライブラリの前面から見ると、列およびパネルの番号は、ドライブパネルの中央 (1) から開始し、先にスイープするごとに番号が増加します。 (SL8500 はアドレスとしてパネルを使用しません。) HLI パネルは、各 LSM の両サイド (左と右) と両ウォール (内部と外部) を含むライブラリの幅全体にわたります。
	サイド	ウォールの場所: 外部ウォール 内部ウォール HandBot 番号: 左 (-) 右 (+)
行	行	行は、テープカートリッジの垂直方向の場所を示し、上から下に番号が付けられます。
列		HLI アドレスの行は次のとおりです。 ストレージパネルは 2 から始まり、列 0 = 左 および列 1 = 右です 行 0-12 外部ウォール 行 13-26 内部ウォール 通常のストレージパネル内の各列には 27 の行があります。 パネル当たり 54 のカートリッジの合計容量について。 SL8500 アドレスの行は次のとおりです。

ストレージスロットは
 列 -3 = 左
 列 +3 = 右から始まります

行 1 – 13 外部ウォール
 行 1 – 14 内部ウォール

- 0 を基準とした番号付け (HLI と同様) は番号付けを 0 から開始します。
- 1 を基準とした番号付け (SL8500 と同様) は番号付けを 1 から開始します。
- これが、ソフトウェア (ACSLs または HSC) とハードウェア (物理的な SL8500 アドレス) の番号付けの順序における重要な違いです。

SL コンソールを使用したアドレス変換

SL コンソール検索ユーティリティを使用して、SL8500 内部アドレスと ACSLS または HSC のパネル、行、および列を変換します。カートリッジを検索するには:

1. SL コンソールにログインします。
2. 「Tools」 > 「Diagnostics」 > 「Search」を選択します。
3. 「Location」を選択します。
4. 「Location」フィールドで、次のいずれかの操作を選択します。

contains	例: 1,1,-9 は、両サイドのすべての行に対し、ライブラリ 1、ルール 1、列 -9 内にあるコンテンツを一覧表示します
endsWith	例: 1,5 は、サイド 1、行 5 のすべてのルールおよび列のスロットコンテンツを一覧表示します
equals	例: 1,1,-9,1,1 は、その特定の場所 (L,R,C,S,W) にあるコンテンツを一覧表示します
startsWith	例: 1, 3 は、ライブラリ 1、ルール 3 にあるすべての列、サイド、および行のスロットコンテンツを一覧表示します。

5. 「Requestor」プルダウンメニューから次のいずれかを選択します。

- *default*

ライブラリ内部の物理的な場所 (セル、ドライブ、CAP)。

物理的な場所 (内部アドレス) がわかっており、HLI-PRC アドレスを検出する必要がある場合、そのアドレスを「location」に入力 (*enter*) し、リクエストとして「*default*」を選択します。

- *hli#*

これは、ライブラリ管理ソフトウェアからカートリッジの HLI-PRC アドレスを選択します。ここで # は次のいずれかです。

- hli0 はパーティション分割されていないことを示します。
- hli1-8 はパーティション分割されたライブラリを示します。ここで数値はパーティション番号です。

このオプションは、内部アドレスと *hli# Requester* の両方を表示します。

6. SL コンソールの右上にある「*Search*」ボタンをクリックします。

検索結果には、スロットタイプ (セル、ドライブ、または CAP) ごとの場所が一覧表示されます。

7. 「*Details (...)*」フィールドをクリックします。

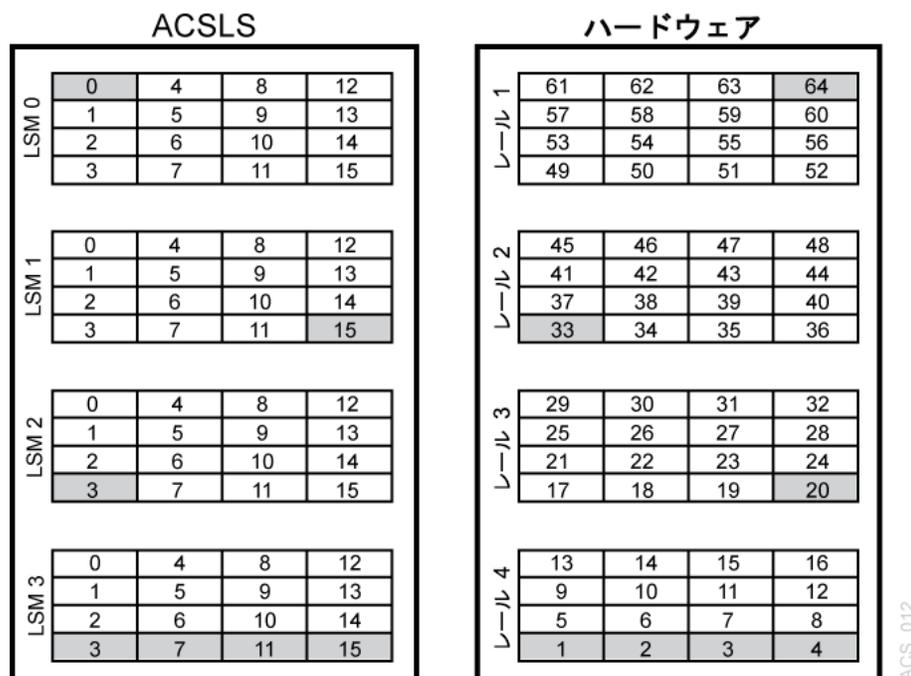
ポップアップウィンドウには、カートリッジの VOLID、メディア、カートリッジタイプ (LTO、SDLT テープ、および T シリーズ、データ、クリーニング、または診断) などの詳細が示され、場所に関する内部アドレスと HLI アドレスの両方が表示されます。

テープドライブの場所

テープドライブは、LSM に関連付けられ、これに属しています。別の LSM にカートリッジテープをマウント (*mount*) するには、ドライブへの内部パススルー操作 (この場合はエレベータ) をカートリッジに行う必要があります。

次の表には、内部のソフトウェアマッピング (ライブラリ内部のテープドライブから見た場合)、およびドライブの外部の物理的な番号付け (外部のドライブおよび電子モジュールの背面から見た場合) が表示されます。

図C.2 ソフトウェアおよび物理ドライブの番号付け



強調表示されているドライブは、一致するドライブを示します。例:

- 内部/ソフトウェア LSM 0 ドライブ 0 は、外部/物理ドライブ 64 に一致します。
- 内部 LSM 1 ドライブ 15 は、外部/物理ドライブ 33 に一致します。
- 内部 LSM 2 ドライブ 3 は、外部物理ドライブ 20 に一致します。

パーティションからセルを取り外す前のカートリッジの移動

SL8500 は、拡張パーティション分割を使用して、ドライブおよびセルアレイレベルまでパーティション分割できます。詳細は、[パーティションからセルを取り外す前のカートリッジの移動](#)を参照してください。

SL8500 CAP

ACSLs 8.4 以降、2 つの SL8500 CAP タイプが ACSLS でサポートされます。レガシー回転式タイプでは、1 つまたは 2 つの 39 セル CAP を各 SL8500 ライブラリに取り付けることができます。最新の CAP タイプである一括 CAP では、8 つの 36 セル CAP が各ライブラリに取り付けられています。

一括 CAP

大規模で頻繁なボールテイングアクティビティーが一般的に行われるデータセンターで挿入および取り出し操作をより迅速で効率的に行えるように、より新しい SL8500 ライブラリが設計されています。それぞれの SL8500 ライブラリには 8 つの一括 CAP があり、各レールの各サイドに 1 つの CAP があります。各 CAP には、リムーバブルな 12 スロットマガジンが 3 つ含まれています。

ACSLs は、一括 CAP を使用して、カートリッジを効率的に挿入し取り出します。CAP から挿入されたボリュームは、CAP と同じサイド、同じレール上にあるスロットに移動されます。そのサイドがいっぱいになった場合、もう一方のサイドにある空のスロットが選択されます。レールがいっぱいになった場合は、隣接するレールが選択されます。この方法により、ロボットの移動が最小限に抑えられ、ロボット間の競合が防止されます。同様に、*ejecting.sh* でボリュームのリストを取り出す場合、各ボリュームは、指定した CAP のなかでもっとも近い CAP に取り出されます。「[ejecting.sh](#)」を参照してください。

取り出し中にハンドルとして機能するように各マガジンの空きスロットを予約するには、*dv_config* を使用して、動的変数 *BULK_CAP_EJECT_HANDLE* を **TRUE** に設定します。

```
$ 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.
```

この変数を *TRUE* に設定すると、各マガジン内の 11 個のストレージセルが取り出し操作に使用されます。最上部の 3 つのレール上の各マガジンの最下部のセルは空のままであり、最下部のレール上の各マガジンの最上部セルは空です。これにより、ハンドルとして空のスロットを使用できます。この設定は、挿入操作中の動作には影響しません。

モジュールごとに 8 つの CAP があり、10 スtring SL8500 構成で 8 つの一括 CAP を設置できます。大型の SL8500 コンプレックスでは、複数の挿入および取り出しが進行中の場合、CAP 操作によってマウントおよびマウント解除操作が遅れることがあります。この問題を軽減するために、動的変数 *LIMIT_CAP_CONCURRENT_MOVES* を使用して、マウントおよびマウント解除を続行しながら、並行挿入および取り出しのロボット移動回数を制限できます。この機能を作動させるには、*dv_config* を使用して、動的変数 *LIMIT_CAP_CONCURRENT_MOVES* を **TRUE** に設定します。

```
$ 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

ACSLS で一括 CAP を処理するための SL8500 のアップグレード

一括 CAP を 1 台または複数の SL8500 に取り付けるときに ACSLS を更新するには:

1. ACSLS 8.4 をインストールします。

これは、一括 CAP を取り付けるはるか以前に行えます。

2. Oracle フィールドサービスエンジニア (FSE) は、一括 CAP をサポートする SL8500 ファームウェアを、影響のある SL8500 にロードしてアクティブ化する必要があります。

最低限の SL8500 ファームウェアレベルは 8.50 です。

3. 一括 CAP ハードウェアを取り付ける前に、ACSLS *cmd_proc* を使用して、一括 CAP を取り付けるライブラリをオフラインに変更します。
 - スタンドアロンの SL8500 に一括 CAP を取り付ける場合、またはストリング内のすべての SL8500 内に一括 CAP を取り付ける場合は、ACS (ライブラリコンプレックス) 全体をオフラインに変更します。
 - ライブラリコンプレックス内の一部の SL8500 だけに一括 CAP を取り付ける場合は、オフラインに変更する必要があるのは関与する LSM だけです。
4. この手順中に、FSE は影響のあるライブラリに一括 CAP ハードウェアを取り付けます。
 - a. FSE が一括 CAP ハードウェアを取り付ける前に、保守用ドアにもっとも近い 3 つのセルアレイ列内のカートリッジを取り外し、ライブラリ外で保管しておく必要があります。(取り付けが完了したあとで、カートリッジをふたたび挿入します。)

理由: 一括 CAP を取り付けるために、3 パックアレイに加え、2 列のシステムセルアレイを取り外す必要があります。3 番目の列のほとんどのストレージセルが、ACSLS にアクセスできなくなるシステムセルになります。

- b. FSE は、ライブラリに一括 CAP ハードウェアを取り付けます。
5. すべての SL8500 がリブートされ、ライブラリのハードウェア監査が完了したあとで、ACSLS *cmd_proc* を使用して、SL8500 ACS を診断モードに変更します。

診断モードでは、ACSLS 構成が更新され、ライブラリが監査されている間、ACSLS クライアントはこれらのライブラリにアクセスできません。

6. ACSLS が実行しているときに、*config acs acs_id* ユーティリティを使用して、データベースに記録された ACSLS 構成に一括 CAP を追加します。

注:

また、ACSLS を無効にし、*acsss_config* (オプション 8) を実行して構成を更新することもできます。これを行う場合、ACSLS をシャットダウンする前に *cmd_proc* から *query lmu all* を実行し、出力を保存します。続いて、同じ ACS 番号と同じポート接続を使用して、ACS を *acsss_config* に指定します。*acsss_config* が行われたあとで、ACSLS を有効にします。

7. CAP ステータスを表示および確認し、次を使用して ACSLS *cmd_proc* から入力します。

```
display cap * -f state mode status size type
```

出力例:

```
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. *cmd_proc* を使用して、一括 CAP が取り付けられたライブラリを監査します。次のいずれかを実行できます。

- ACSLS 全体を監査します。

```
audit <cap_id> acs <acs_id>
```

- 一括 CAP が取り付けられた LSM だけを監査します。

```
audit <cap_id> lsm <lsm_id> <lsm_id> <lsm_id> <lsm_id> ...
```

9. *vary* コマンドを使用して、ACS および LSM を ACSLS に対してオンラインに変更します。

これで ACSLS クライアントは一括 CAP を使用できるようになります。

10. CAP を使用して、前述の手順 4 でライブラリから取得されたすべてのカートリッジをふたたび挿入します。

挿入と取り出しの目的を示すカスタム SL コンソールメッセージ

SL コンソールは、一括 CAP に挿入しそこから取り出す目的を示したカスタムオペレータメッセージを CAP ステータス画面に表示できます。これらのメッセージでは、カートリッジが挿入されているパーティションや、カートリッジが取り出されるパーティションも報告できます。

これらのオペレータメッセージはオプションです。これらはベースとなる入力および取り出し処理には影響せず、SL8500 一括 CAP でのみサポートされています。

カスタムオペレータメッセージを使用するには:

1. SL コンソールを使用して *opmsg* 番号に対して表示するメッセージを定義します。次のオプションを選択します。

Tools

Configuration

CAP Usage Message

4 から 99 のメッセージ番号および関連付けられたメッセージを定義します。可能な場合は、メッセージを 20 文字以内に抑え、使用可能な領域に納まるようにします。

2. 手動の挿入または取り出しに対するオプションのオペレータメッセージ番号を入力します。

```
enter <cap_id> [opmsg <opmsg_nbr>]
```

```
eject <cap_id> [opmsg <opmsg_nbr>] vol_id | vol_range ...
```

カスタムの *opmsg* 番号は、ACSAPI クライアント、ACSL GUI、または *lib_cmd eject* からの取り出しには指定できません。これらの場合は、デフォルトのメッセージが表示されます。

カートリッジを挿入するため、または取り出されているカートリッジを取り外すために CAP がロック解除されたあとで、*opmsg* メッセージがシステム詳細の CAP ステータスページに表示されます。

例: 一括 CAP 1,2,1 からカートリッジを挿入するときに、カスタムオペレータパネルメッセージ番号 55 を指定するには:

```
enter 1,2,1 opmsg 55
```

回転式 CAP

SL8500 の回転式 CAP は、LSM 1、2、3 に対応する 3 つのレール (2、3、4) にわたります。基本構成には、SL8500 モジュールごとに 1 つの CAP と、オプションとして取り付けられる 2 番目の CAP が含まれます。

各回転式 CAP には、それぞれ 13 セルを備えたマガジンが 3 つあります。マガジンはそれぞれ別のレール上に置かれ、そのレール上の HandBot だけがアクセスできます。挿入中、ACSLs は、各マガジンから隣接する LSM (レール) にカートリッジを移動させようと試みます。隣接するレールが満杯の場合にかぎり、ボリュームは別のレールに移されます。同様に、特定のレール上のボリュームは、そのレール上の隣接するマガジンに取り出されます。

最上部のレール (LSM 0) には隣接する CAP マガジンがないため、エレベータが自動的に移動し始め、その最上部のレールからボリュームが取り出されます。挿入時、最上部のレールには、それより下のレールが満杯になるまで、搭載されません。最上部のレール上のドライブにマウントされたボリュームは、最終的には、マウント解除されるときに最上部の LSM に移されます。それ以外の場合は、最上部のレール上の LSM にボリュームを置くために、明示的な移動操作が必要です。このような追加の移動は、`watch_vols` ユーティリティーを使用して挿入後に自動的に処理できます。「[watch_vols](#)」を参照してください。

単一の回転式 CAP で複数の LSM に対応しているため、回転式 CAP の状態は、どの LSM のオンラインまたはオフライン状態にも結び付けられていません。隣接する LSM のいずれかまたはすべてがオフラインになっていても、CAP はオンラインのままにできます。逆に CAP がオフラインの場合、どの LSM がオンラインに変更されても、自動的にオンラインにはなりません。

複数の LSM が CAP にアクセスしている場合でも、SL8500 の回転式 CAP は LSM 1 にあるかのようにアドレス指定されます (1,5,9,13 など)。各パーティションが異なるホストに割り当てられている、パーティション分割されたライブラリでは、ユーザーは CAP が共有リソースであることを認識する必要があります。回転式 CAP は、挿入または取り出し操作の直後に予約済みになります。共有された環境でのオペレータは、即座に CAP を満杯にするか空にして、CAP 操作が完了したらドアを閉じる必要があります。[549 ページの付録 I「ライブラリのパーティション分割」](#)を参照してください。

以前のリリースの SL8500 では、2 番目の CAP が実際には取り付けられていなかった場合、オプションの CAP を存在しているが動作していないと報告していました。回避方法として、ACSLs ユーザーは、存在しない CAP の目的の状態をオフラインに維持するように指示されていました。これは、ファームウェアレベルが 6.07 以上のライブラリから問題ではなくなりました。

挿入または取り出し操作

enter 中、ACSLs は常に、CAP マガジンに隣接する LSM (レール) にカートリッジを移動させようとしています。取り出しの場合、ACSLs は常に、カートリッジを含む LSM に隣接する CAP セルにカートリッジを取り出そうとしています。

これら 2 つの操作ができない場合は、ライブラリコントローラが、エレベータを通じた別の LSM へのカートリッジの移動を担います。これには、2 つの HandBot とエレベータ間の移動が必要です。

一部の ACSLS クライアントでの挿入、取り出し、および監査の操作

その他のライブラリとは異なり、SL8500 は、SL8500 ライブラリ内の LSM ID ごとに CAP を定義していません。SL 8500 上の CAP は、その CAP ID に LSM 1 を含みます。SL8500 には LSM ID が 0、2、または 3 の CAP は存在しません。パーティション分割を行うと、LSM 1 (SL8500 の CAP ID における LSM ID) がパーティションに割り当てられないので、この問題が複雑になります。(CAP はまだ、共有リソースとしてすべてのパーティションで使用できることに留意してください)。

ACSLs クライアントのなかには、*enter*、*eject*、または *audit* に対して CAP を選択する前に、ACSLs に照会して (*query*)、どの CAP が存在し使用できるかを識別しないものがあります。これらは、存在しない *cap_ids* やオンラインではない CAP を指定する場合があります。たとえば、ACSAPI クライアントのなかには、すべての LSM ID に対して CAP が存在していると想定するものがあります。これらは、管理しているカートリッジやドライブの場所と同じ LSM ID を持つ CAP を自動的に指定する可能性があります。存在しない CAP ID を指定した挿入、取り出し、または監査は失敗します。

ACSLs *cmd_proc* を使用して次の操作を行う必要があります。

- 存在しない CAP ID を指定するクライアントに対して、カートリッジを挿入 (*enter*) し取り出し (*eject*) ます。
- これらのクライアントによって使用されるすべての ACS とパーティションに対して監査を実行します。

挿入、取り出し、または監査のあと、クライアントアプリケーションのデータベースを ACSLS データベースと再同期させる必要があります。

エレベータおよび PTP アクティビティを最小限に抑える

次のような、エレベータおよび PTP アクティビティを最小限に抑えるために行えるいくつかの操作があります。

カートリッジのマウント	可能な場合は、テープをマウントするときに、同じ LSM 内にあるカートリッジとテープドライブを使用してください。LSM は、SL8500 ライブラリ内の単一のレーンを指します。各 SL8500 には 4 つの LSM が含まれます。
フロートの使用	<p>各 LSM 内の一部の空きセルを管理することによって、ACSL の「float」オプション (ACSL ではデフォルトで有効) を活用します。カートリッジフロートは、ACSL がマウント解除されたテープカートリッジを同じ LSM 内の空のスロットに配置するか、テープがもともとパススルー操作を使用して別の LSM から取得された場合は、テープドライブにより近い LSM 内の空のスロットに配置できるようにする機能です。</p> <p>カートリッジがマウント解除されると、ACSL は、カートリッジの古いホームセルが別の LSM 内にあるときはいつでも、新しいホームセルを割り当てることによって、LSM 間のエレベータ (パススルー) アクティビティを回避しようとします。ACSL は次の場所にカートリッジを配置しようとします。</p> <ul style="list-style-type: none"> マウント解除されたテープドライブと同じ LSM またはドライブにもっとも近い LSM (空のストレージセルを含む)
カートリッジの挿入	<p>挿入されているメディアについて互換性のあるテープドライブを備えた LSM にカートリッジを挿入 (enter) します。</p> <p>例: LSM 2 および 3 上に LTO ドライブだけが、LTO カートリッジをこれらの LSM に配置します。これらのカートリッジを挿入するときに、LSM 2 および 3 に隣接する CAP マガジンに配置する必要があります。ACSL は続いて、その CAP マガジンに隣接する LSM にカートリッジを配置するために最大限努力します。</p>
スクラッチカートリッジ	使用されるそれぞれの LSM で、十分な数量のスクラッチカートリッジが利用できることを確認します。SL8500 の場合、これは、ライブラリの各レーン (LSM) 上でスクラッチカートリッジを使用できることを意味します。
空きセル	各 LSM 内に十分な空きセルがあることを確認します。

ワークロードをサポートするためのテープドライブの構成

SL8500 でテープドライブを構成する方法により、テープワークロードをサポートしながら、エレベータと PTP アクティビティの両方を最小限に抑えることができます。SL8500 内でテープドライブが置かれている場所の判断に使用する方法は、次のとおりです。

- ワークロードに必要な最大数のドライブをサポートする十分なドライブを使用して、ワークロードごとにカートリッジをクラスター化します。それぞれのワークロードで使用されるカートリッジを別々のレーンに分離し、ワークロード専用のレーンに、ワークロードのピーク時の使

用量に応じた最大の並行マウント数に対応できる十分なドライブがあることを確認します。ワークロード用のテープカートリッジだけではなく、必要になるスクラッチカートリッジもレールに備わっていることを確認します。

- 各主要アプリケーションのワークロードに別々のレールを割り当てます。これは、Symantic NetBackup や Tivoli などの一部のアプリケーションが独自のメディアとドライブを使用する場合があるためです。
- 単一のレールでのドライブおよびメディアのクラスタ化は、時間あたりのマウント数のしきい値に到達した、すべてのドライブが使用されている、またはレール上に収めるアクティブなカートリッジが多すぎるといった事態になるまで機能します。ワークロードに必要なリソースがレールの容量を超えた場合、カートリッジとドライブを 2 つ以上のレールに分散させます。
- タイプごとにドライブをクラスタ化し、異なるメディアタイプを使用するドライブを別々のレール (LSM) に配置します。たとえば、T9840 ドライブを 1 つのレール上に、T10000 ドライブを別のレール上に配置します。
- ライブラリ構成のパフォーマンスの制限を超えないように、負荷のかかるテープアプリケーションを構成します。
- 冗長性が得られるように 8 つの HandBot (レール当たり 2 つの HandBot) で SL8500 を構成します。これにより、ワークロードをサポートするカートリッジとドライブに常にアクセスできます。

カートリッジの場所の管理

カートリッジがもともとライブラリにどのように挿入されているか、つまりライブラリでのステータスが、ACSLS のパフォーマンスに影響する場合があります。考慮事項は次のとおりです。

カートリッジの挿入 **推奨事項:** CAP からカートリッジを挿入します。

前面アクセスドアを開いたまま手動でカートリッジをライブラリに配置すると、ライブラリ操作は中断し、ACSLS は完全な監査を実行して、ライブラリの実際の内容と一致するようにライブラリデータベースを更新する必要があります。

パフォーマンスを最大限に高めるには: カートリッジアクセスポート (CAP) からカートリッジを挿入します。

挿入中、ライブラリはオンラインのままであり、マウントは続行でき、ライブラリ管理ソフトウェアは常に、CAP マガジンに隣接した LSM にカートリッジを移動しようと試み、この結果パズルアクティビティが最小限に抑えられます。

これが可能ではない場合、ライブラリコントローラはエレベータを通じてカートリッジを別の LSM に移動させます。これにより、2 つの HandBot とエレベータ間をさらに移動させることが必要になります。

カートリッジのクラスタ化	ワークロードの最大のアクティビティ (ピーク時の使用量) に対応できるだけのテープドライブを備えた個別のレールに、カートリッジをワークロードごとにクラスタ化します。
フロートの使用	<p>推奨事項:</p> <p><i>float</i> がオンの場合、ACSLs は、マウント解除時にできるかぎりドライブに近い LSM 内にある新しいホームセルをカートリッジ用を選択します。このオプションは自動的に、ワークロードのドライブごとにカートリッジをクラスタ化します。</p> <p>各 LSM に、その LSM 内で新しいホームセルを選択できるだけ十分な空きセルが含まれていることを確認します。</p>
スクラッチカートリッジの供給	各レールに、正しい数量とタイプのデータカートリッジがあり、さらにワークロードに対応するための十分なスクラッチカートリッジがあることを確認します。

見つからないカートリッジを見つける

カートリッジが所定の場所がない、または ASSLS で行方不明になっている場合:

1. SL コンソールを使用して、SL8500 の物理的監査を実行します。

SL8500 の物理的監査は、`mount` の処理とその他のライブラリ操作リクエストの間のバックグラウンドタスクとして実行されます。

注意:

カートリッジを直接ロードするなどの手動の操作のために、SL8500 のコンテンツが ACSLS と同期しなくなった場合は、操作を継続しようとすることはお勧めしません。

テープを手動で追加する必要がある場合は、SL8500 内の特定の LSM に追加することをお勧めします。特定の LSM にテープを追加し、影響を受ける LSM だけを監査することが、より迅速で信頼性の高い解決法になります。

監査が進行中の際、影響を受ける LSM を ACSLS に対して診断状態に変更 (`vary`) する必要があります。SL8500 ライブラリの監査を実行したあとで、LSM を ACSLS に対してオンラインに変更 (`vary`) します。

2. ACSLS `audit` を実行して、ライブラリカートリッジの実際のインベントリに一致するように、ACSLs データベースを更新します。

SL8500 のオフラインへの変更

SL8500 コンポーネントの電源を切断する前 (これらが動作していない場合)、または SL8500 アクセスドアを開く前に、ACSLs に対してこれらをオフラインに変更 (`vary`) します。これにより、これらが使用できないことが ACSLS に通知されます。それらが使用可能になると、オンラインに戻します (`vary`)。

SL コンソールではなく ACSLS を使用して SL8500 コンポーネントをオフラインに変更する

SL8500 コンポーネント (ACS、LSM、および CAP) を SL コンソールではなく ACSLS に対してオフラインに変更 (*vary*) します。

ACSLS を使用すると、*vary offline* の強制でないかぎり、コンポーネントをオフラインにする前に、未処理のリクエストを完了させることができます。ACSLS に対する未処理のリクエストは SL コンソールにはわかりません。

SL コンソールを使用してコンポーネントをオフラインに変更した場合、処理中のリクエストが失敗する原因になることがあります。

SL8500 コンポーネントを ACSLS に対してオフラインに変更する場合

このセクションでは、SL8500 コンポーネントを ACSLS に対してオフラインに変更する場合について説明します。

アクセスドアを開く前

SL8500 アクセスドアを開く前に、ACS または 4 つの LSM すべてをオフラインに変更します。

- スタンドアロンの SL8500 の場合、次のコマンドを使用して ACS をオフラインに変更 (*vary*) します。

```
vary acs acs_id offline
```

- PTP を介して接続された SL8500 の場合、(アクセスドアが開いている SL8500 内の) 4 つの LSM すべてを、次のコマンドを 4 回使用して (4 つの LSM ごとに 1 回)、オフラインに変更 (*vary*) します。

```
vary lsm lsm_id offline
```

注:

SL8500 内のすべての CAP が自動モードである場合、次の手順を行う必要があります。

1. アクセスドアを開く前にそれらを手動モードに設定します。
2. アクセスドアを閉じて SL8500 がオンラインに戻ったあとに、自動モードに戻します。

CAP が動作していない場合

CAP が動作していない場合、次のコマンドを使用してオフラインに変更 (*vary*) します。

vary cap cap_id offline

サービス安全ドアを閉じたとき

ハードウェアの交換でサービス安全ドアを使用する必要があるときは常に、サービス安全ドアを閉じておく時間をできるだけ最小限に抑えることをお勧めします。サービス安全ドアは、特定のリクエストを完了するためにアクセスが必要になる可能性があるその他のハードウェアコンポーネント (エレベータ、CAP、セル) をブロックします。

- SL8500 の左または右サイドのどちらかのサービス安全ドアを閉じる前に、SL コンソールを通じて、そのサイドにあるエレベータをオフラインに変更 (*vary*) します。

サービス安全ドアを開いたあとで、SL コンソールを通じて、そのサイドにあるエレベータをオンラインに戻します (*vary*)。

- 右サイドでサービス安全ドアを閉じた場合、このドアは CAP へのアクセスをブロックします。
- SL8500 の右サイドにあるサービス安全ドアを閉じる前に、ACSLs を通じて、CAP をオフラインに変更 (*vary*) します。
- サービス安全ドアを開いたあとで、ACSLs を通じて、CAP をオンラインに変更 (*vary*) します。

注:

SL8500 のサービス安全ドアを閉じて、ライブラリのほかの部分からサービスベイを分離すると、CSE は LSM または ACS をオフラインにせずに、そのサイドにあるアクセスドアを開くことができます。

サービス安全ドアを使用するときにこれらの ACSLS コマンドおよびユーティリティーを使用しない場合

サービス安全ドアを使用するときに、進行していたり開始していたりしてはならない複数の ACSLS のコマンドおよびユーティリティーがあります。これらのコマンドは次のとおりです。

どちらかのサイドでサービス安全ドアが閉じられているときには、次のユーティリティーは使用しないでください。

- *acsss_config*
- *config* (*config drives* は OK です)

右 (CAP) サイドでサービス安全ドアが閉じられているときには、次のコマンドは使用しないでください。

- *enter*

- *eject*
- *set cap mode auto <cap_id>*

右 (CAP) サイドでサービス安全ドアが閉じられているときには、次のコマンドは使用できませんが、特別な考慮事項が適用されます。

- *audit*

audit コマンドは使用できます。ただし、監査の結果、カートリッジを取り出す (*eject*) 必要が生じた場合 (監査により重複または読み取り不能なラベルを検出したため)、監査が完了し、ACSL S データベースが更新されても、カートリッジは取り出されません。

- *vary acs and vary lsm*

これらのコマンドは成功しますが、*cmd_proc* とイベントログに、CAP の障害と動作不可能な CAP を報告したメッセージが表示されます。

動的構成 (config) ユーティリティの使用

動的構成 (*config*) ユーティリティを使用すると、ACSL S がオンラインで実行しているときに、ACSL S ライブラリ (およびコンポーネント) に構成の変更を実装できます。これらの構成変更は、*acsss_config.log* ファイルに記録されます。

次の動的構成ユーティリティがサポートされています。

- *config acs*
- *config drives*
- *config lsm*
- *config ports*

config ユーティリティの使用には次の利点があります。

- ACSLS を実行し続けながら、影響を受けないライブラリコンポーネントに対して *mount* リクエストを実行できます。
- ほかの構成情報はすべて変更せずに、指定したライブラリコンポーネントを再構成できます。たとえば、次を指定する場合:
 - ACS の場合、ほかの ACS の構成は影響を受けません。
 - LSM の場合、ほかの LSM の構成は影響を受けません。
 - ドライブパネル、パネル上のドライブ、既存のすべてのドライブに対するマウントおよびマウント解除は影響を受けません。

SL8500 の拡張

ストレージ拡張モジュール (SEM) を SL8500 に追加して、その容量を増やします。SEM は、CAP を含む顧客インタフェースモジュール (CIM) と、CIM に現在接続している SEM またはロボット工学インタフェースモジュール (RIM) の間に挿入されます。

SL8500 を拡張した場合:

- ライブラリ構成が変更し、ACSL S の動作中に ACSLS Dynamic 構成ユーティリティーを使用するか、ACSL S の停止中に `acs_ss_config` を実行して、ACSL S を再構成する必要があります。
- SL8500 を拡張するにはカートリッジを取り外す必要があります。これらのカートリッジをライブラリに戻す場合は、これらが以前に置かれていたセルアレイを空のままにしておきます。
- 拡張が完了したあと、SL8500 を 2 回リブートする必要があります。最初は新しい構成を検出するために、2 回目はライブラリのすべてのコンポーネントを更新された構成で再起動するためです。
- ライブラリでは、物理的監査を行なってすべてのカートリッジの場所を更新する必要があります。

ライブラリは、アクセスドアが閉じられると、自動的に物理的監査を開始します。ライブラリの物理的監査は、すべてのロボットの移動が少なくとも 1 分間停止すると完了します。

- ライブラリの物理的監査が完了したあとで、「[拡張 SL8500 の監査](#)」の手順を使用して、ACSL S データベースを更新します。

注:

ACSL S 監査が完了し、ACSL S データベースが新しいカートリッジの場所で更新されるまで、自動ライブラリ操作を開始しないでください。

内容

SL8500 の物理的な拡張の内容は次のとおりです。

- 新しい SEM が既存の SEM または RIM と CIM の間に挿入されます。

これで新しい SEM と CIM でのパネル番号は、既存の SEM と RIM でのパネル番号よりも高くなります。

- CIM は取り出す必要があるため、より高い新しいパネル番号が CIM 上の 3 つのセルパネル (カラム) に割り当てられます。より高いパネル番号が CIM 上のセルパネルに割り当てられると、CIM 上のすべてのカートリッジのアドレスが変更します。
- SL8500 を拡張するには多くのカートリッジを取り外す必要があります。既存のレールのボルトを外して新しいレールをボルトで固定し、CIM を押し上げるためにセルアレイを取り外す必要があります。
- 新しいレールが取り付けられ SL8500 が拡張されたあとで、取り外されたカートリッジをライブラリに戻すことができます。ライブラリを拡張するために一時的に取り外されたセルアレイは空のままにしておきます。

拡張後、ACSL S がこれらのカートリッジの新しいアドレスでそのデータベースを更新できるように、ライブラリを監査 (*audit*) します。カートリッジを新しい場所に配置したときに、以前セル内にあったカートリッジを ACSLS が検出しようとする、ライブラリのパフォーマンスが監査中に大幅に低下することがあります。ACSL S とライブラリの両方のパフォーマンスを最適化するには、次の手順を完了します。

1. ACSLS 監査を開始する前に、ライブラリの物理的監査を完了させます。ライブラリがそのデータベースからカートリッジの場所を ACSLS に報告できる場合、結合された監査がより迅速に終了します。それ以外の場合、ライブラリは、ACSL S の *audit* リクエストに応答する前に、カートリッジの場所を再検証する必要があります。
2. ライブラリを拡張するために取り外されたカートリッジに対して、次のいずれかの方法を使用してライブラリに再挿入します。
 - a. ライブラリからカートリッジを取り外し、ACSL S からライブラリを監査 (*audit*) し、続いて取り外されたカートリッジをライブラリに戻します (*enter*)。
 - b. ライブラリに追加されたパネル番号だけにカートリッジを挿入し、最初にこれらのパネルを監査 (*audit*) します。

スクラッチカートリッジを追跡するために ACSLS スクラッチプールを使用していない場合は、不在ボリュームの保持を有効にしていれば (デフォルト)、特別な手順は必要ありません。ライブラリを監査 (*audit*) すると、移動したカートリッジが不在とマークされている場合がありますが、*audit* は、新しい場所でそれらを見つけ、それらを再アクティブ化します。カートリッジの場所が更新され、重要な情報は失われません。

カートリッジを ACSLS スクラッチプールに割り当てることによってスクラッチカートリッジを管理しており、カートリッジのスクラッチステータスをクリアしない場合、新しく追加したパネルにカートリッジを置き、最初にこれらのパネルを監査 (*audit*) します。

- ライブラリに追加されたパネルに、取り外されたカートリッジを挿入します。つまり、最初の新しい SEM にあるドライブにもっとも近い 3 つのパネル以外のパネル (列) にのみそれらを挿入してください。(最初の SEM の最初の 3 つのパネルには、以前に CIM 内の 3 つのパネルに割り当てられたパネル番号が付けられます。
- 続いて、もっとも高いパネル番号から下に追加された新しいパネル番号すべてにわたって、ACS または LSM のパネルを監査 (*audit*) します。

audit cap_id panel panel_id コマンドを使用します。

- 最後に、残りの ACS または LSM を監査 (*audit*) します。

audit cap_id acs acs_id または

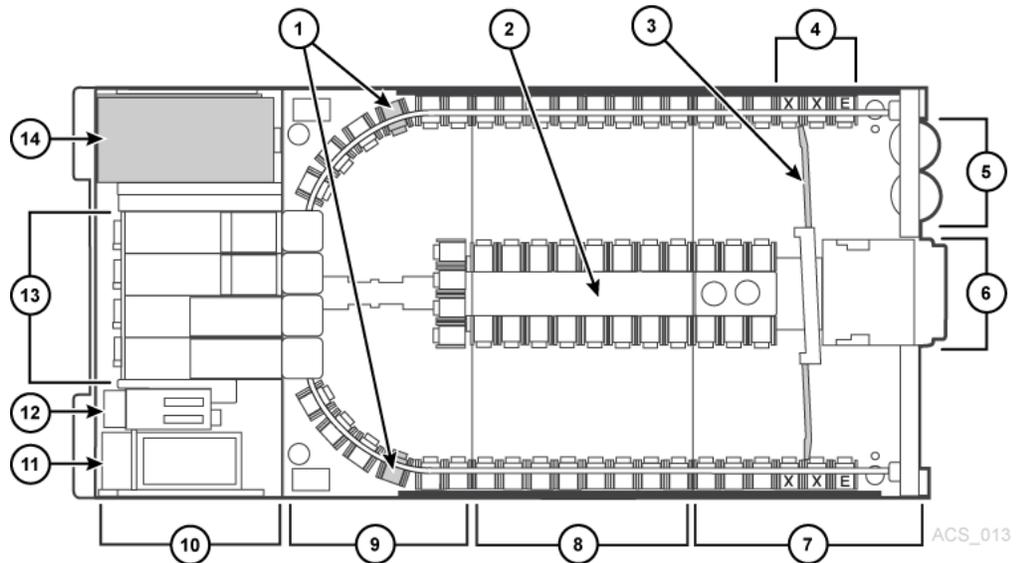
audit cap_id lsm lsm_id コマンドを使用します。

この手順は、「[拡張 SL8500 の監査](#)」で詳細に説明されています。

SL8500 モジュールの図:

次の図では、顧客拡張モジュール (#5) 内の 3 つのセルパネルは、CEM のドライブ側にある外部および内部の両方のウォールにある 3 列のカートリッジです。これらは、サービス安全ドアがアクティブ化されたときに、HandBot がアクセスできる CIM 内の唯一のセルアドレスです。

図C.3 SL8500 顧客拡張モジュール



凡例:

1. カートリッジアクセスポート (2) CAP
2. ファサードオペレータパネル (オプション) Icey パッド
3. 顧客インタフェースモジュール
4. ストレージ拡張モジュール
5. ロボット工学インタフェースモジュール
6. PTP (パススルーポート)
7. ドライブ電子モジュール
8. AC 電源装置電子制御モジュール
9. DC 電源装置
10. テープドライブ
11. 付属品ラック
12. 内部ウォール
13. 保守用ドア
14. 予約済み列
 - E = エンドストップ
 - S = システムセル

拡張 SL8500 の監査

SL8500 を拡張する前に:

1. カートリッジのスクラッチステータスを保持する場合:
 - a. 次のコマンドを実行して、ライブラリ内の LSM についてもっとも高いパネル番号を特定します。

```
display panel acs,lsm, *
```

ここでは:
 - *acs* は ACS です
 - *lsm* はいずれかの LSM です。SL8500 上のすべてのレール (LSM) には同じ数のパネルがあります。
 - *** は LSM 内のすべてのパネルを表示します。
 - b. もっとも高いパネル番号を記録します。
2. 構成が更新され、カートリッジアドレスが *audit* で更新されるまで、LSM を診断状態に変更 (*vary*) します。

3. ACSLS データベースをバックアップします。
4. SL8500 を拡張します。

拡張中、新しいまたは拡張したレールを取り付けるため、およびストレージ拡張モジュールを追加するために、カートリッジを取り外す必要があります。

注:

ライブラリを拡張するために一時的に取り外されていたセルアレイに、これらのカートリッジを戻さないでください。これにより、これらの場所が最初に監査されるときライブラリのパフォーマンスにおける大幅な低下が回避されます。

5. ライブラリに追加されたパネルに、取り外されたカートリッジを挿入します。

つまり、最初の新しいストレージ拡張モジュール内のドライブにもっとも近い 3 つのパネル以外のパネル (列) にのみ、これらのカートリッジを挿入する必要があります。(最初のストレージ拡張モジュール内の最初の 3 つのパネルは、以前に CIM 内のパネルに割り当てられた番号を保持しています。)

SL8500 の物理的な拡張が完了したあとで:

6. ライブラリを 2 回リブートします。
7. LSM を診断状態のままにしておきます。
8. 次の方法のいずれかを使用して、ACSLS データベースの構成を更新します。
 - ACSLS が実行している場合:

```
config acs acs_id
```

または、SL8500 内の LSM (レール) ごとに:

```
config lsm lsm_id
```

- ACSLS をシャットダウンする場合:
 - ACSLS を停止します: `acs disable`
 - ACSLS 構成を更新します: `acs config`
 - ACSLS を起動します: `acs enable`

カートリッジのスクラッチステータスを保持する場合は、手順 9 から 11 を使用して、ライブラリを監査 (*audit*) します。ACSLS スクラッチプールを使用してカートリッジのスクラッチステータスを追跡していない場合は、手順 12 にスキップします。

9. ライブラリ内の LSM の新しくもっとも高いパネル番号を特定し、次のコマンドを使用してこの番号を記録します。

```
display panel acs,lsm, *
```

ここでは:

- *acs* は ACS です。
- *lsm* はいずれかの LSM です。SL8500 上のすべてのレーン (LSM) には同じ数のパネルがあります
- * は LSM 内のすべてのパネルを表示します。

10. 新しくもっとも高いパネル番号から古くもっとも高いパネル番号 + 1 まで、拡張中に追加されたパネルを監査 (*audit*) します。次のコマンドを使用して、降順でこれらのパネルを監査 (*audit*) します。

```
audit cap_id panel panel_id
```

ここでは:

- *cap_id* は、重複した *vol_ids* または読み取り不能なラベルを持つカートリッジを取り出す必要のある CAP です。
- *panel_id* は監査されているパネルです (*acs,lsm,panel*)

11. 各 LSM 内の追加されたパネルごとに別々の監査を実行します。

注:

ライブラリコンプレックス内に存在する CAP と同じ数の並行監査だけを実行できます。それぞれの並行監査には別々の CAP が必要です。監査によってすべてのカートリッジが CAP に取り出された場合、監査が終了する前に取り外す必要があります。

12. 拡張した SL8500 内の各 LSM 全体を監査 (*audit*) します (または ACS 全体を監査します)。
13. ACSLS データベースをバックアップします。
14. LSM をオンラインに変更 (*vary*) します。
15. 自動ライブラリ操作を再開します。

パススルーポートを使用した SL8500 の接続

SL8500 ライブラリを既存の SL8500 ライブラリコンプレックスに追加した場合、カートリッジアドレスを更新するために SL8500 を監査 (*audit*) する必要があることがあります。

SL8500 コンプレックス内の LSM は、次の図に示すように CAP 側から見たときに上から下、右から左に番号が付けられます。

図C.4 4つの接続された SL8500 ライブラリ



サイトの設定方法に応じて、新しい SL8500 を既存の SL8500 の右側に追加していたり、左側に追加していたりします。その結果:

- 新しい SL8500 を左側に追加した場合、新しい LSM にはより高い番号が付けられ、既存の LSM の番号は変更されません。中断は最小限に抑えられます。
- 新しい SL8500 を右側に追加した場合、既存のすべての LSM の番号が変更されます。これにより、既存のすべてのカートリッジのホームセルのアドレスが変更します。

SL8500 PTP 接続を設置する前に

1. SL8500 接続を設置する前に ACSLS にすべての関連するメンテナンスを適用します。
2. 不在カートリッジのサポートを有効にします。

不在カートリッジのサポートが有効になっている場合、ACSLS は、削除される LSM 内のカートリッジや、*audit* が古いアドレスで見つけられないカートリッジを削除しません。これらのカートリッジは、不在としてマークされ、スクラッチステータスを除いたそれらに関するすべての情報を保持します。これらのカートリッジが *audit* によってあとから検出されると、それらは再アクティブ化されます。

- *ABSENT_VOLUME_RETENTION_PERIOD* 変数が 0 ではない場合、不在カートリッジのサポートが有効になっています。デフォルト値は 5 日です。
 - *acsss_config* (オプション 3) を使用します。手順については、「[CSI チューニング変数の設定](#)」と変数 *ABSENT_VOLUME_RETENTION_PERIOD* を参照してください。
3. 次のいずれかの方法によって、構成を更新します。
 - ACSLS を起動して、次のコマンドを使用します。

```
config acs acs_id
```

- ACSLS をシャットダウンして、次のコマンドを使用します。
 - ACSLS を停止します: *acsss disable*
 - ACSLS 構成を更新します: *acsss_config*
 - ACSLS を起動します: *acsss enable*

`acsss_config` を実行するときに、ACSL S は実行できません。

詳細は、「CSI チューニング変数の設定」または「`acsss` マクロ」を参照してください。

新しい SL8500 の追加

既存の SL8500 ライブラリコンプレックスにさらに SL8500 を追加した場合、新しい ACSLS 構成を更新する必要があります。新しい SL8500 を追加したために既存の SL8500 の LSM の番号が変更される場合、それらの LSM のカートリッジアドレスを更新する必要があります。

カートリッジアドレスは、そのカートリッジに関連するその他の情報を失わずに更新する必要があります。これには、スクラッチステータス、プール、ロック、所有権などのカートリッジ情報、クリーニングカートリッジの使用回数、およびカートリッジが挿入された日付と時間が含まれます。

接続された SL8500 内の LSM は、(CAP 側から見て) 上から下、右から左に番号が付けられます。

図C.5 4つの接続された SL8500 ライブラリ



左側への新しい SL8500 の追加

元の SL8500 の (CAP 側から見て) 左側に、新しい SL8500 を中断せずに追加した場合、ホストソフトウェア (ACSL S) を使用して、追加ドライブおよびライブラリを動的に構成します。動的構成がインストールされている場合、ACSL S をリサイクルする必要はありません。`mount` リクエストは、このフェーズ中、既存の SL8500 に対して通常どおり続行します。

新しい SL8500 内のセルにカートリッジを配置した場合、これらのカートリッジを ACSLS データベースに追加するために、新しい SL8500 の LSM 上で ACSLS `audit` を実行する必要があります。

既存の SL8500 内の LSM は、監査中、オンラインにできます。

新しい ACSLS 構成の動的な構成

1. 新しい SL8500 をライブラリコンプレックスに追加します。
2. 構成変更を行う前に、ACSLS をバックアップします。
3. ACSLS 構成を動的に更新し、次のコマンドを使用します。

```
config acs acs_id
```

ACSLS が停止したときに、次のコマンドを使用して ACSLS 構成を更新することもできます。

```
acsss_config
```

4. 構成変更を行なったあとに、ACSLS をバックアップします。

注:

新しい SL8500 内にカートリッジがある場合は、新しい SL8500 内の LSM (レール) を監査 (*audit*) して、これらのカートリッジを ACSLS データベースに追加します。

右側への新しい SL8500 の追加

右側に新しい SL8500 を追加する場合は、次の図に示すように、既存の LSM の番号を変更します。

右側に SL8500 を追加するときの考慮事項

新しい SL8500 が右側に追加されると、すべての既存の LSM の番号が変更され、カートリッジアドレスが変更されます。LSM 番号を変更すると、既存のすべてのカートリッジアドレスが変更されます。アドレスが変更したカートリッジを ACSLS がマウント (*mount*) しようとした場合、ACSLS がカートリッジを見つけられないので、*mount* は失敗します。

- すべてのカートリッジアドレスが更新されるまで、マウントアクティビティを休止します。
- ACS 内のすべての LSM を診断状態に変更 (*vary*) してマウントを防止します。
- 特定の順序で、既存および新規の SL8500 内の LSM を監査することによって、カートリッジアドレスを更新します。

新しい ACSLS 構成の動的な構成

新しい SL8500 の追加後に ACSLS 構成を動的に更新するには:

1. 既存の LSM を診断状態に変更 (*vary*) します (*vary lsm_id diag*)。

注意:

これらの LSM は、監査が終了するまで診断状態のままにしておく必要があります。それ以外の場合は、次の問題が発生します。

- 最後に認識されたアドレスでカートリッジが見つからないため、マウントが失敗します。
 - 番号が変更された LSM 内の空のセルの ACSLS マップを監査が更新するまで:
 - 新しいカートリッジを挿入すると既存のカートリッジと衝突します。
 - 既存の (番号が変更された) LSM にカートリッジを移動すると、すでにセル内にあるカートリッジと衝突します。
2. 新しい SL8500 をライブラリコンプレックスに追加します。
 3. ACSLS を (構成変更を行う前に) バックアップします。
 4. 次のコマンドを使用して、ACSLS 構成を動的に更新します。

`config acs acs_id`

注:

または、ACSLS の停止中に ACSLS 構成を更新します。最初に、`acsss disable` コマンドを使用して ACSLS を停止させます。`acsss_config` で構成を変更し、最後に、`acsss enable` で ACSLS を動作状態に戻します。

注:

オンライン状態のときに新しい LSM が追加されます。監査が終了するまで、これらの LSM を診断状態に変更 (*vary*) します。

5. ACSLS をバックアップします (構成の変更後)。
6. ライブラリを監査 (*audit*) してカートリッジアドレスを更新します (番号の変更以降)。

次の手順を使用します。

- a. 番号が変更された既存の SL8500 内の各 LSM を監査 (*audit*) します。
 - カートリッジを損失しないように慎重な手順で既存の SL8500 を監査 (*audit*) します。

もっとも高い番号の LSM からもっとも低い番号の LSM ID へという順序で各 LSM (レール) を監査 (*audit*) します。監査は、(番号が変更された) LSM アドレス内のすべてのカートリッジを検出します。

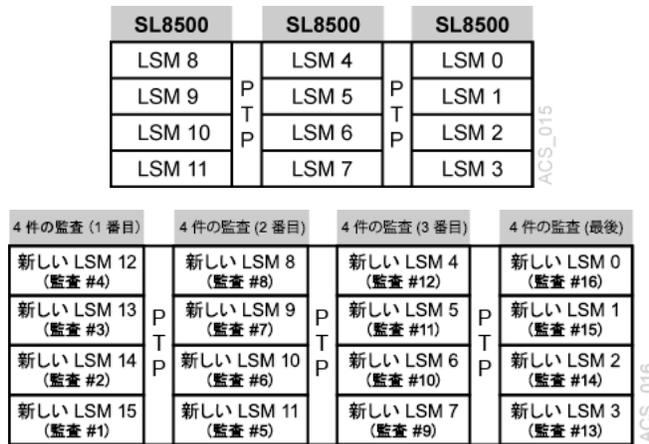
- ACSLS の場合、各 LSM を個別に監査 (*audit*) し、ある LSM の監査を次の LSM の監査前に終えられるようにする必要があります。

audit は、カートリッジの以前のホームセルアドレス (右側の LSM 内) を調べたあと、カートリッジのアドレスを更新します。左端の SL8500 内の LSM の監査には時間がかかります。ほかの SL8500 の監査はすべてこれよりも速くなります。

左端の LSM の監査に時間がかかる理由は、それらが *Cartridge Recovery* リクエストのカスケードをトリガーするためです。監査でホームセルアドレスが異なっているカートリッジを検出すると、*audit* は、データベースに記録されたセルアドレスをチェックします。そのセルに別のカートリッジがある場合、*Cartridge Recovery* はそのカートリッジなどを調べます。すべての LSM アドレスが変更されているため、これらの回復は SL8500 ライブラリコンプレックス全体をカスケードします。

- 監査が終了するまで、LSM を診断状態に維持します。監査が終了したら、これらをオンラインに変更 (*vary*) できます。*audit* で、この LSM 内のカートリッジのアドレスが更新されるため、監査された LSM のカートリッジで、自動 *mount* アクティビティを再開できます。
- b. 最後に、新しく追加した SL8500 内の LSM (LSM ID が最低) を監査 (*audit*) します。

これらの LSM が監査されたあとで、それらをオンラインに変更でき、次の図に示すように、それらのカートリッジをテープドライブにマウントできます。



7. 監査の完了後に ACSLS をバックアップします。

ACS のマージの手順

SL8500 PTP では、単一の ACS に複数の個別の SL8500 をマージできます。カートリッジおよび動作停止時間に関する情報の損失を最小限に抑えるために、これらの推奨される手順に従います。

注:

ACSL S 構成が更新されるときに、グローバルカートリッジアドレスの変更は行われません。

ACSL S に関する 2 つのシナリオを次に示します。最初のシナリオでは、マージする ACS に、(CAP 側から見て) 右から左に番号が付けられます。2 番目のシナリオでは、左から右に番号が付けられます。結果得られる ACS がより低いもっとも低い ACS ID を使用することが想定されます。

右から左に番号が付けられた ACS のマージ

次のシナリオでは、マージする ACS に、(CAP 側から見て) 右から左に番号が付けられます。

図C.6 マージ対象の ACS

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

図C.7 目的の構成: 単一の ACS

ACS 0		
LSM 0,8	P	LSM 0,4
LSM 0,9	T	LSM 0,5
LSM 0,10	P	LSM 0,6
LSM 0,11	P	LSM 0,7
		LSM 0,0
		LSM 0,1
		LSM 0,2
		LSM 0,3

ACS_018

右から左に番号が付けられた ACS をマージする手順

1. マージされる右端の ACS を除いたすべての ACS をオフラインに変更 (*vary*) します。

これにより、カートリッジアドレスが更新されている間のマウントおよびマウント解除が防止されます。

2. ACSLS を停止 (*stop*) します: `acsss disable`
3. 構成変更を行う前に ACSLS をバックアップします。
4. `acsss_config` を使用して、(ACSL S が停止している間に) ACSLS 構成を更新します

5. ACSLS を起動します: `acsss enable`
6. (ACS に追加された) 新しい LSM を診断状態に変更 (`vary`) します。

オンライン状態にある間に、これらの LSM が追加されます。

7. 構成の変更後、ACSLS をバックアップします。
8. ACS に追加された LSM を監査 (`audit`) します。

これらを監査する順序は重要ではありません。ACS 全体、つまりすべての LSM を一度に監査できます。

カートリッジが再アクティブ化されます。

9. 監査の完了後に ACSLS をバックアップします。
10. 新しい LSM をオンラインに変更 (`vary`) し、通常の自動処理を再開します。

左から右に番号が付けられた ACS のマージ

次のシナリオでは、マージする ACS に左から右に番号が付けられます。

図C.8 既存の構成: 3つの個別の ACS

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

図C.9 目的の構成: 単一の 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

左から右に番号が付けられた ACS をマージする手順

1. すべての ACS をオフラインに変更 (`vary`) します。
 これにより、カートリッジアドレスが更新されている間のマウントおよびマウント解除が防止されます。
2. ACSLS を停止 (`stop`) します: `acsss disable`。
3. 構成変更を行う前に ACSLS をバックアップします。
4. `acsss_config` を使用して、(ACSLS が停止している間に) ACSLS 構成を更新します。

5. ACSLS を起動します: `acsss enable`。
6. (ACS に追加された) 新しい LSM を診断状態に変更 (*vary*) します。

(オンライン状態にある間に、これらの LSM が追加されます)。

7. 構成の変更後、ACSLS をバックアップします。
8. 左側に追加された新しい LSM を診断状態に変更 (*vary*) します。

これらの LSM の監査が完了するまで、マウントおよびマウント解除を実行する必要はありません。

9. 新しく構成された LSM を監査 (*audit*) します。

これらを監査する順序は重要です。次の順序で監査 (*audit*) します。

- a. まず、同じ ACS 番号を保持する SL8500 内の LSM を最初に監査します。
 - これらのカートリッジの古いアドレスに現在割り当てられている LSM を監査 (*audit*) する前に、*audit* で新しい LSM アドレスにあるこれらのカートリッジを検出させます。
 - これらの SL8500 内のすべてのカートリッジに対して、LSM アドレスが更新されません。
 - ACSLS の場合、各 LSM を個別に監査 (*audit*) し、ある LSM の監査を次の LSM の監査前に終わらせるようにします。
 - 監査が終了するまで、LSM を診断状態に維持します。

監査が終了したら、これらをオンラインに変更 (*vary*) できます。*audit* で、カートリッジのアドレスが更新されるため、監査された LSM のカートリッジで自動 *mount* アクティビティを再開できます。

- b. 最後に、もっとも低い ACS にマージされた SL8500 を監査 (*audit*) します。これらの SL8500 内のカートリッジが再アクティブ化されます。
 - これらの LSM すべてを同時に監査できます。
 - これらの LSM を監査する順序は重要ではありません。

10. (監査の完了後に) ACSLS をバックアップします。
11. すべての LSM をオンラインに変更 (*vary*) し、通常の自動処理を再開します。

PTP の取り外しと ACS の分割

2 台の SL8500 に接続する PTP メカニズムを取り外して、単一の ACS を 2 つの別個の ACS に分割できます。この構成の変更は、2 つの ACS を単一の ACS にマージする場合の逆です。

SL8500 を含む新しい ACS を分割の左側に追加することは非常に簡単です。これにより、より高い番号の LSM が新しい ACS に割り当てられ、既存の ACS 内に残っている LSM の番号は変更されません。

左側に SL8500 から新しい ACS を作成して ACS を分割する - 考えられるシナリオ

図C.10 既存の構成: 1 つの ACS

ACS 0						
LSM 0,12	P T P	LSM 0,8	P T P	LSM 0,4	P T P	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

図C.11 目的の構成: 2 つの ACS

ACS 1			ACS 0		
LSM 1,4	P T P	LSM 1,0	LSM 0,4	P T 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

ACS を分割するための ACSLS の手順

動的構成を使用すると、ACSLs は実行中に ACS を追加できます。ただし、動的構成はグローバルカートリッジアドレスの変更を行いません。

1. 構成の変更前に、ACSLs をバックアップします。
2. 新しい ACS に移動する LSM をオフラインに変更 (*vary*) します。

これにより、カートリッジアドレスが更新されている間のマウントおよびマウント解除が防止されます。

既存の ACS 内に残されている LSM はオンラインのままにしておくことができます。

これらの LSM 内のマウントおよびマウント解除を続行できます。

3. 分離する SL8500 に接続している 4 つの PTP メカニズムを取り外します。
4. パススルーポートが取り外されたあとで、既存の ACS を再構成します。

これにより、新しい ACS に移動している LSM が取り外されます。次のコマンドを使用します。

```
config acs acs_id
```

5. 次のコマンドを使用して、新しい ACS を追加します。

```
config acs acs_id new
```

または、ACSL S の停止中に ACSLS 構成を更新できます。

```
acsss_config
```

新しい LSM はオンライン状態で追加されます。

6. 監査が終わるまで、これらの LSM を診断状態に変更 (*vary*) します。
7. 新しい ACS を診断状態に変更 (*vary*) します。
8. 構成の変更後、ACSL S をバックアップします。
9. 新しい ACS 内の LSM を監査 (*audit*) します。

これらの LSM の監査が完了するまで、マウントおよびマウント解除を実行する必要はありません。これらを監査する順序は重要ではありません。これらすべてを一度に監査できます。

カートリッジが再アクティブ化されます。

10. 監査の完了後に ACSLS をバックアップします。
11. 新しい LSM をオンラインに変更 (*vary*) し、通常の自動処理を再開します。

LSM の *audit* が完了するとすぐに、その LSM をオンラインに変更できます。

分割の右側への新しい ACS の追加

分割の右側に新しい ACS を追加すると、すべてのカートリッジアドレスが再マップされます。

これは推奨されていないので、詳細な手順は記しません。

ただし、いくつかの考慮事項があります。

- 右側の SL8500 の LSM が既存の ACS から取り外されると、これらの LSM 内のカートリッジは、不在とマークされます (不在カートリッジの保持がアクティブな場合)。これらは、新しい ACS が監査されるときに再アクティブ化されます。
- 既存の ACS 内の LSM を一度に 1 つの LSM ずつ監査して、これらの LSM 内のカートリッジのアドレスを更新する必要があります。最初にもっとも高い番号が付けられた LSM を監査 (*audit*) し、続けて次に低い LSM を監査します。次の監査の開始前に、1 つの LSM の *audit* を終える必要があります。

付録D SL3000 の ACSLS サポート

SL3000 には次の利点があります。

- 200 から 4500 のストレージセルに対応するストレージ容量。
- 1 - 56 台のテープドライブに対応するパフォーマンス。
- 混在したメディアとさまざまなテープドライブのサポート。
- ストレージセルおよびドライブのレベルで SL3000 をパーティション分割する機能。
- アクセス拡張モジュール (AEM) を取り付けるオプション (ACSLS 7.3 PUT0801 以降)。このオプションは、「[AEM の使用](#)」で説明されています。

注意:

SL3000 は、ACSLS に対して構成または再構成されたあとで、監査 (*audit*) する必要があります。ライブラリのアクティブ化された容量を増加または変更したあとにも、SL3000 を監査 (*audit*) する必要があります。

ライブラリは、ACSLS が監査するときに、ACSLS にアクセスできない可能性のあるセルの場所を報告します。アクセスできないセルの場所には、CAP、ドライブ、およびオペレータパネルが取り付けられている場所、ロボットがアクセスできないセルの場所、アクティブ化されていないセルの場所、およびこのパーティション内にはない場所が含まれます。

ACSLS サポート

ACSLS は、次の SL3000 機能のサポートを提供します。

- 新しい LSM およびパネルタイプと、SL3000 ライブラリ用のパネルマップ
- 最大 8 個のパーティション。

SL3000 はドライブおよびセルレベルまでパーティション分割できます。あるパーティションから別のパーティションにセルが割り当て直されている場合は、それらのセル内のカートリッジは孤立し、以前に含まれていたパーティションではアクセスできなくなります。これを回避するには、SL3000 を再パーティション分割する前に、パーティションに残るセルにカートリッジを移動します。

- 新しい最大数の SL3000 用の CAP (12)、パネルあたりのドライブ (32)、セル行 (52)。
- CAP を SL3000 内のパーティション専用にする機能。

CAP は、SL コンソールを使用してパーティション専用になります。ライブラリが ACSLS に専用の CAP を報告します。

ACSLS によって管理されたパーティション専用の CAP は、自動モードに設定できます。

- SL3000 CAP 別名処理

SL3000 には最大 12 の CAP を備えられますが、バックアップアプリケーションの中には、現在、LSM あたり最大 3 つの CAP しかサポートしないものがあります。CAP 別名処理を使用すると、CAP 3 - 10 に、CAP 0、1、または 2 の別名を付けることができます。

PUT0801 では、CAP 0 の別名を付けられなくなりました。CAP 0 はアクセス拡張モジュール (AEM) 用に予約されていて、特殊な特性があります。このため、別名に使用できる CAP 番号は 1 および 2 だけです。

- PUT0801 対応の ACSLS 7.3 では、一括挿入および取り出しに AEM をサポートしていません。

SL3000 向けに ASSLS を構成する前に

SL3000 向けに ACSLS を構成する前に:

- SL3000 を ACSLS に接続します
- SL3000 のすべてのコンポーネントが動作していることを確認します。

ACSLS は、ライブラリによって報告される情報から、そのライブラリ構成を構築します。SL3000 コンポーネントが動作していない場合は、ライブラリ情報が ACSLS に報告されない可能性があり、SL3000 の ACSLS 構成が不完全である場合があります。

ACSLS への SL3000 の接続

SL3000 ライブラリは、Ethernet 物理インタフェースを介した TCP/IP プロトコルを使用して、ホストおよび ACSLS を管理し、これらと通信します。このインタフェースにより、ACSLS は SL3000 に接続し、通信できます。

SL3000 への TCP/IP 接続を使用すると、次の利点が得られます。

- ACSLS は、複数の並行リクエストをライブラリに送信でき、ライブラリはそれらを並列で処理できます。これにより、ライブラリのパフォーマンスが向上します。
- TCP/IP を使用して接続した場合、SL3000 はデュアル TCP/IP と冗長電子装置をサポートします。

- TCP/IP ホスト/ライブラリインタフェースは、SCSI Media Changer over Fibre Channel よりも機能豊富なインタフェースです。ライブラリのステータスやエラーなどに関する詳細を ACSLS で使用できます。

別のサブネットまたは制御されたネットワーク上に SL3000 ライブラリを配置して、ARP ブロードキャストのフラッドからこれを保護することをお勧めします。

すべての SL3000 コンポーネントが動作していることの確認

SL3000 のすべてのコンポーネントが動作していることを確認するために、StorageTek ライブラリコンソール (SL コンソール) にログオンします。

手順については、『SL3000 ユーザーズガイド』を参照してください。

ACSLS に対する SL3000 の構成

SL3000 コンポーネントが動作するようになったあとで、ACSLS に対して SL3000 を構成します。acsss_config (「CSI チューニング変数の設定」を参照) と動的構成 (「動的構成 (config) ユーティリティの使用」を参照) のどちらでも使用できます。

SL3000 の監査

ACSLS に対して構成または再構成したあとで、SL3000 を監査します。監査は、どのセルがこのライブラリで管理しているパーティションの一部であるかを ACSLS に通知し、パネルドライブまたは CAP 上に実際に存在しているセルを表示します。

- まず、ACSLS に対して SL3000 を構成または再構成します。
- ACS を監査 (*audit*) します。
- 次のいずれかのコマンドを入力して、割り当てられたセルと空きセルを調べます。
 - `free_cells.sh -a`
 - `get_license_info`
 - `display cell * -f status`
- 次のいずれかを使用してドライブを調べます。
 - `query drive all`
 - `display drive * -f type serial_num wwn`

SL3000 アドレス指定

ドライブとストレージセルのどの ACSLS アドレスも変更せずに、SL3000 を拡張できます。ACSLS アドレスは、可能な最大の SL3000 ライブラリをマップするため、既存のセル、ドライブ、CAP のアドレスを変更せずにモジュールを追加できます。

モジュールが取り付けられていない場合、その上のパネルは、単に「not installed」としてマップされます。パネルが取り付けられている場合、これらのパネルでのアドレスが使用できるようになります。このように、新しいモジュールを取り付けるときに、既存のモジュール上のパネルのアドレスは変更しません。

図D.1 SL3000 アドレス指定



最小の SL3000 構成は、単一の基本ドライブモジュール (BDM) です。単一の BDM 構成では、パネル 12 および 13 が取り付けられており、ほかのすべての潜在的なパネル場所は取り付けられていません。ドライブ拡張モジュール (DEM) を追加すると、パネル 10 および 11 も取り付けられます。カートリッジ拡張モジュール (CEM) を BDM の右側に追加すると、パネル 14 および 15 が取り付けられます。BDM 上のセルおよびドライブのアドレスは決して変更しません。これにより、SL3000 の容量を増加したときに次を確実にします。ドライブおよびストレージセルの ACSLS アドレスは変更されません。

CAP 番号付け

CAP には、次のように番号が付けられます。

- CAP 0 は左側の AEM に予約されています
- CAP 1 - 5 - 左側の CEM 内、および DEM 内の CAP
- CAP 6 - 基本内の CAP (唯一の必須 CAP)
- CAP 7 - 10 - 右側の CEM 内の CAP
- CAP 11 は右側の AEM に予約されています

CAP が存在しない場合、ACSLs はそれらを「取り付けられていない」と報告します。これにより、新しい CAP を追加するときに、CAP の場所は SL3000 で変更しません。

CAP はパーティション専用でできます。

SL3000 モジュール

SL3000 ライブラリのモジュールには次の 5 つのタイプがあります。

- 基本モジュール、1 台、必須

基本モジュールは単一のフレームから構成されます。このモジュールは、電源構成、ロボット (TallBot)、電子モジュール、カートリッジアクセスポート、ストレージセル、テープドライブ、オペレータ制御を含む、ライブラリ内のほかのモジュールすべてに対するインフラストラクチャーを集約しています。

ライブラリを設置するごとに 1 台の基本モジュールが必要になります (1 台のみ)。

このモジュールの前面には:

- 単一の 26 カートリッジデュアルマガジンカートリッジアクセスポート (CAP)。
 - ライブラリにアクセスするための保守用ドア。
 - 3 つの LED (ライブラリアクティブ、保守必要、および待機) を備えたフロントパネル。
 - タッチスクリーンオペレータパネルまたはウィンドウのオプション機能。
- ドライブ拡張モジュール (DEM)、1 台、左サイドのみ

ドライブ拡張モジュール (DEM) は、左側にのみ、基本モジュールに隣接して接続できます。このモジュールを使用すると、テープドライブをさらに拡張し、データカートリッジ容量を追加できます。

- カートリッジ拡張モジュール (CEM)、左または右サイド

カートリッジ拡張モジュール (CEM) は、カートリッジセル容量の追加と拡張を実現します — このモジュールにはテープドライブはありません。基本モジュール (必須) およびオプションのドライブ拡張モジュール以外に、単一のライブラリで最大 4 台の CEM がサポートされています。

- パーキング拡張モジュール (PEM)、左端および右端のサイドのモジュール。

パーキング拡張モジュール (PEM) はカートリッジ拡張モジュールと同じですが、ライブラリが障害のあるロボットをパーキングできるようにアクセス不能になっている 6 列のアレイ

(前面壁の 3 列と背面壁の 3 列) を含みます。左側の PEM では、これは 3 つの左端の列 (前面および背面) です。右側の PEM では、3 つの右端の列です。

注:

デュアルロボットを装備した SL3000 には AEM と PEM のどちらかがあります。

- アクセス拡張モジュール (AEM)

AEM については、「[AEM の使用](#)」で説明されています。

新しいパネルタイプ

可能性のある各セルの場所を定義した、SL3000 における特定パネル定義の代わりに、汎用パネル定義が用意されています。

3 つの汎用パネルタイプが、さまざまな特定のパネルマップに対応します。それらの機能を次に示します。

- 単一の SL3000 セルパネルタイプ。

どのセルがこのライブラリ内に存在し、アクティブ化された容量の一部になっており、このパーティションに割り当てられているかを調べるには、ライブラリを監査 (*audit*) する必要があります。

- ドライブパネルタイプ。

ACSLs は、ライブラリの構成によってどのドライブが存在しているかを識別します。これを行う前に、すべてのドライブに電源が投入され、ライブラリの構成前の準備ができていることを確認する必要があります。

- 取り付けられていない。

これは、まだ取り付けられていないモジュール上のフロントパネルと背面パネルのプレースホルダです。

内部 SL3000 アドレス指定について

SL3000 の内部アドレスと、ACSLs および HSC でサポートされるほかのライブラリの内部アドレスには、違いがあります。

- SL3000 は 1 を基準にしており、負の数を使用します。
- その他のライブラリは 0 を基準にしており、負の数は使用しません。
- SL3000 は、ライブラリ、ルール、列、サイド、および行の 5 つのパラメータを使用します。

- その他のライブラリは、ACS、LSM、パネル、行、および列を使用します (HLI-PRC)。

ライブラリのモジュールは、カートリッジ、テープドライブ、カートリッジアクセスポート、ロボットユニットを格納した壁、列、および行から構成されます。

SL コンソールを使用すると、SL3000 内部アドレスと ACSLS のパネル、行、および列との間で変換できます。

詳細と手順は、『SL3000 ユーザーズガイド』を参照してください。

AEM の使用

AEM (アクセス拡張モジュール) は基本的には、オンライン/オフライン状態、パーティションで共有される機能など、CAP のすべての特性を備えた非常に大きな CAP です。

AEM を使用すると、最大 234 個のカートリッジのライブラリへの一括ロード、またはライブラリからの一括アンロードを一度に実行できます。また、ライブラリ TallBot の中断不要のメンテナンスも可能になります。

AEM は、機能の多くを回転式 CAP として共有していますが、いくつかの独特の特徴もあります。それらの機能を次に示します。

- アクセสดア
- 安全ドア
- CAP の操作
- 中断不要のメンテナンス

AEM の詳細は、『StorageTek SL3000 モジュラーライブラリシステムユーザーズガイド』(部品番号 316194401) を参照してください。

アクセสดア

AEM アクセสดアは、ライブラリへの一括ロードまたはライブラリからの一括アンロードを行うために開く外部ドアです。

安全ドア

安全ドアは、内部の「ガレージドア」であり、下降して AEM をライブラリのほかの部分から分離します。このドアは、ライブラリオペレータがカートリッジをロードまたはアンロードするか、ストレージ CSE が TallBot やほかの AEM コンポーネントに保守を実行するときに AEM の内部への安全なアクセスを確保します。

CAP の操作

AEM の利点は、次を実行できるようにすることです。

- 一括挿入

一括挿入を使用すると、最大 234 個のカートリッジを一度に挿入できます。

AEM `cap_id` を指定して通常の挿入を開始する必要があります。これは手動挿入にすることも、AEM を自動挿入モードに設定することもできます。

パーティション分割されたライブラリでは、CAP がパーティション専用の場合、CAP の自動挿入モードへの設定だけを行えます。AEM を自身のパーティション専用にすると、ほかのパーティションからは利用できなくなります。これを実行する前にその他の SL3000 ユーザーを考慮してください。

- 一括取り出し

42 を超えるボリュームを AEM に取り出す場合、大量の取り出し用の推奨のツールが、`lib_cmd` または `ejecting.sh` です。`cmd_proc`、`xeject`、ACSAPI 取り出しなどのほかのユーザーインタフェースでは、取り出し可能な数量は一度に 42 個のカートリッジに制限されます。「[lib_cmd の使用](#)」および「[ejecting.sh](#)」を参照してください。

- 中断不要のロボットメンテナンス

注意:

AEM を使用して一括取り出しを実行するときには、すべての CAP マガジンが存在していることを確認します。ACSL5 がカートリッジの取り出し (`eject`) を試みている場所で、CAP マガジンが見つからない場合、取り出し (`eject`) は失敗します。

注:

少数のカートリッジの挿入や取り出しには、AEM を使用しないでください。一括挿入および取り出しにのみ使用してください。

少数の挿入と取り出しに AEM を使用すると、ボリュームの挿入または取り外しで AEM にアクセスするときにサービス安全ドアを降下させ、AEM へのアクセスが終了したときに上昇させる必要があるため、非常に遅くなります。

少数の場合は、CAP 1 - 10 を使用してください。「[CAP 番号付け](#)」を参照してください。

少数の挿入と取り出しには AEM を使用しないことが必要になるので、最大の CAP 優先順位は 1 になります。これは、`audit`、`enter`、または `eject` でアスタリスクを使用して CAP ID がワイルドカード指定されている場合、AEM が選択されないようにするために役立ちます。

中断不要のメンテナンス

AEM を使用すると、CSE は、安全ドア (または「ガレージ」ドア) を使用してライブラリ TallBot にアクセスして保守できるため、ライブラリをオフラインにする必要はありません。TallBot に障害がある場合、ライブラリがオンラインになっている間に、TallBot が自身で AEM 内にパーキングします。冗長な TallBot が取り付けられている場合、ライブラリは、残りの動作している TallBot で通常の操作を続行できます。

CSE は、特別な鍵を使用して、安全ドアでのサービスアクセスロックをロック解除します。これにより、AEM 安全ドアは降下し、ライブラリのほかの部分から AEM を分離します。安全ドアが完全に降下したら、ライブラリがオンラインになっている間に、CSE は AEM 安全ドアを開き、障害のある TallBot またはその他の AEM のコンポーネントに対して安全にメンテナンスを実行します。メンテナンス期間中、AEM 自体はオフラインにされ、カートリッジのロード/アンロード機能は一時停止されます。

メンテナンスが完了すると、CSE は安全ドアを閉じ、安全ドアが上昇します。TallBot は再初期化され、AEM CAP カートリッジスロットに対して *audit* を実行し、AEM はオンラインに戻されます。

SL3000 CAP ID の別名処理

SL3000 には 12 台の CAP がありますが、バックアップアプリケーションの中には現在、LSM あたり最大 3 台の CAP しかサポートしていないものがあります。CAP 別名処理を使用すると、CAP 3 - 10 に、CAP 1、または 2 の別名を付けることができます。

CAP 0 はアクセス拡張モジュール (AEM) の一括ロード CAP に予約されています。SL3000 の CAP 0 には特別な属性があるため、CAP 0 を別名 CAP ID にすることはできません。

次のテンプレートを使用して、ライブラリ CAP ID を別名 CAP ID にマップします。

```
$ACS_HOME/data/external/SL3000/SL3000_CAP_Aliases.SAMPLE
```

注:

CAP ID を設定する場合:

ライブラリ CAP ID:

- LSM は 0 にする必要があります
- ID は 3 - 10 にする必要があります
- 重複したライブラリ CAP ID は使用できません

別名 CAP ID

- CAP 番号は 1 または 2 にする必要があります
- 重複した別名 CAP ID は使用できません

```
# 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.
```

上記の例では、変更された `SL3000_CAP_Aliases` ファイルを示しています。コメント (#) 文字は太字で示されている行から削除され、別名 CAP 番号の割り当てが有効になっていることを指定します。この例では、

- 物理的な CAP ID 0,0,4 に CAP 番号 1 の別名が付けられます。
- 物理的な CAP ID 0,0,6 に CAP 番号 2 の別名が付けられます。

パーティションからセルを取り外す前のカートリッジの移動

SL3000 はドライブおよびセルレベルまでパーティション分割できます。詳細は、[パーティションからセルを取り外す前のカートリッジの移動](#) を参照してください。

SL3000 CAP の動作

SL3000 は最大 12 のカートリッジアクセスポート (CAP) をサポートします。各 SL3000 パネルには 1 つの CAP を備えることができます。

パーティション分割されたライブラリでの CAP の動作については、「[ライブラリのパーティション分割またはパーティション ID の変更](#)」を参照してください。

注意:

すべての ACS には少なくとも 1 つの CAP が存在する必要があります。これは専用にするこも、共有することもできます。SL3000 内のすべての CAP はほかのパーティション専用に行けるため、これは SL3000 パーティションを構成するときに重要な制限になります。

見つからない SL3000 カートリッジを見つける

カートリッジが所定の場所がない、または ASSLS で行方不明になっている場合:

1. SL コンソールを使用して、SL3000 の物理的監査を実行します。

SL3000 の物理的監査は、`mount` の処理とその他のライブラリ操作リクエストの間のバックグラウンドタスクとして実行されます。

注意:

カートリッジを直接ロードするなどの手動の操作のために、SL3000 のコンテンツが ACSLS と同期しなくなった場合は、操作を継続しようとするはお勧めしません。

2. ACSLS `audit` を実行して、ライブラリカートリッジの実際のインベントリに一致するように、ACSLS データベースを更新します。

SL3000 のオフラインへの変更

SL3000 コンポーネントの電源を切断する前 (これらが動作していない場合)、または SL3000 アクセスドアを開く前に、ACSLs に対してこれらをオフラインに変更 (*vary*) します。これにより、これらが使用できないことが ACSLS に通知されます。それらが使用可能になると、オンラインに戻します (*vary*)。

SL コンソールではなく ACSLS を使用して SL3000 コンポーネントをオフラインに変更する

SL3000 コンポーネント (ACS、LSM、および CAP) を SL コンソールではなく ACSLS に対してオフラインに変更 (*vary*) します。

ACSLs を使用すると、*vary offline* の強制でないかぎり、コンポーネントをオフラインにする前に、未処理のリクエストを完了させることができます。ACSLs に対する未処理のリクエストは SL コンソールにはわかりません。

SL コンソールを使用してコンポーネントをオフラインに変更した場合、処理中のリクエストが失敗する原因になることがあります。

SL3000 コンポーネントを ACSLS に対してオフラインに変更する場合

このセクションでは、コンポーネントを ACSLS に対してオフラインに変更する場合について説明します。

アクセスドアを開く前

SL3000 アクセスドアを開く前に、次のコマンドを使用して ACS をオフラインに変更 (*vary*) します。

```
vary acs acs_id offline
```

注:

SL3000 内のすべての CAP が自動モードである場合、次の手順を行う必要があります。

1. アクセスドアを開く前にそれらを手動モードに設定します。
2. アクセスドアを閉じて SL3000 がオンラインに戻ったあとに、自動モードに戻します。

CAP が動作していない場合

CAP が動作していない場合、次のコマンドを使用してオフラインに変更します。

```
vary cap cap_id offline
```

動的構成 (config) ユーティリティの使用

動的構成 (*config*) ユーティリティを使用すると、ACSL S がオンラインで実行しているときに、ACSL S ライブラリ (およびコンポーネント) に構成の変更を実装できます。これらの構成変更は、*acsss_config.log* ファイルに記録されます。

次の動的構成ユーティリティがサポートされています。

- *config acs*
- *config drives*
- *config lsm*
- *config ports*

config ユーティリティの使用には次の利点があります。

- ACSLS を実行し続けながら、影響を受けないライブラリコンポーネントに対して *mount* リクエストを実行できます。
- ほかの構成情報はすべて変更せずに、指定したライブラリコンポーネントを再構成できます。たとえば、次を指定する場合:
 - ACS の場合、ほかの ACS の構成は影響を受けません。
 - LSM の場合、ほかの LSM の構成は影響を受けません。
 - ドライブパネル (パネル上のドライブ) の場合、すべての既存のドライブに対するマウントおよびマウント解除は影響を受けません。

新しいカートリッジアドレスの監査

拡張が行われたあとで、次の手順を完了します。

1. ACS とポートをオフラインに変更 (*vary*) します。
2. 次のいずれかを実行します。
 - モジュールの追加または削除
 - 容量の変更
 - ライブラリの再パーティション分割
3. ACS とポートをオンラインに変更 (*vary*) します。
4. 次の手順のいずれかを使用して、ACSL S データベースの構成を更新します。
 - 動的:

- `config acs acs_id` または `config lsm lsm_id`
- ライブラリを監査 (*audit*) します。
- 静的:
 - ACSLS を停止します: `acs disable`
 - ACSLS 構成を更新します: `acs config`
 - ACSLS を起動します: `acs enable`
 - ライブラリを監査 (*audit*) します。

ACSLS のデュアル TCP/IP

ACSLS サーバーとライブラリ間のデュアル TCP/IP 接続はオプションで、SL3000 用に購入できます。

SL3000 の要件

- システム管理者とネットワーク管理者の両方と協力して、現在のネットワーク環境について理解し、必要なすべての IP アドレスを事前に識別します。
- システム管理者と協力して、ネットワークインタフェースを構成するか、またはそれが正しく構成されていることを検証します。

この機能の詳細について「[概要](#)」と、*StreamLine Modular Library System デュアル TCP/IP* 機能のドキュメントを参照してください。

注:

デュアル TCP/IP 接続を実装する手順は、SL8500 ライブラリと SL3000 ライブラリの両方で同じです。

付録E SL500 の ACSLS サポート

SL500 ライブラリは、SCSI 接続の単一 LSM ライブラリです。制御パス操作に、ACSLS サーバーへのファイバチャネルまたは LVD SCSI 接続のどちらかが必要になります。この章では、SL500 ライブラリの ACSLS サポートについて説明します。

ACSLS 接続

ACSLS サーバーは、SL500 ライブラリ内のインタフェースカード (LVD SCSI またはファイバ) と互換性のあるホストバスアダプタ (HBA) が必要です。ACSLS は、インタフェースカードの代わりに、SL500 内のブリッジドライブとも通信できます。

`$ACS_HOME/install/install_scsi_sol.sh` を実行すると、新しいハードウェアの存在 (JNI カードまたは LSI Logic カードのいずれか) についてシステムにプローブが行われます。これらの HBA カードのどちらかがシステム上に存在しているが、対応するドライバが所定の位置にない場合は、このルーチンは適切なドライバパッケージを見つけ、それをインストールするための権限を求めるプロンプトを表示します。

注:

SL500 をパーティション分割できますが、ACSLS はパーティション分割した SL500 をサポートしません。

注:

制御パス用の SAS ブリッジドライブは、サポートされません。ブリッジドライブへのファイバチャネル接続のみがサポートされています。

ACSLS と SL500 ライブラリの相違点

このセクションでは、ACSLS と SL500 ライブラリの相違点について説明します。

ライブラリの構成

- ACS、LSM - SL500 ライブラリは、単一の LSM を持つ ACS です (1 ACS および 1 LSM)。
- SL500 は現在、パススルーポートをサポートしていません。
- SL500 には少なくとも 1 つの基本モジュールがあり、最大 4 個の拡張モジュールを追加できます。

- ACSLS は各 SL500 モジュールをパネルとみなします。
- ACSLS で管理される SL500 ライブラリは、ACSL S インタフェースを介して 6 文字の `volser` ラベルを報告します。

ライブラリの場所の特定

このセクションではライブラリの場所について説明します。

アドレス指定スキーム

ACSL S のアドレス指定は、ACS、LSM、パネル、行、列です。

- SL500 アドレス指定は、ライブラリ内の LSM、モジュール、行、列です。
- ACSLS のアドレスはゼロが基準です。パネル 0、行 0、列 0 から始まります。
- SL500 ライブラリは 1 が基準です。モジュール 1、行 1、列 1 から始まります。
- SL500 ライブラリと ACSLS の内部アドレス間で整合性を確保するために、ACSL S はパネル 0、行 0、列 0 に対してプレースホルダを定義します。

パネル

- パネル 0 = 取り付けられていません
- パネル 1 = 基本モジュール
- パネル 2 - 5 = 拡張モジュール

行番号

- 各モジュール (パネル) 内で、行には 1 - 12 の番号が割り当てられます。
- 行には、上から下に連続して番号が割り当てられます。
- 番号付けは SL500 と ACSLS で同一です。

列番号

- 1 から 11 (最大) の番号が付けられます。
- SL500 の現在の構成は 9 列に制限されています。
- 列 1 - 4 は左側の前から後ろを示します。
- 列 5 - 8 は右側の後ろから前を示します。
- 列 9 は、ドライブが配置されている背面壁です。
- 番号付けは SL500 と ACSLS で同一です。

ドライブのアドレス

SL500 アドレス指定は LSM, モジュール, 行, 列です

- 行はドライブ番号またはスロット番号に等しくなります。
- ドライブ番号は基本モジュールで 1 - 2、拡張モジュールで 1 - 4 です。
- 列番号はドライブの場合は常に 9 です。

ACSLS アドレス指定 = ACS, LSM, パネル, ドライブ

- ドライブ番号 = 行またはスロット

表E.1 ドライブのアドレス指定の例

モジュール	ACSLS のドライブ識別子	SL500 のドライブアドレス
基本モジュール	0, 0, 1, 1	0, 1, 1, 9
	0, 0, 1, 2	0, 1, 2, 9
ドライブ拡張モジュール	0, 0, 2, 1	0, 2, 1, 9
	0, 0, 2, 2	0, 2, 2, 9
	0, 0, 2, 3	0, 2, 3, 9
	0, 0, 2, 4	0, 2, 4, 9
次の拡張モジュール	0, 0, 3, 1	0, 3, 1, 9

ACSLS の制限事項

- SL500 ライブラリの最大数 = 31。
- LSM の最大数 = 127。
- SL500 ライブラリ内のパネルの最大数 = 5。
- 拡張モジュールには、それぞれ 1 つのパネルを追加します。
- 基本モジュールだけを含む SL500 にはパネルは 1 つだけです。
- SL500 ファームウェアは、ラベルのないカートリッジをサポートしません。これは、「venter」コマンドを使用できないことを意味します。
- パーティション分割された SL150 ライブラリは、サポートされません。
- 制御パス用の SAS ブリッジドライブは、サポートされません。

SL500 ライブラリ設定の構成

SL500 は、ACSL5 に影響を与える新しい構成設定をサポートしています。それらの機能を次に示します。

- ストレージとしての CAP – CAP セル
 - 拡張モジュールでのみ許可されます。
 - 基本モジュールの CAP には常に CAP として使用されます。
 - モジュールごとに構成されます。
 - ストレージとして使用される CAP はどれも、挿入および取り出し操作で使用できません。
- 予約済みセルはライブラリでの用途 (診断、クリーニングカートリッジ) にのみ使用します。
 - 基本モジュールだけに影響します。
 - 数 (n) は 0 から 9 (基本モジュールでの最大行数) まで構成できます。
 - 列 1 での最初の (n) 個のセルが予約されます。

これらのセルは ACSLS にアクセスできないとみなされます。

SL500 CAP の動作

基本モジュールの CAP には 5 スロットマガジンが 1 つあります。ドライブ拡張モジュールを追加した場合、ドライブ拡張モジュール内の CAP には 5 スロットマガジンが 2 つあります。拡張モジュールの場合、CAP は、拡張モジュールあたり 10 の追加ストレージスロットに対して構成できます。さらに:

- ブリッジドライブを通じて接続されている SL500 の CAP は、別の ACSLS インスタンスがライブラリの管理を引き継ぐときに、ロック状態になることがあります。詳細とこの問題の解決方法は、SL500 の付録で「[取り出し中に開かない CAP \(メールスロット\)](#)」を参照してください。
- CAP セル (基本モジュールのもの以外) は、CAP セルかストレージセルのどちらかとして構成できます。
- 基本モジュール CAP は常に CAP として使用されます。
- SL500 では、カートリッジの挿入および取り出しに 1 つのアクティブな CAP が必要です。
- 複数の CAP がある場合、SL500 ライブラリは常に、1 つの CAP があると ACSLS に報告します。
- すべての CAP セクションへのドアは、CAP がストレージセルとして構成されていないかぎり、一体としてロックおよびロック解除します。

CAP セルをストレージセルとして使用するようモジュールが構成されている場合は、CAP セクションは、CAP にアクセスするライブラリ操作の影響を受けません。

- ACSLS は CAP を監査するときに、すべてのセルを調べます。

ライブラリの監査

次の場合は、常に監査を実行する必要があります。

- これが新しいライブラリである場合。
- 1 つ以上のモジュールが追加、取り外し、またはスワップされた場合。
- ドアからカートリッジを手動で追加または取り外した場合。
- ライブラリ構成設定が変更されたあと。

次の場合に、ライブラリの自己監査が行われます。

- ライブラリの電源を投入するか、再初期化した場合。
- ドアを開いて閉じた場合。

自己監査が完了したあとで、ACSLS を使用してデータベースを更新する必要があります。

これが新しいライブラリである場合

新しいライブラリは、ライブラリの実際の内容をデータベースと同期するために監査する必要があります。

モジュールの追加、取り外し、またはスワップ後

SL500 モジュールが追加、取り外し、スワップされたあとで、またははじめてライブラリを監査している場合に、次の手順を実行します。

1. SL500 ライブラリを ACSLS に対してオフラインに変更 (*vary*) して、ライブラリの電源を切断します。
2. モジュールの追加などのライブラリの変更を実行します。
3. SL500 ライブラリの電源を投入するか、再初期化します。
4. ACSLS が実行されている場合、次のコマンドを使用して ACSLS を終了します。

```
acsss disable
```

5. 次のコマンドを使用することによって、*acsss_config* を実行します。
6. 次のコマンドを使用して ACSLS を起動します。

acs enable

7. 影響を受ける SL500 パネルの監査 (*audit*) を実行します。

ドアを通じた手動によるカートリッジの追加または取り外し後

ライブラリのドアから手動でカートリッジを LSM に対して追加したり取り外したりした場合、データベースを同期させる必要があります。「[query pool](#)」の説明に従って、*audit* を実行します。

ACSL *cmd_proc* を使用して、CAP を通じてカートリッジを LSM に対して挿入または取り出した場合、データベースは自動的に更新されます。

ライブラリ構成設定の変更後

ライブラリ構成設定を変更した場合は、*acs* *config* を使用して ACSL を再構成し、続いてライブラリを監査 (*audit*) します。次のライブラリ設定を変更したあと、ライブラリを再構成し監査 (*audit*) します。

- 予約済みのセルの数
- ストレージセルとしての CAP

次の手順を実行します。

1. SL500 ライブラリを ACSL に対してオフラインに変更 (*vary*) して、ライブラリの電源を切断します。
2. ライブラリの変更を実行します。
3. SL500 ライブラリを再初期化します。
4. ACSL が実行されている場合、次のコマンドを使用して ACSL を終了します。

acs *disable*

5. *acs* *config* を実行します。
6. 次のコマンドを使用して ACSL を起動します。

acs *enable*

7. 影響を受ける SL500 パネルの監査 (*audit*) を実行します。

付録F SL150 の ACSLS サポート

SL150 ライブラリは、SCSI 接続の単一 LSM ライブラリです。この章では、SL150 ライブラリの ACSLS サポートについて説明します。

ACSLS 接続

ACSLS サーバーには、SL150 ライブラリ内のブリッジドライブとの互換性がある FC ホストバスアダプタ (HBA) が必要です。

`$ACS_HOME/install/install_scsi_sol.sh` を実行すると、新しいハードウェアの存在 (JNI カードまたは LSILogic カードのいずれか) についてシステムにプロンプトが行われます。これらの HBA カードのどちらかがシステム上に存在しているが、対応するドライバが所定の位置にない場合は、このルーチンは適切なドライバパッケージを見つけ、それをインストールするための権限を求めるプロンプトを表示します。

注:

SL150 はパーティション分割できますが、ACSLS はパーティション分割した SL150 をサポートしません。

注:

制御バス用の SAS ブリッジドライブは、サポートされません。ブリッジドライブへのファイバチャネル接続のみがサポートされています。

ACSLS と SL150 ライブラリの相違点

このセクションでは、ACSLS と SL150 ライブラリの相違点について説明します。

ライブラリの構成

- ACS、LSM - SL150 ライブラリは、単一の LSM を持つ ACS です (1 ACS および 1 LSM)。
- SL150 は現在、パススルーポートをサポートしていません。
- SL150 には少なくとも 1 つの基本モジュールがあり、最大 9 個の拡張モジュールを追加できます。
- ACSLS は各 SL150 モジュールをパネルとみなします。

- ACSLS で管理される SL150 ライブラリは、ACSL S インタフェースを介して 6 文字の `volser` ラベルを報告します。

ライブラリの場所の特定

このセクションではライブラリの場所について説明します。

アドレス指定スキーム

- ACSLS のアドレス指定は、ACS、LSM、パネル、行、列です。
- SL150 のアドレス指定は、モジュール、サイド、行、ストレージセルの列です。
- ACSLS のアドレスはゼロが基準です。パネル 0、行 0、列 0 から始まります。
- SL150 ライブラリは 1 が基準です。モジュール 1、行 1、列 1 から始まります。
- SL150 ライブラリと ACSLS の内部アドレス間で整合性を確保するために、ACSL S はパネル 0、行 0、列 0 に対してプレースホルダを定義します。

パネル

- パネル 0 = 取り付けられていません
- パネル 1 = 基本モジュール
- パネル 2-10 = 拡張モジュール

行番号

- 各モジュール (パネル) 内で、行には 1-3 の番号が割り当てられます。
- 行には、上から下に連続して番号が割り当てられます。
- 番号付けは SL150 と ACSLS で同一です。

列番号

SL150 のアドレス指定:

- モジュール, サイド, 行, 列
- 列 1-5 (各サイドの前から後ろ)

ACSL S のアドレス指定:

- 1 から 10 の番号が付けられます。
- 列 1-5 は左側の前から後ろを示します。

- 列 6-10 は右側の前から後ろを示します。

ドライブのアドレス

SL150 のアドレス指定:

- モジュール、位置
- モジュール 1 = 基本
- モジュール 2-10 = 拡張
- 位置 = 上から下

ACSLS のアドレス指定:

- ACS, LSM, パネル, ドライブ
- パネル = モジュール
- ドライブ番号 = 1-20 (番号は上から下の順で付けられます)

表F.1 ドライブのアドレス指定の例

モジュール	ACSLS のドライブ識別子	SL150 のドライブアドレス
基本モジュール	0, 0, 1, 1	1, T
	0, 0, 1, 2	1, B
拡張モジュール	0, 0, 2, 3	2, T
	0, 0, 2, 4	2, B
拡張モジュール	0, 0, 3, 5	3, T
	0, 0, 3, 6	3, B
拡張モジュール	0, 0, 4, 7	4, T
	など。	など。

ACSLS の制限事項

- SL150 ライブラリの最大数 = 31。
- LSM の最大数 = 127。
- SL150 ライブラリ内のモジュール (パネル) の最大数 = 10。
- 拡張モジュールには、それぞれ 1 つのパネルを追加します。
- 基本モジュールだけを含む SL150 にはパネルは 1 つだけです。

- SL150 ファームウェアは、ラベルのないカートリッジをサポートしません。これは、*venter* コマンドを使用できないことを意味します。
- パーティション分割された SL150 ライブラリは、サポートされません。
- 制御パス用の SAS ブリッジドライブは、サポートされません。

SL150 ライブラリ設定の構成

SL150 は、ACSL5 に影響を与える構成設定をサポートしています。

- パーティション分割

パーティション分割された SL150 ライブラリは ACSLS によってサポートされていません。

- ブリッジドライブ

ACSL5 は FC 制御パスをサポートしています。制御パス用の SAS ブリッジドライブは、サポートされません。

- メールスロットの構成

標準メールスロットの拡張機能として、基本モジュールの一方のサイドにあるストレージスロットを使用できるようにします。

- ACSLS の必須設定: 標準メールスロット (4 スロット)

- ドライブ要素アドレス指定モード

空のドライブスロットがライブラリ内に存在する場合、SCSI 要素のアドレスを報告する方法を制御できるようにします。

- ACSLS の必須設定: 取り付けられているドライブだけのアドレス指定

- 予約済みセル

予約済みセルは、ライブラリでの用途 (診断およびクリーニングカートリッジ) にのみ使用します。

- 数 (n) は 0 から 3 (基本モジュールでの最大行数) まで構成できます。
- 基本モジュールにのみ影響 - 列 1 の最初の (n) 個のセルが予約されています。
- これらのセルは ACSLS にアクセスできないとみなされます。

SL150 CAP の動作

SL150 では、CAP に対して「メールスロット」という用語を使用します。基本モジュールのメールスロットには 4 スロットマガジンが 1 つあります。

すべてのメールスロット操作は、ライブラリのタッチスクリーンまたは BUI のどちらかを使用して実行されます。

- SL150 には、カートリッジの挿入および取り出しに 1 つのアクティブなメールスロットが必要です。
- ACSLS はメールスロットを監査するときに、すべてのセルを調べます。

取り出し中に開かない CAP (メールスロット)

以前にライブラリを管理していた ACSLS のインスタンスによって SL150 CAP (メールスロット) がロックされたままである場合に、取り出し (*eject*) (または挿入 (*enter*)) 操作中に問題が起きる可能性があります。この問題が起きる可能性のあるシナリオには、新しい ACSLS サーバーへの移行、つまり HA フェイルオーバーイベントが含まれます。

メールスロットがホストによってロックされている場合、ライブラリのタッチスクリーンディスプレイには、次のように表示されます。

状態: SCSI 防止メディアによってロックされている

ACSLS は通常、自動 CAP をロックしたままにしないので、自動モードの CAP でこの問題が見られる可能性は高くありません。この問題は、手動モードの CAP、またはオフラインに変更されたライブラリ内の CAP (どちらかのモード) で起きる可能性が高くなっています。

軽減:

この問題が発生しないようにするために、次の手順をお勧めします。

- SL150 ライブラリの制御を新しい ACSLS サーバーに移行する場合 (非 HA 構成):
 - 自動モード CAP (推奨):

元のサーバーで ACSLS をシャットダウンする前に、ライブラリをオフラインに変更しないでください。ACSLS は、ライブラリがオンラインの場合、シャットダウン時に CAP をロック解除したままにします。

- 手動モード CAP:

元のサーバーで ACSLS をシャットダウンする前に、CAP を自動モードで動作するように設定します。ライブラリをオフラインに変更しないでください。

- ACSLS HA インストールでライブラリを管理するときには、自動モードで CAP を操作することを強くお勧めします。
 - 自動モード CAP (推奨):

オンラインライブラリにはアクションは必要ありません (CAP は通常ロック解除されています)。フェイルオーバーイベントの時点でライブラリがオフラインだった場合、次の手順に従って、ブリッジドライブを再起動することにより CAP のロックを解除します。

。 手動モード CAP:

次の手順に従って、ブリッジドライブを再起動することにより CAP のロックを解除します。

解決方法:

以前のホストにより保持されていたロックは、SL150 BUI を使用してブリッジドライブを再起動することによって解除できます。ブリッジドライブの再起動の詳細は、SL150 の製品ドキュメントを参照してください。

警告:

この操作は、ドライブでの現在のデータパス操作をすべて中断させます。データパスのアクティビティが中断されないときに、ブリッジドライブの再起動をスケジュールできます。

この問題が発生し、ロックを解除する必要がある場合:

1. データパス (読み取り/書き込み) アクティビティがブリッジドライブで発生していることを確認します。
2. SL150 BUI を使用して、ブリッジドライブを再起動します。

ライブラリの監査

次の場合に、監査を実行します。

- これが新しいライブラリである場合。
- 1 つ以上のモジュールが追加、取り外し、またはスワップされた場合。
- メールスロットからカートリッジを手動で追加または取り外した場合。
- ライブラリ構成設定が変更されたあと。

次の場合に、ライブラリの自己監査が行われます。

- ライブラリの電源を投入するか、再初期化した場合。
- メールスロットを開いて閉じた場合。

自己監査が完了したあとで、ACSL5 を使用してデータベースを更新する必要があります。

これが新しいライブラリである場合

新しいライブラリは、ライブラリの実際の内容をデータベースと同期するために監査する必要があります。

モジュールの追加、取り外し、またはスワップ後

SL150 モジュールが追加、取り外し、スワップされたあとで、またははじめてライブラリを監査している場合に、次の手順を実行します。

1. SL150 ライブラリを ACSLS に対してオフラインに変更 (*vary*) して、ライブラリの電源を切断します。
2. モジュールの追加などのライブラリの変更を実行します。
3. SL150 ライブラリの電源を投入するか、再初期化します。
4. ACSLS が実行されている場合、次のコマンドを使用して ACSLS を終了します。

```
acsss disable
```

5. *acsss_config* を実行します。
6. 次のコマンドを使用して ACSLS を起動します。

```
acsss enable
```

7. 影響を受ける SL150 パネルの監査 (*audit*) を実行します。

メールスロットを通じた手動によるカートリッジの追加または取り外し後

ライブラリのメールスロットから、手動でカートリッジを LSM に対して追加したり取り外したりした場合、データベースを同期させる必要があります。「[query pool](#)」の説明に従って、*audit* を実行します。

ACSLS の *cmd_proc* を使用し、メールスロットを通じてカートリッジを LSM に対して挿入または取り出した場合、データベースは自動的に更新されます。

ライブラリ構成設定の変更後

ライブラリ構成設定を変更した場合は、*acsss_config* を使用して ACSLS を再構成してから、ライブラリを監査 (*audit*) する必要があります。予約済みセルの数を変更したあとでも、ライブラリを再構成して監査 (*audit*) する必要があります。

次の手順を実行します。

1. SL150 ライブラリを ACSLS に対してオフラインに変更 (*vary*) して、ライブラリの電源を切断します。
2. ライブラリの変更を実行します。
3. SL150 ライブラリを再初期化します。
4. ACSLS が実行されている場合、次のコマンドを使用して ACSLS を終了します。

acsss disable

5. *acsss_config* を実行します。
6. 次のコマンドを使用して ACSLS を起動します。

acsss enable

7. 影響を受ける SL150 パネルの監査 (*audit*) を実行します。

付録G StorageTek Virtual Tape Library の ACSLS のサポート

ACSLS は、FalconStor ベースの VTL 製品ラインに実装されているような一般的な仮想ライブラリを識別する新しい LSM タイプ (StorageTek VTL) をサポートします。StorageTek VTL ライブラリは、ほかの単一 LSM、SCSI 接続のライブラリと同様に、構成、管理、および操作できます。

ACSLS は、馴染みのある PRC スタイルの識別子 (*panel*、*row*、*column*) を使用して、StorageTek VTL 構成を表します。StorageTek VTL ライブラリに対して ACSLS または ACSAPI リクエストを発行するときに、これらの PRC スタイルの識別子を使用します。これらには、セル識別子 (*acs*、*lsm*、*panel*、*row*、*column*)、ドライブ識別子 (*acs*、*lsm*、*panel*、*transport*) および CAP 識別子 (*acs*、*lsm*、*cap*) が含まれます。

サポートされている構成

StorageTek VTL 構成は、ACSLS に対してライブラリが構成されているときに、*acsss_config.log* ファイルに記録されます。このファイルは、通常、診断目的にのみ使用されます。パネルの説明は、動的構成 (*config* ユーティリティを使用) 中にも表示されます。

次の表では、ACSLS によってサポートされている VTL 構成について説明します。

ストレージセルの数	開始要素アドレス	ドライブの数	開始要素アドレス	インポート/エクスポート要素の数	開始要素アドレス
1 - 10,000	1000	1 - 100	500	1 - 20	10

- ストレージセルの数: 1 から 10,000
- 開始要素アドレス: 1000
- ドライブの数: 1 から 100
- 開始要素アドレス: 500
- インポート/エクスポート要素の数: 1 から 20
- 開始要素アドレス: 10

次に、ライブラリコンポーネントのサポートについて説明します。

- パネル

サポートされている各構成は、次のような一連のパネルとして示されます。

- パネル 0 - CAP のみが含まれます
- パネル 1 - 10 - ストレージセルおよびドライブが含まれます
- パネル 11 - 50 - ストレージセルが含まれます

- CAP

1 つの CAP がサポートされています。これは CAP 0 として識別され、最大 20 の CAP セルを含めることができます。CAP は、StorageTek VTL のパネル 0 上に定義されています。

- ストレージセル

ストレージセルは、パネル当たり 200 個のセルを含むパネルに編成されています。最大の構成には、200 セルのストレージパネルが 50 含まれ、合計 10,000 個のセルに対応します。

ストレージセルは、StorageTek VTL のパネル 1 - 50 上に定義されています。

- ドライブ

ドライブは、パネル当たり最大 10 台のドライブを含むパネルに編成されます。最大のドライブ構成には、10 のドライブパネルが含まれ、合計 100 台のドライブに対応します。

ドライブは、StorageTek VTL のパネル 1 - 10 上に定義されています。

- PTP

パススルーポートはサポートされていません。

VTL の動作

次の動作は、特に VTL に適用されます。

- 挿入操作

ACSL5 での *enter* 操作は、StorageTek VTL ライブラリにテープボリュームを追加する場合には役立ちません。VTL ライブラリ内の新しいボリュームを検出するには、そのライブラリに対して *audit* を実行する必要があります。これには、VTL 仮想ポルトからインポートされている可能性があるボリュームの検出が含まれます。

enter 操作は、StorageTek VTL の場合、明示的に拒否されません。*enter* 操作を開始した場合は、単に取り消すことができます。

- 取り出し操作

ACSLS の *eject* 操作はサポートされています。取り出されたボリュームは、VTL 製品では、現在仮想ボルト内に存在していると表示されますが、これは VTL 専用の機能であり、ACSLS では仮想ボルトは認識されません。

- マウントおよびマウント解除操作

StorageTek VTL ライブラリでは、ACSLS *mount* オプション (*read-only*、*bypass*) および *dismount* オプション (*force*) は受け入れられますが、無視されます。さらに、(*force* オプションを使用しない) 通常のマウント解除が、*force* オプションを指定していた場合と同様に機能しているように見えます。

注意:

データパス操作が進行中のときに仮想テープボリュームをマウント解除する可能性があるため、マウント解除操作を実行するときには注意してください。

- 仮想テープドライブ

VTL 内で仮想ドライブを追加または削除したあとで、*config* ドライブユーティリティーを使用して、ACSLS 内のドライブ構成を再確認します。仮想テープドライブを VTL から削除するときに、VTL コンソールを使用して、ドライブを割り当て解除するだけでなく、ドライブを削除する必要があります。仮想ドライブの割り当てまたは割り当て解除は、データパスに影響を与え、ACSLS には影響を与えません。

ACSLS に対する VTL の構成

このセクションでは、VTL を ACSLS に対して構成するときの前提条件とインストール手順について説明します。

前提条件

VTL を ACSLS に対して構成する前に、次の前提条件を満たしている必要があります。

- ACSLS がインストールされていて、ファイバチャネル HBA が含まれています
- VTL システムがインストールされています
- システム間にファイバチャネル接続が存在しています (ACSLS イニシエータポートを VTL ターゲットポートに接続する必要があります)

インストール

VTL を ACSLS に対して構成するには、次の手順を実行します。

1. VTL コンソールを使用して仮想ライブラリ (StorageTek-VTL) を作成します。

注:

- ACSLS が使用する仮想テープライブラリを作成する場合、そのライブラリのテープ識別子は、ACSL S でサポートされている従来の 6 文字のボリュームラベル形式に適合している必要があります。
- VTL コンソールを使用すると、ACSL S によるサポートの制限 (10,000 スロット、100 ドライブ) を超える StorageTek VTL ライブラリを作成できます。ただし、ACSL S に対してサポートされていない構成を行おうとすると、構成リクエストは失敗します。
- 新しい StorageTek VTL ライブラリを作成するときに、VTL コンソールは、スロット数が 678 を超えたときに警告するように設定されています。この警告制限は L700 の容量に基づいており、StorageTek VTL ライブラリモデルの場合は無視できます。

手順については、*StorageTek Virtual Tape VTL Plus 2.0 (Update 2)* ライブラリソフトウェアの構成ガイドを参照してください。

2. VTL ライブラリを ACSLS に割り当てます。

手順については、*StorageTek Virtual Tape VTL (Plus 2.0 Update 2)* ライブラリソフトウェアの構成ガイドを参照してください。

3. SCSI mchanger デバイスドライバを作成します。

ACSL S は、StorageTek VTL ライブラリを自動的に検出して、対応する `/dev/mchanger` エントリを作成できます。この検出は、インストール中に `install_scsi_sol.sh` スクリプトを実行しているときに行われます。

[「ライブラリハードウェアへの接続のインストール」](#)を参照してください。

4. ACSLS に対して VTL を構成します。

- VTL コンポーネントが動作するようになったあとで、ACSL S に対して VTL を構成します。
- 次のコマンドを使用します。

```
acsss_config
```

詳細は、「[CSI チューニング変数の設定](#)」または `config acs new` ([「新しい ACS の追加」](#)を参照) を参照してください。

5. ライブラリを監査します。

ACSLS に対して StorageTek VTL ライブラリを構成したあと、仮想テープボリュームを検出し、アクセス不可のセル (パネルには部分的にのみアクセスできることがあります) を特定するために、*audit* を実行する必要があります。

6. `display lsm * -f type serial_num` コマンドを使用して、VTL 構成を表示します。

付録H 論理ライブラリのサポート

この付録では、論理ライブラリのサポートについて説明します。

論理ライブラリについて

ACSLs 物理ライブラリ構成の一部は、SAN 上のターゲットデバイスとして動作できる論理ライブラリとして SCSI クライアントに提供できます。論理ライブラリは、実際には、テープドライブやボリュームなどの、ユーザーに割り当てられた特定のリソースを含む物理ライブラリの定義された部分です。論理ライブラリは、SCSI (ファイバ接続) インタフェース経由でクライアントアプリケーションソフトウェアに公開されます。

ファイバチャネル HBA ポートは通常、イニシエータモードで動作し、ディスクドライブ、テープドライブ、メディアチェンジャーデバイスなどのリモートのターゲットデバイスに SCSI リクエストを発行できます。論理ライブラリを使用するには、1 つ以上のファイバポートを、代わりにターゲットモードで動作するように構成する必要があります。

install.sh または getHba.sh ユーティリティの実行時に、ターゲットモード動作のための 1 つまたは複数の FC ポートを選択するか、またはターゲットモードポートをイニシエータモードに戻すことができます。いずれの変更を有効にする場合もリポートが必要です。

論理ライブラリは、適格な任意の物理ライブラリをバックアップ ACS として使用して作成できます。バックアップ ACS は、論理ライブラリの作成元の物理ライブラリを識別します。バックアップ ACS は:

- ACSLS ライブラリサーバーに対して構成する必要がありますが、論理ライブラリを作成するためにそれがオンラインである必要はありません。
- 物理 ACS 全体にするか、またはライブラリ自体が物理パーティション分割をサポートしている場合は ACS の物理パーティションにできます。

論理ライブラリを作成、管理、または削除するには、ACSLs グラフィカルユーザーインタフェース (GUI) または lib_cmd コマンド行インタフェース (CLI) のどちらかを使用できます。

利点

論理ライブラリを使用すると、次のことを実行できます。

- 物理ライブラリを論理ライブラリにパーティション分割します。

論理ライブラリは、クライアントアプリケーションが、あたかも個別のライブラリであるかのように管理および使用できます。

ACSL S は、特定の物理ストレージの場所に縛られない柔軟なパーティション分割メカニズムを提供します。論理ライブラリはボリュームとドライブに関して定義され、ACSL S がこれらを、論理的な場所を使用してクライアントに提供します。論理ライブラリは、パネルやレール境界などの、ACS 内のどの物理的な部分によっても制限されません。

論理ライブラリはまた、中断を伴う可能性のある変更からもクライアントを保護します。ACSL S がボリュームの新しい物理ホームセルを選択する場合、それを互換性のあるドライブに近い場所に保持しても、あるいは既存の論理ライブラリが拡張されても、クライアントアプリケーションには影響しません。

- 8 文字のバーコード全体を *volser* として報告します。

論理ライブラリを作成する場合は、クライアントのボリュームラベル形式として 6 文字、8 文字接頭辞、8 文字接尾辞、またはすべてを指定できます。8 文字のバーコードには、*volser* のあとまたは前にあるメディアドメインとタイプが含まれます。

注:

6 文字より長いボリュームラベルは ACSAPI クライアントからアクセスできません。

- ドライブおよびボリュームへのクライアントアクセスを管理します。

論理ライブラリに割り当てられたドライブおよびボリュームは、そのライブラリを使用しているクライアントからのみアクセスでき、その他の FC または ACSAPI クライアントには表示されません。ただし、ACSL S GUI と *cmd_proc* はシステム管理者の権限で動作するため、これらのインタフェースでは表示できます。

- 複数イニシエータのサポートを提供します。

論理ライブラリは、複数のクライアント (特に、複数のイニシエータポート) からアクセス可能にできます。その目的は、複数のクライアントによる論理ライブラリの同時操作を許可することではなく、クライアント側の冗長性をサポートすることです。ある特定の時点では、1 つのクライアントシステムのみが論理ライブラリを操作するべきです。

複数のイニシエータを許可することは、1 つのクライアントシステムが複数の FC HBA またはポート経由で、割り当てられている論理ライブラリにアクセスできることを示します。ク

クライアント環境がフェイルオーバー機能をサポートしている場合、論理ライブラリは、新しいアクティブなクライアントシステムからただちにアクセスできます。

論理ライブラリに関して発生したユニットアテンションやチェック条件はすべて、構成されている各接続上で提供されます。

制限

論理ライブラリは次のとおりです。

- ACSLS SCSI メディアチェンジャークライアントインターフェースを使用しているクライアントからのみアクセスできます。レガシー ACSAPI を使用するクライアントでは使用できません。
- 複数の物理 ACS (または物理パーティション) にまたがることはできません。
- 物理 ACS 内の特定のストレージセルを予約することはできません。同様の結果を実現するためには、論理および物理パーティション分割を組み合わせる方法があります。
- ターゲットモードの FC ポートが、特定の Solaris コマンド (fcinfo など) の出力には含まれなくなる可能性があります。

論理ライブラリの作成

ACSL GUI または `lib_cmd` を使用して論理ライブラリを作成する前に、`acsss_config` または `config acs` ユーティリティを使用して物理ライブラリが作成されている必要があります。また、論理ライブラリを作成する前に、監査 (`audit`) も実行してください。

論理ライブラリを作成する場合は、次を実行します。

- 物理 ACS を指定します
- 論理ライブラリの属性を指定します
- 1 つ以上の物理ドライブを割り当てます
- 1 つ以上の物理ボリュームを割り当てます
- 論理ライブラリへのクライアントアクセスを指定します

物理 ACS の指定

論理ライブラリのバックアップ ACS を指定すると、以降の手順で論理ライブラリに割り当てることができるドライブおよびボリュームが自動的に制限されます。その ACS 内にあり、かつまだ別の論理ライブラリに割り当てられていないドライブおよびボリュームのみを使用できます。

論理ライブラリが作成されると、ACSLs によって論理 ACS 番号が割り当てられます。論理ライブラリの ACS 番号の範囲は n001-n999 です。ここで、n = (1 + バックリング物理 ACS ID) です。

例:

- 物理 ACS 0 によってバックリングされた論理ライブラリは 1001 になる可能性があります
- ACS 4 によってバックリングされた論理ライブラリは 5001 になる可能性があります

論理ライブラリの属性の指定

論理ライブラリには、次の属性を指定できます。

- 論理ライブラリの一意の名前
- 容量

これは、論理ライブラリでいつでもアクセスできるボリュームの最大数 (インベントリになることのできるストレージ要素の数) です。最小は 0、最大は 64,536 です。デフォルトはありません。

この容量は、論理ライブラリのサイズを制限するため (実際の物理容量より小さい制限の設定)、または物理容量を過剰に割り当てるため (実際に使用可能な容量より大きい制限の設定) の両方に使用できます。

- インポート/エクスポートセルの数

最小は 2、最大は 400 です。デフォルト値は 2 です。

- ドライブスロットの数

これは、割り当てることのできるドライブの最大数です。最小は 0、最大は 500 です。デフォルトはありません。

- ラベル形式

現在、物理ライブラリ内の長いボリュームラベルのサポートは、ライブラリファームウェアと構成に依存しています。

ただし、論理ライブラリは、バーコード全体 (6 文字の volser、および接尾辞として付加されたメディアドメインとタイプ) を報告できます。これにより、8 文字の volser である xxxxxxL4 が提供されます。論理ライブラリは構成可能であるため、メディアドメインとタイプは volser のあとか、または volser の前 (L4xxxxxx など) のどちらにも付加できます。

サポートされているラベル形式は、6 文字、8 文字接尾辞 (デフォルト)、および 8 文字接頭辞です。

1 つ以上の物理ドライブの割り当て

バッキング ACS に取り付けて構成された物理ドライブはすべて、論理ライブラリに追加できません。論理ライブラリのドライブタイプに制限はありません。

論理ライブラリに割り当てられた物理ドライブは次のとおりです。

- ACSAPI クライアントからはアクセスできなくなります。

物理ライブラリは、論理ライブラリに割り当てられていないすべてのドライブおよびボリュームとともに、引き続き ACSAPI クライアントからアクセスできます。

- 排他的に割り当てられます。

これらは論理ライブラリ間で共有できません。

論理ライブラリの作成時に指定されたドライブスロットの数より多い物理ドライブを論理ライブラリに割り当ててはできません。ただし、論理ライブラリには空のドライブスロットを含めることができるため、少ないドライブを割り当てては可能です。

注:

ロックされたドライブを論理ライブラリに割り当ててはできません。(ロックされたドライブは、ある ACSAPI クライアントによってすでに使用されています。)

ドライブスロットの数は、いつでも増やしたり減らしたりできます。スロットの数を、現在割り当てられている物理ドライブの数未満の値に減らす場合は、まずいくつかのドライブを割り当て解除する必要があります。

1 つ以上の物理ボリュームの割り当て

論理ライブラリを作成する場合は、その容量、つまり、いつでもアクセスできるボリュームの最大数を指定します。ボリュームを、論理ライブラリに割り当てることによってアクセス可能にします。

割り当ては排他的です。カートリッジは、1 つのライブラリにしか割り当てできません。バッキング ACS 内に存在し、まだどの論理ライブラリにも割り当てられていないほとんどのカートリッジを割り当てることができます。

次のボリュームは割り当てることができません。

- クリーニングカートリッジ (ACSLs およびライブラリハンドルのクリーニング)。
- ロックされたボリューム (ロックは ACSAPI クライアントによる使用を示します)。
- 所有されたボリューム (所有者は ACSAPI クライアントによる使用を示します)。

論理ライブラリ内のアクセス可能なボリュームの数が最大容量に達したら、それ以上カートリッジを追加することはできません。ライブラリ容量を増やすか、または使用されている領域の一部を解放してください。

論理ライブラリ内の領域は、割り当てられたボリュームにアクセスできなくなると解放されます。これは、割り当てられたボリュームが次の状態になったときに発生する場合があります。

- FC クライアントによって *eject* とマークされた。
- ライブラリから取り出された。
- ACSLS によって不在とマークされた。

割り当てられているが、アクセスできないボリュームは、再度アクセス可能になった (たとえば、取り出されたボリュームをバッキング ACS に再挿入できる) 場合や、論理ライブラリに空き領域ができた場合は、自動的に再度アクティブ化されます。

最後に、ボリュームを割り当て解除することによって、論理ライブラリ内の領域を解放できます。カートリッジは、使用中でない場合 (論理ドライブにマウントできない場合や、取り出し (*eject*) 操作のために選択できない場合など) に論理ライブラリから割り当て解除できます。

注:

- 論理ライブラリからボリュームを割り当て解除しても、削除や取り出しといった種類の操作が行われるわけではありません。ボリュームを取り出しても、割り当て解除操作が行われるわけではありません。
- 物理ストレージセルは事前に割り当てられたり、論理ライブラリに割り当てられたりしないことに注意してください。論理ライブラリには特定のセルではなく、特定のボリュームが含まれます。セルは、割り当てられたボリュームを格納したときに論理ライブラリに「属する」だけです。
- バッキング ACS を共有する論理ライブラリ間でアクセス可能なボリュームの総数が、バッキング ACS の物理容量を超えることはありません。

論理ライブラリに割り当てられたボリュームの数は、*lib_cmd* か、または GUI の「Logical Library Detail Information」ページから表示できます。割り当てられたボリュームは、次の 2 つの個別の数として示されます。

- 「Accessible Volumes」 - これは、割り当てられていて、ライブラリ内にあり、かつ一般にクライアントから使用できるボリュームを表します。
- 「Inaccessible Volumes」 - この数は、割り当てられているが、次のいずれかの理由のためにクライアントから使用できないボリュームを表します。1) ボリュームが不在または取り出し済みとしてマークされている、2) ボリュームが FC クライアントによる取り出しとしてマークされている、3) ボリュームが再挿入されたが、論理ライブラリがいっぱいだった、4) ボリュームが間違った物理ライブラリに再挿入された。

ストレージスロットの数は、いつでも増やしたり減らしたりできます。スロットの数を、現在割り当てられている物理ボリュームの数未満の値に減らす場合は、まずいくつかのボリュームを割り当て解除する必要があります。

クライアント接続の指定

ACSL SCS I メディアチェンジャーインタフェースは、論理ライブラリを、ファイバチャネル (FC) 接続経由でクライアントから使用できるようにします。FC クライアントが論理ライブラリにアクセスできるようにするには、ターゲットモードで動作するように構成された少なくとも 1 つの FC ポートが必要になります。

ACSL S インストールプロセスの `install.sh` の実行中にターゲットポートを構成した場合は、引き続き論理ライブラリを作成します。まだターゲットポートを構成していない場合は、`getHba.sh` を使用してターゲットポートを構成します。次に、ACSL S をシャットダウンして ACSL S サーバーをリブートします。詳細は、「[FC ポートをターゲットモードで構成する方法](#)」を参照してください。

ターゲットポートを構成したあとは、論理ライブラリの作成または更新時に GUI でクライアント接続を管理することが容易になります。GUI 画面には、ACSL S システム上で使用可能なターゲットポートのリストが表示されるほか、FC 接続経由でこれらのポートについて照会したすべてのクライアントのリストも表示されます。ACSL S はこれらの照会を自動的に検出し、各クライアントの WWN を記憶します。

クライアントに論理ライブラリを割り当てるには、GUI の該当するボックスにチェックマークを付けます。

ターゲットポートを構成したあとは、論理ライブラリの作成または更新時に `lib_cmd` または GUI からクライアント接続を管理することが容易になります。GUI と対話型モードの `lib_cmd` のどちらにも、ACSL S システム上で使用可能なターゲットポートのリストが表示されます。また、FC 接続経由でこれらのポートについて照会したすべてのクライアントのリストも表示されます。ACSL S はこれらの照会を自動的に検出し、各クライアントの WWN を記憶します。

クライアントに論理ライブラリを割り当てるには、`lib_cmd` で `create mapping` を使用するか、または GUI の「接続を管理」ページで該当するボックスにチェックマークを付けます。

各クライアントシステムを識別するために、わかりやすい「別名」を WWN の代わりに使用できます。別名を割り当てるには、`lib_cmd` で `edit initiator` を使用するか、または GUI の「Manage Connections」ページで「`Edit Initiator Alias`」操作を選択します。

高速ロード

ACSL S 高速ロード設定は、移動先要素がテープドライブであるときに `move` コマンドの完了がどのように報告されるかを制御します。

高速ロードが有効になっている場合は、操作が検証されて ACSLS によって受け入れられると成功ステータスが返されますが、それはカートリッジの移動が開始される前です。移動中に何らかのエラーが発生した場合、ACSL S はその情報を報告しません。クライアントは、ボリュームがいつロードされて使用可能になったかを識別し、何らかのエラーが発生した場合はそのリクエストをタイムアウトすることに責任を負っています。

高速ロードが無効になっている場合 (デフォルト設定)、成功ステータスは、物理ライブラリが移動の完了を報告するまで返されません。ただし、移動中にエラーが発生した場合、ACSL S はその情報をクライアントに報告します。

物理ライブラリは、ACSL S が成功ステータスを返すまでに必要な時間に影響を与える場合のある独自の高速ロードオプションを提供する可能性があります (ただし、ACSL S 高速ロードが無効になっている場合のみ)。ACSL S 高速ロードが有効になっている場合、ライブラリ設定はクライアント通知に影響を与えません。

論理ライブラリの削除

論理ライブラリは、必要でなくなった場合は ACSLS 構成から削除できます。論理ライブラリを削除すると、論理ドライブ、ボリューム割り当て、およびクライアントマッピングもすべて削除されます。この論理ライブラリに割り当てられていた番号は、新しい論理ライブラリが作成されるときに使用できるようになりました。ACSL S では、論理ライブラリの番号のギャップは許可されず、使用可能な番号が再利用されます。

注:

ライブラリ構成から ACS を削除する前に、物理 ACS に関連付けられた論理ライブラリをすべて削除します。これにより、関連付けられたファイバチャネル接続が正しく削除されるようになります。

論理ライブラリを削除する前に、次の点に注意してください。

- その論理ライブラリをオフラインにしてください。
- すべての論理ドライブをオフラインにしてください。

これにより、ライブラリの削除時に、進行中であるクライアント操作は確実になくなります。論理ドライブまたはボリュームが使用中であることが見つかった場合、論理ライブラリは削除されません。

トラブルシューティング

このセクションでは、可能性のあるいくつかの問題シナリオについて説明し、それらの問題を診断して解決する方法に関するアイデアを提供します。

クライアントが割り当て済みの論理ライブラリを表示できない場合はどうしたらよいですか。

1. `acsss status` を実行することにより、ACSLS が実行中であることを確認します。

このユーティリティーは、ACSLS が一般に有効になっているかどうかを示します。

FC の問題については、`smce` サービスを調べます。このサービスは、論理ライブラリに SCSI メディアチェンジャーインタフェースを提供します。

2. ACSLS が実行中である場合は、クライアント接続が正しく定義されていることを確認します。
 - 実際に FC 経由で接続されていないターゲットとイニシエータのペアを割り当てている可能性があります。
 - 論理ライブラリを間違ったクライアント (間違ったターゲットとイニシエータのペア) に割り当てている可能性があります。
 - `getHba.sh` ユーティリティーは、どのイニシエータ (クライアント) が検出され、どれが特定のターゲットポートに接続されているかの識別に役立つ場合があります。

クライアントは正しく接続されているが、引き続き論理ライブラリが表示されない場合はどうしたらよいですか。

クライアントに強制的にターゲットデバイスを再検出させることが必要になる場合があります。実際のメカニズムは、クライアントシステムのオペレーティングシステムによって異なります。クライアントアプリケーションがデバイスを使用するには、その前に、クライアント OS がそのデバイスを表示する必要があります。

たとえば、Solaris クライアントでは、`cfgadm` および `devfsadm` コマンドを使用してデバイスのファイルシステムと構成を更新できます。

クライアントでの論理ライブラリの構成で問題が発生している場合はどうしたらよいですか。

クライアントのバックアップアプリケーションが、ACSL5 論理ライブラリをサポートされているライブラリタイプとして認識しない可能性があります。論理ライブラリは、ソフトウェアベンダーからの更新を必要としている可能性のある一意の照会文字列を返します。ライブラリとそのテープドライブを正しく構成するために、アプリケーション固有のマッピングファイルも必要になる可能性があります。

クライアントソフトウェアで、テープライブラリが SCSI ターゲット上で常に LUN 0 として表示されることを想定している場合があります。通常、クライアントのターゲットポート上の最初の論理ライブラリは実際に LUN 0 として表示されます。ただし、特定のターゲットポート上で複数のライブラリがクライアントに割り当てられている場合は、ライブラリが常に LUN 0 として表示されることを当てにすることはできません。これはクライアントの問題です。

FC ポートをターゲットモードで構成する方法

インストール中に、ターゲットポートとして使用する HBA を選択するよう求められます。この操作は、SCSI (FC) クライアントアプリケーションに 1 つ以上の論理ライブラリを提供する場合にのみ適用されます。

インストールのあと、`getHba.sh` ユーティリティを使用して、ファイバチャネルポートをターゲットモードで構成します。`getHBA.sh` ユーティリティは、新しい HBA がシステムに追加されたり、HBA ポートが並べ替えられたりした場合はいつでも実行できます。このユーティリティは、ACSL5 論理ライブラリへのクライアントアクセスポイントを明らかにするために、イニシエータモードからターゲットモードに変更する適切な HBA ポートを識別します。

1 つ以上の論理ライブラリを SCSI ダイレクト用に構成する場合は、クライアントを接続したあと、イニシエータモードからターゲットモードに変換されるべき HBA を選択します。選択する HBA がファブリックに接続され、SCSI クライアントイニシエータとして使用されるリモート HBA を認識できることが理想的です。

指定された HBA のターゲットモードの変更が設定されたあと、HBA の変換を完了するにはリブートが必要です。

ファイバチャネルポートをターゲットモードで構成するには、次の手順を実行します。

1. `root` としてログインします
2. `$ACS_HOME/install` ディレクトリに移動するか、またはそのディレクトリをパスに含めます。

3. `getHba.sh` を実行します

例:

```
# ./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.
```


付録I ライブラリのパーティション分割

SL8500 および SL3000 では、ライブラリのパーティション分割をサポートしています。

1 台の ACSLS サーバーで、同じライブラリ内の複数のパーティションを管理できます。

- SL8500 パーティション分割

- SL8500 コンプレックスのパーティション分割

SL8500 8.3 以降のファームウェアでは、ライブラリコンプレックス全体にわたるパーティション分割をサポートします。パススルーポートに接続された SL8500 のライブラリコンプレックスには、最大 16 個のパーティションを作成できます。

ACSL 8.3 以降のリリースでは、最大 16 個のパーティションをサポートします。ACSL 8.2 以前のリリースでは、最大 8 個のパーティションをサポートします。

- 拡張 SL8500 パーティション分割 (SL8500 7.0x ファームウェアで使用可能)

- › 拡張 SL8500 パーティション分割では、スタンドアロンの単一の SL8500 だけをパーティション分割できます。
- › 拡張 SL8500 パーティション分割は、最大 8 個の物理的なパーティションをサポートできます。
- › 拡張 SL8500 パーティション分割のレベルは、ドライブおよびセルアレイです。

- レガシー SL8500 パーティション分割

- › レガシー SL8500 パーティション分割では、スタンドアロンの単一の SL8500 だけをパーティション分割できます。
- › レガシー SL8500 パーティション分割は、最大 4 個の物理的なパーティションをサポートできます。
- › レガシー SL8500 パーティション分割のレベルはルール (LSM) であり、ルールには、ルール上のライセンス付与されたすべてのテープドライブとカートリッジが含まれます。
- › レガシー SL8500 パーティションには、1 から 4 個のルールを含めることができます。レガシー SL8500 パーティション内のルールは連続している必要があります。たとえば、パーティションには、ルール 2、3、および 4 を含めることができますが、ルール 2 と 4 だけにすることはできません。

パーティション分割の詳細は、SL8500 のドキュメントセットを参照してください。

- SL3000 パーティション分割
 - SL3000 では、最大 8 個の物理的なパーティションをサポートできます。
 - SL3000 はドライブおよびセルレベルまでパーティション分割できます。

詳細は、『SL3000 ユーザーズガイド』を参照してください。

SL8500 と SL3000 の両方のパーティション分割に共通した点

SL8500 ライブラリまたは SL3000 ライブラリをパーティション分割すると、次の機能が得られます。

- ACSLS は、オープンシステムのドライブおよびカートリッジを管理でき、HSC はライブラリを共有するメインフレームのドライブおよびカートリッジを管理できます。
- ACSLS の複数のインスタンスがライブラリにアクセスできます。
- ACSLS の 1 つのインスタンスで、同じライブラリ内の複数のパーティションを管理できます。
- 別々のパーティション内のデータカートリッジの保護と隔離を提供します。
- よりレベルの高いデータ編成。
- ユーザー効率の向上。

注意:

パーティション分割されたライブラリは、ACSLs に対して構成または再構成されたあとで、監査 (*audit*) する必要があります。ライブラリは、ACSLs が監査するときに、ACSLs にアクセスできない可能性のあるセルの場所を報告します。アクセスできないセルの場所には、CAP、ドライブ、およびオペレータパネルが取り付けられている場所、ロボットがアクセスできないセルの場所、アクティブ化されていないセルの場所、およびこのパーティション内にはないセルの場所が含まれます。

注意:

あるパーティションから別のパーティションにセルが割り当て直されている場合は、それらのセル内のカートリッジは孤立し、以前に含まれていたパーティションではアクセスできなくなります。ほかのパーティションを管理するホストが、カートリッジ上のデータを上書きする可能性があります。

注:

SL500 をパーティション分割できますが、ACSLs はパーティション分割した SL500 をサポートしません。

パーティション分割のガイドライン

ライブラリをパーティション分割するときに、従う必要のあるいくつかの手順があります。それらの機能を次に示します。

1. 新しい構成を計画します。
2. ライブラリをパーティション分割または再パーティション分割するための停止のスケジュールを設定します。

ライブラリを再パーティション分割すると、その変更によってパーティションに影響がない場合でも、そのライブラリはすべてのパーティションを管理するすべてのホストへの通信を切断します。

注:

ACSAPI クライアントは、ACS に割り当てることができないのと同じように、ライブラリの物理的なパーティションに割り当てることができません。

3. 保守担当者のライセンスを用意し、SL8500 または SL3000 でのパーティション分割を有効にします。
4. アクセスできない場所から、まだ ACSLS で管理されるパーティションに残っている場所に、カートリッジを移動します。
5. SL コンソールを使用して、パーティションを作成します。

手順については、SL コンソールのヘルプを参照してください。

6. 「[ライブラリのパーティション分割またはパーティション ID の変更](#)」の説明に従って、ACSLs を使用してパーティション分割を有効にします。
7. ライブラリがパーティション分割されていることを指定し、パーティション番号を入力することによって、`acsss_config` を使用してパーティション分割されたライブラリに対して ACSLS を構成する必要があります。

新しい構成を計画する

あらかじめパーティション分割された新しい構成を計画します。

- パフォーマンス向上のためにカートリッジとドライブを編成します。
- パーティションに対して挿入と取り出しを何度も行う場合、下部の 3 つのレールのうち少なくとも 1 つを SL8500 パーティションに含める必要があります。挿入と取り出しが少ないパーティションは、上部のレールに置くことができます。

SL3000 または SL8500 のパーティション分割時の混乱を最小限に抑える

SL8500 または SL3000 がパーティション分割されたり、SL8500 または SL3000 内のパーティションが変更されたりすると、データへのアクセスに影響があります。あるホストサーバーで管理されていたテープドライブとカートリッジは、別のホストで管理できます。

構成における変更

次の場合、構成での変更が生じます。

- ライブラリの構成が変更した場合
- ライブラリが再パーティション分割され、ACSLs で管理されるパーティションが変更された場合 (メッセージが表示されます)
- ライブラリの容量が変更され、ACSLs に利用できると報告されたセルが変更された場合
- LSM 構成または容量が変更された場合

構成が変更された可能性がある場合は、ACSLs イベントログおよびイベント通知メッセージで通知されます。

停止を最小限に抑える

次に示す手順に従えば、データへのアクセスを継続しながら、停止を最小限に抑えることができます。これらの手順では次の方法について説明します。

- パーティション分割されていない SL8500 または SL3000 を 2 つ以上のパーティションに分割する方法。
- すでにパーティション分割されている SL8500 または SL3000 内のパーティションを変更する方法。
- パーティション分割された SL8500 または SL3000 をマージして、パーティション分割されていない単一の ACS にする方法。

問題

LSM (SL8500 レール) がレガシー SL8500 パーティションから取り除かれた場合、または、セルおよびドライブが SL3000 または拡張 SL8500 パーティションから取り除かれた場合、最大の問題が生じます。

詳細は、「[パーティションからセルを取り外す前のカートリッジの移動](#)」の手順を参照してください。

ライブラリのパーティション分割またはパーティション ID の変更

次の手順では、パーティション分割されていないライブラリをパーティション分割する方法、またはパーティション分割された既存のライブラリでのパーティション ID を変更する方法について説明します。

1. SL8500 LSM や SL3000 または SL8500 のセルを ACS から取り除く前に、カートリッジを移動します。

ACSL S が新しいライブラリ構成でいずれかのパーティションを管理する場合、[ボリュームリストファイルの作成](#) または「[パーティションからセルを取り外す前のカートリッジの移動](#)」の手順に従って、別のパーティションに割り当てられる LSM からカートリッジを移動します。

2. 再パーティション分割する前にライブラリを休止し、ACS とポートをオフラインに変更 (*vary*) します。
3. SL コンソールを使用してライブラリをパーティション分割します。
4. 「*Apply*」を押すと、次のようになります。ライブラリが
 - ACSLS からの新しいリクエストを拒否します。
 - すべてのホスト接続を切断します。
 - パーティション分割の変更を適用します。
 - 新しいホスト接続を承認します。
5. パーティションの変更が有効になるように、ACSL S を再構成します。

ライブラリがパーティション分割されたのがはじめてだったため、またはライブラリが再パーティション分割されパーティション ID が変更したために、ACSL S はライブラリに再接続できません。再接続するには、次の手順に従います。

- a. ACSLS を停止します: *acsss disable*。
- b. *acsss_config* を実行し、ACS を構成して、オプション 8 を選択します。

ACS がパーティション分割されたライブラリであるかどうかを尋ねられたら、y (はい) と答えます。y (はい) と答えたあと、パーティション番号を入力 (*enter*) します。詳細は、「[CSI チューニング変数の設定](#)」を参照してください。

6. 新しい構成で ACSLS を起動します: *acsss enable*
7. ポートをオンラインに、ACS を診断モードに変更 (*vary*) します。
8. ACS を監査 (*audit*) します。
9. ACS をオンラインに変更 (*vary*) します。

ライブラリの再パーティション分割

次の手順では、パーティション分割された既存のライブラリを再パーティション分割する方法について説明します。

1. ACS から LSM またはセルを取り外す前に、カートリッジを移動します。

ACSL S が新しいライブラリ構成でいずれかのパーティションを管理する場合、[ボリュームリストファイルの作成](#) または「[パーティションからセルを取り外す前のカートリッジの移動](#)」の手順に従って、別のパーティションに割り当てられる LSM からカートリッジを移動します。

2. 再パーティション分割する前にライブラリを休止し、ACS とポートをオフラインに変更 (vary) します。

これにより、ACSL S はクライアントからの新しいリクエストを拒否しますが、ACS は引き続き動的な再構成に利用できます (再パーティション分割後)。

3. SL コンソールを使用して、ライブラリを再パーティション分割します。

続いて、「Apply」を押すと、次のようになります。ライブラリが

- ACSLS からの新しいリクエストを拒否します。
- すべてのホスト接続を切断します。
- パーティション分割の変更を適用します。
- 新しいホスト接続を承認します。

4. パーティションの変更が有効になるように、ACSL S を再構成します。

注:

特定のパーティションに変更がなされていない場合があります。変更は、ほかのパーティションにのみ影響している可能性があります。この場合は、ACSL S 構成を変更する必要はありません。

注:

ライブラリがパーティション分割されたのがはじめてではなかったため、またはライブラリが再パーティション分割されパーティション ID が変更されなかったために、ACSL S はライブラリに再接続できます。

次のいずれかを選択します。

- ACSLS が実行している間に動的構成 (`config acs acs_id`) を実行します。これにより、新しいライブラリ構成と一致するように ACSLS 構成が更新されます。この手順に従います。

- ポートをオンラインに変更 (*vary*) します。
- `config acs acs_id` を使用して ACS 構成を更新します。
- または、ACSL S を停止し、`acsss_config` (オプション 8) を実行して、ACSL S でのパーティション構成を更新します。この手順に従います。
 - ACSLS を停止します: `acsss disable`
 - `acsss_config` を使用して構成を更新します。
 - ACSLS を再起動します: `acsss enable`。
 - ポートをオンラインに変更 (*vary*) します。
- 5. ACS を診断モードに変更 (*vary*) します。
- 6. ACS を監査 (*audit*) します。
- 7. ACS をオンラインに変更 (*vary*) します。

パーティション分割されている ACS をパーティション分割されていない ACS に変更する

1. オプションで、ACSL S で管理されるパーティションからカートリッジを取り出します (*eject*)。

ACSL S がパーティション分割されていないライブラリを管理しない場合、ACSL S で管理されているパーティションからカートリッジを取り出すことができます。これにより、ACSL S が管理するほかの ACS にこれらを挿入できます。

複数のカートリッジの取り出しには、`ejecting.sh` ユーティリティを使用できます。

2. 再パーティション分割する前にライブラリを休止し、ACS とポートをオフラインに変更 (*vary*) します。
3. SL コンソールを使用して、ライブラリを再パーティション分割します。

続いて、「**Apply**」を押すと、次のようになります。ライブラリが

- ACSLS からの新しいリクエストを拒否します。
 - すべてのホスト接続を切断します。
 - パーティション分割の変更を適用します。
 - 新しいホスト接続を承認します。
4. ACS をパーティション分割されていないライブラリに変更するか、その構成からこの ACS (パーティション) を取り除くように ACSLS を再構成します。

パーティション分割されたライブラリからパーティション分割されていないライブラリに変更されたので、ACSL S はライブラリに再接続できません。再接続するには、次の手順に従います。

- a. ACSLS を停止します: `acsss disable`
- b. `acsss_config` を実行して、ACS を構成します。

ACS がパーティション分割されたライブラリであるかどうかを尋ねられたら、n (いいえ) と答えます。詳細は、「[CSI チューニング変数の設定](#)」を参照してください。

5. 新しい構成で ACSLS を起動します: `acsss enable`。
6. ポートと ACS をオンラインに変更 (`vary`) します。

ACS パーティション ID の表示

1 つまたは複数の ACS のパーティション ID を表示するには、次のいずれかのコマンドを使用します。

- `query lmu all for all ACSs`
- `query lmu acs_id for a single ACS`

出力例:

```
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 の動作

このセクションでは、CAP の動作について説明します。

パーティション分割されたライブラリ

警告:

専用であるか、共有されている 1 つの CAP が、すべての ACS に必要です。SL3000 内のすべての CAP はほかのパーティション専用に行けるため、これは SL3000 パーティションを構成するときに重要な制限になります。

パーティション分割されたライブラリでの CAP の専用化

SL3000 ライブラリでは、CAP を 1 つのパーティション専用を設定できます。CAP をパーティション専用にすると、ACSL5 はこれらの CAP を自動モードに設定できます。

パーティション分割されたライブラリでの CAP の共有

レガシーパーティション分割の SL8500 ライブラリでは、CAP は常にすべてのパーティションで共有されます。

パーティション分割された SL3000、または拡張パーティション分割の SL8500 では、ライブラリ CAP をパーティションで共有できます。このプロセスは次のとおりです。

1. ホストは、CAP を使用してカートリッジを挿入または取り出す場合、排他的に使用できるようにこの CAP を予約します。
2. ホストは、挿入または取り出しを完了して、CAP を空のまま、または閉じたままにしておく必要があります。

次のそれぞれで、挿入を終了する必要がある場合:

- ACSLS - キャンセルされた *enter* コマンドを発行する必要があります
- HSC - *drain* コマンドを発行する必要があります

3. これで CAP は、ライブラリ内のすべてのパーティションで使用できるようになります。

別のホストが CAP を必要とする場合

ホストが挿入または取り出しを完了しない場合、CAP にはまだ、影響を受けるパーティションのカートリッジが含まれている可能性があります。CAP はまだ、予約済みのステータスです。ほかのパーティションで CAP を使用する必要があると、このために問題が生じます。これを解決するには、ライブラリは予約を保持しているホストを識別し、そのホストからの挿入または取り出しを終了する必要があります。

まれに、ホストがクラッシュし再起動できないため、または必要とするコマンドを発行する物理的アクセスまたはセキュリティー権限が不足しているために、予約を保持しているホストにアクセスできないことがあります。このような場合は、SL コンソールを使用して CAP 予約をオーバーライドできます。

注意:

SL コンソールを使用した CAP 予約のオーバーライドは、最後の手段としてのみ実行してください。

共有された CAP の CAP 優先順位の割り当て

ACSLs は、同じライブラリ内の複数のパーティションを管理でき、これらのパーティションが同じ CAP を共有できます。監査は、重複し読み取り不能な `vol_id` を持つすべてのカートリッジを取り出します。同時に複数のパーティション (ACS) を監査する場合、監査は、同じ共有 CAP を使用してカートリッジを取り出そうとする可能性があります。最初の監査が CAP を予約し、ほかの監査はその CAP を予約できません。この場合、以降の監査は、カートリッジを取り出すことができません。

このような事態にならないようにするため、異なる共有 CAP が別々のパーティションでもっとも高い優先順位の CAP になるように、CAP 優先順位を設定します。たとえば、SL3000 内のパーティション 2 および 3 が CAP 5 および 6 を共有する場合、CAP 5 をパーティション 2 でもっとも優先順位の高い CAP として設定し、CAP 6 をパーティション 3 でもっとも優先順位の高い CAP として設定します。

CAP の予約

CAP を予約するときには、次の点を考慮してください。

- CAP が挿入または取り出し操作に使用されている場合、39 個のスロット (SL8500) および 26 個のスロット (SL3000) がすべて、その操作に予約されます。CAP はさらに分割することはできません。
- アドレス指定を行うため、CAP には場所 (ACS, LSM, CAP#) が必要です。LSM 番号は各ライブラリの 2 番目のルールに関連付けられます。例: CAP A には ACS#,1,0、CAP B には ACS#,1,1
- CAP の予約

CAP はパーティション間で共通のコンポーネントです。各ホストは、

- 挿入 (*enter*) または取り出し (*eject*) に使用するために CAP を予約します。
- 挿入または取り出しが終了したときに、CAP を解放します。これで別のホストが使用できるようになります。

CAP 予約の終了

1. ACSLS は、CAP が別のパーティションによって予約されていることを検出すると、その予約を保有しているパーティション ID とホスト ID を報告します。

予約された CAP を保有しているホストに接続して、挿入 (*enter*) または取り出し (*eject*) を完了するようにリクエストします。

これが行われると、空の CAP が予約解除され、使用可能になります。

2. CAP を保有しているホストがその CAP 予約を解放できない場合、*SL Console* (SLC) コマンドによって CAP 予約をオーバーライドします。
 - a. SLC コマンドは、特定のパーティション所有者からの予約が削除されることを知らせるホストへの警告を表示します。
 - b. SL コンソールは CAP に照会して、パーティションの所有者/依頼者/予約者を検出します。
 - c. SLC は、警告にパーティション名 (HLI1、HLI2、HLI3、HLI4、またはデフォルト) を表示します。
3. ライブラリは、予約を保有していたホストに CAP オーバーライドメッセージを送信します。

このホストは、進行中であった挿入 (*enter*) または取り出し (*eject*) タスクを終了します。

4. これでライブラリが CAP を保有します。

CAP は、別のホストで使用できるようにするため、その前に空にして閉じる必要があります。もし、

- CAP にカートリッジが含まれている場合、パーティションが CAP を使用できるように、その前に SL コンソールを使用してその CAP を空にする必要があります。
- CAP が開いている場合、空にして閉じる必要があります。ライブラリは続いてこれを監査 (*audit*) し、これが空であることを確認します。CAP は閉じられるまで、どのパーティションでも使用可能にすることはできません。
- CAP が空の場合、CAP はどのパーティションでもロックして予約できるようになります。

5. これで CAP は予約解除され、どのパーティションでも使用できるようになります。

特定のセルへのカートリッジの移動

SL3000 は、ドライブおよびセルレベルまでパーティション分割でき、拡張パーティション分割の SL8500 は、ドライブおよびセルアレイレベルにパーティション分割できます。あるパーティションから別のパーティションにセルが割り当て直されている場合は、これらのセル内のカートリッジは孤立し、以前に含まれていたパーティションではアクセスできなくなります。ほかのパーティションを管理するホストが、これらのカートリッジ上のデータを上書きする可能性があります。

パーティションの境界を変更するとき、およびライブラリを再パーティション分割する前に、カートリッジが孤立しないように、パーティションに残すセルにカートリッジを移動します。

SL3000 は単一の LSM であるため、既存の ACSLS の *move* コマンドは機能しません。カートリッジは、ライブラリの別の場所に移動され、同様にパーティションから取り除かれる別のセルに移動される可能性もあります。これに対処するため、*move* コマンドには、特定のセルにカートリッジを移動する機能が用意されています。

セルにカートリッジを移動する構文は、LSM に移動する構文と似ています。ただし次のように、LSM ID を指定する代わりにセル ID を指定します。

a=acs, l=lsm, p=panel, r=row, and c=column

move AAAAAA a, l, p, r, c (セル移動の場合)

move AAAAAA a, l (標準の LSM 移動の場合)

セル移動の例:

move EDU010 0, 1, 0, 5, 1

注:

display コマンドを使用して、利用可能な (空の) セルのリストを指定のパネルに表示できます:
*display cell a, l, p, *, * -status empty -f status*

付録J トラブルシューティング

この付録では、ACSL Sにおける問題のトラブルシューティングのためのツール、ヒント、および手法についてまとめています。トラブルシューティングリソースの範囲には、ログ、鍵監視ポイント、診断プローブが含まれます。

ACSL S イベントログ

ACSL S イベントログは、ライブラリの操作に問題が発生した場合に、有用な情報が得られる最初の場所です。このログには、ライブラリイベント、ステータスの変更、およびエラーに関する情報が含まれます。ACSL S 内のすべてのサブコンポーネントは、イベントロガーと呼ばれるプロセスにメッセージを送信することによって、`acsss_event.log` にイベントを報告します。標準のイベントログは、ACSL S がインストールされるときに自動的に作成され、`$ACS_HOME/log/acsss_event.log` のファイルに含まれます。ここで、`$ACS_HOME` は通常、`/export/home/ACSSS/` です。

記録されるイベントは次のとおりです。

- 重大なイベント

重大なイベントとは、ライブラリの管理に役立つ通常のイベントです。たとえば、イベントは、監査が開始または終了したとき、デバイスが状態を変更したとき、CAP が開かれたり閉じられたりしたときに記録されます。

- ライブラリのエラー

ライブラリのエラーは、ハードウェアおよびソフトウェアの致命的なエラーと致命的でないエラーの両方が記録されるイベントです。たとえば、LSM の障害、カートリッジの問題、データベースエラー、プロセスの障害、ライブラリ通信障害などがあります。

イベントログ内の各メッセージには、タイムスタンプ、メッセージを報告しているコンポーネントの名前、およびイベントの説明が含まれます。各メッセージの詳細な説明については、『ACSL S メッセージ』のマニュアルを参照してください。

ACSL S コンソールのウィンドウには、イベントログの実行中の最後の部分が表示されます。どのシェルウィンドウからでも同様の表示を生成できます。

1. ユーザー `acs` として、次のコマンドを実行します。

```
acs_tail $ACS_HOME/log/acs_event.log
```

2. イベントログ全体を表示するには、ログ内を移動したり、特定のエラーを検索したり、特定のイベントシーケンスをたどったりできる、`vi` などのテキストエディタを使用します。

イベントログの管理

ACSL S は、`acs_event.log` にメッセージを送信し続けます。

- このファイルがしきい値サイズ (デフォルトで 500K バイト) に達すると、ファイルは `event0.log` の名前に変更され、ログディレクトリに保存されます。そのあと、`acs_event.log` は新しいファイルとして存続します。
- `acs_event.log` が再度しきい値サイズに達すると、`event0.log` の名前が `event1.log` に変更され、`acs_event.log` の名前が `event0.log` に変更されます。
- このプロセスは、ログファイルが保持するように構成されている数に達するまで続行します。

デフォルトでは、9 つのイベントログファイルがログディレクトリ内に保持されます。それ以降にしきい値に達するごとに、もっとも古いファイルから削除され、残りのすべてのファイルの名前が順次変更されます。

`acs_config`、オプション 2 を使用して、`acs_event.log` の最大サイズと保持するログファイル数を構成できます。[「CSI チューニング変数の設定」](#)を参照してください。

`greplog` を使用したイベントログの検索

診断ツール、`greplog` を使用すると、あらゆるイベントログファイルに対してキーワード検索を実行できます。`greplog` は、UNIX の `grep` ユーティリティと使用法が非常によく似ており、指定されたキーワード式に関連した完全なログメッセージを返します。これを使用して、メッセージの日付とタイムスタンプ、メッセージ番号、およびその式を含んでいるすべてのメッセージに関連する関数テキストを表示できます。

形式

```
greplog [-iv] pattern file_1 file_2 ... feline
```

オプション

- `-i` は、検索パターン式で大文字と小文字の区別を無視するように、`greplog` に指示します。

- `-v` は、式を含むすべてのメッセージを除外しログファイル内のすべてのエントリを表示するように、`greplog` に指示します。例外はパターン式に一致するエントリです。

パターン: パターンは、使用される検索条件です。

```
file_1 file_2 ... file_n
```

`greplog` ツールは、ファイルリスト内の複数のファイルパラメータおよびワイルドカード式を受け入れます。

例

- イベントシーケンス内のすべての出現を表示するには、シーケンス番号を使用します。

```
greplog 1392 acsss_event.log
```

- ボリューム `CART89` に関するすべてのメッセージをイベントログで検索するには:

```
greplog CART89 acsss_event.log
```

- イベントログのすべてのアーカイブされたコピーを調べて、テープマウントに関するメッセージを検索するには:

```
greplog -i mount event*.log
```

追加のログ

`log/acsss_event.log` には、ACSL5 実行プロセスのあらゆる側面に関するすべてのメッセージが含まれます。ただし、バックアップおよび復元ユーティリティーやインストールユーティリティーなど、外部ユーティリティーに関するステータス情報を含んだ追加ファイルが、ログディレクトリにあります。

- `acsss.pid` - 現在実行している `acsss_daemon` のプロセス ID が格納されます。
- `acsss_config.log` - 各ライブラリ構成のサマリーが含まれます。
- `acsss_config_event.log` - `acsss_config` ルーチンによって書き込まれたイベントメッセージが含まれます。
- `bdb_event.log` - データベースバックアップユーティリティー、`bdb.acsss` によって書き込まれたイベントメッセージが含まれます。
- `cron_event.log` - `cron` ユーティリティーによって書き込まれたメッセージが含まれます。cron スケジュールを表示するには、`crontab -l` コマンドを実行します。

- *acsls_start.log* - *acsls* サービスが関与する起動または停止メッセージが含まれます。
- *di_trace.log* - データベースのインタフェースに関するトレース情報が含まれます。
- *ejectingLogs* - 過去 10 日間の *ejecting.sh* 操作からのサマリー情報が含まれるディレクトリ。
- *install.log* - インストールスクリプト、*install.sh* を実行している場合に書き込まれたイベントメッセージが含まれます。
- *ipc_trace.log* - ACSLS プロセス間通信に関するトレース情報が含まれます。
- *rdb_event.log* - データベース復元ユーティリティー、*rdb.acsss* によって書き込まれたイベントメッセージが含まれます。
- *timed_bkup.sh.log* - 自動データベースバックアップユーティリティーに関連するイベントメッセージが含まれます。

システム上で有効にしている特定のトレースに応じて、その他のトレースログがログディレクトリに置かれている場合があります。このログには次のものがあります。

- *acsss_stats.log* - ボリューム統計トレースは、*acsss_config* によって有効になります。
- *acsss_trace.log* - クライアントとサーバー間のトレースは、ソフトウェアサポート担当者のリクエストで有効になります。
- *acslh.log* - ホストと LMU 間のトレースは、ソフトウェアサポート担当者のリクエストで有効になります。
- *scsilh.log*、*mchangerX.log*、*scsipkt.log* - これらはすべて、SCSI 接続ライブラリへの SCSI 通信のトレースを含み、ソフトウェアサポートのリクエストで有効になります。

トレースログの管理

ソフトウェアサポートのリクエストで有効になるトレースログは、非常に急速に増大する可能性があります。いっぱいディスクの問題を軽減するために、これらのログをモニターおよび管理する必要があります。

自動ログ管理およびアーカイブサービスを実行するユーティリティー *monitor.sh* が用意されています。その構文は次のとおりです。

monitor.sh <ログの名前>

特定のログをモニタリングするためにこのユーティリティーが有効になっている場合、1M バイトのサイズ (デフォルト) までログを増大でき、そのあと、*gzip* を使用してログを圧縮し、タイ

ムスタンプ付きの名前の圧縮ログファイルを `ACSSS/log/log_archives` サブディレクトリに配置します。トレースが有効なままであれば、この操作は続行されます。

Java コンポーネントのログ

ACSLs GUI や論理ライブラリソフトウェアコンポーネントなど、ACSLs 内の Java コンポーネントによって多数のログが管理されています。これらのログは `$ACS_HOME/log/sslm` ディレクトリにあります。

WebLogic のインストール手順は `weblogic.log` に記録されます。WebLogic および ACSLS GUI の操作は、`AcsIsDomain.log` と `AdminServer.log` に記録されます。

Web ベースの GUI でのユーザーアクティビティの監査トレールは、`guiAccess.log` にあります。

Java コンポーネントとレガシー ACSLS コンポーネント間のトランザクションは `surrogate_trace.0.log` に記録されます。

Java クライアントコンポーネントと ACSLS サーバー間の IPC パケットは `acslm_ipc_trace.0.log` 内でトレースされます。

ACSLs GUI で発生したエラーは、`gui_trace.0.log` に記録されます。

SMCE と SCSI (ファイバ) クライアント間の低レベルの通信は、`smce_trace.0.log` に記録されます。

これらのログは `$ACS_HOME/log/sslm` ディレクトリにあります。

鍵監視ポイント

ACSLs のさまざまな側面のステータスを検証できる多数のユーティリティがあります。

- `psacs` - すべての ACSLS 実行プロセスのサマリーを示します。これは、ACSLs が実行中かどうかをもっとも適切に示します。通常出力では、すべて共通の親プロセスの子である別々のプロセスが 12 以上表示されます。
- `acsss status` - acsdb データベースサービスが実行されているかどうかを確認します
- ACSLS リリースおよび保守レベルを表示するには:
 - Solaris の場合:

```
pkginfo -l STKacsls
```
 - Linux の場合:

```
rpm -q ACSLS
```

- Solaris または Linux の場合:

```
in_get_version
```

ACSLS の起動に関する問題の診断

- *acsss_event.log* を調べます。
- *acsls_start.log* を調べます。
- 問題を説明するメッセージがないか、*acsss_event.log* の末尾を調べます。
- メッセージの説明と、これらを解決するために行える対処法については、『ACSLS メッセージ』ガイドを参照してください。
- *acsss l-status* を使用して、ACSLS サービスのステータスを表示します。

acsss l-status を使用して、ACSLS サービスのステータスサマリーを表示します。サービスごとに、*logfile* エントリは、ACSLS が起動できなかった状況について説明した詳細メッセージを含む可能性のあるログデータを示します。

- ACSLS が起動中にタイムアウトします
- Solaris では、構成に基づいて計算された ACSLS の起動時タイムアウト期間を表示するために、*acsss timeout* を使用します。

ライブラリの接続テスト

ACSLS には、ライブラリへの良好な物理的接続を検証するユーティリティーが用意されています。どのツールを選択するかは、アクティビティーのコンテキストによってもっとも適切に決まります。

testports

このユーティリティーは、StorageTek ACSLS に対して構成されている各ライブラリへの接続をテストします。また、これはもっとも簡単に使用でき、もっとも便利です。このテストは目立たず、通常のライブラリ操作に影響を与えません。*testports* は StorageTek ACSLS データベースを使用してライブラリのポート名とライブラリタイプを判別するため、*testports* が機能するように、ライブラリはすでに StorageTek ACSLS に対して構成されている必要があります。

- TCP/IP ライブラリの場合、*testports* は、接続を検証し、ライブラリがオンラインになっており StorageTek ACSLS で使用されているかどうかを検証します。

- SCSI およびシリアル接続したライブラリの場合、「acs」および「port」は、*testports* がテスト接続を開くために、オフラインになっている必要があります。

このユーティリティーを実行するためのコマンドの構文は次のとおりです。

```
testports
```

ライブラリの互換性レベルまたはマイクロコードレベルを表示します。

testlmutcp

このユーティリティーは、ネットワーク接続ライブラリに TCP/IP パケットを送信します。

ライブラリの接続をテストするには、コマンド行で、ライブラリのホスト名または IP アドレスを含めます。

```
testlmutcp <ip_address> または
```

```
testlmutcp <hostname>
```

ライブラリが ACSLS に対してオンラインになっているときに接続をテストするには、50002 から 50016 の未使用のソケット番号を指定します。次に例を示します。

```
testlmutcp <ip_address>:50002
```

正常な応答には、接続されたライブラリの互換性レベルが含まれます。

testlmu

このユーティリティーは、ACSL S とレガシー StorageTek シリアル接続ライブラリ間の接続のテストに使用できます。このユーティリティーを実行するには、シリアルポートデバイスノードへの devlink パスを送信します。

```
testlmu /dev/term/0
```

testlmu がシリアル接続を開けるように、ACSL S に対してライブラリをオフラインにする必要があります。

pinglmu.sh

このユーティリティーを使用すると、ライブラリが ACSLS に対してオンラインになっているときに、ACSL S とシリアル接続ライブラリ間の通信を検証できます。正常な応答には、ライブラリの互換性レベルが含まれます。

probescsi.sh

このユーティリティーは、ACSL S サーバーと SCSI またはファイバ接続ライブラリ間の接続を実行します。このユーティリティーを実行するには、mchanger デバイスへの devlink パスを指定します。その構文は次のとおりです。

```
probescsi.sh /dev/mchangerX
```

ここで、*X* は、テストされるライブラリの特定の mchanger インスタンスです。

probescsi が SCSI 接続を開けるように、ライブラリを ACSLS に対してオフラインにする必要があります。正常な応答には、接続されたライブラリのマイクロコードレベルが含まれます。

probeFibre.sh

このユーティリティーは、ACSL S サーバーから到達可能なすべてのファイバ接続ライブラリを検出します。その構文は次のとおりです。

```
probeFibre.sh
```

正常な応答には、各ファイバ接続ライブラリのモデル番号が、そのターゲット、LUN ID、および World Wide Port Name (WWPN) とともに表示されます。

-v オプションを使用して、ホストバスアダプタのモデル番号を表示することもできます。

```
probeFibre.sh -v
```

showDevs.sh

このユーティリティーは、mchanger リンクが作成されているすべての mchanger デバイスの詳細を表示します。

- *showDevs.sh*

ライブラリのモデル、リビジョンレベル、および接続されている各 mchanger ライブラリの容量を表示します。

- *showDevs.sh -w*

このオプションには、各ライブラリの WWPN も含まれます。

- *showDevs.sh -s*

このオプションには、各ライブラリのシリアル番号も含まれます。

クライアントの接続テスト

クライアントアプリケーションは、RPC (リモートプロシージャコール) プロトコルを使用して TCP/IP 上で ACSLS と通信します。クライアントシステムが ACSLS と通信できない場合は、`rpcinfo` を使用して、ACSLS がクライアントマシンから到達可能かどうかをテストできます。

1. ACSLS サーバーから、ACSLS が実行していることを検証します。

```
psacs
```

2. ACSLS サーバーから、RPC デーモンが実行していることを検証します。

```
ps -ef | grep rpc
```

3. ACSLS サーバーから、TCP および IDP に対してプログラム番号 300031 が登録されていることを検証します。

```
rpcinfo | grep 300031
```

このプログラム番号は、ACSLS が実行しており、ACSLS が RPC に登録していることを確認します。

4. クライアントマシンから、またはネットワーク上のすべての UNIX マシンから、`rpcinfo` を使用して、ACSLS サーバー上のプログラム番号 300031 とパケットを交換します。

プログラム番号とともに ACSLS サーバーの IP アドレスを指定します。

```
rpcinfo -t <ip address> 300031
```

通信交換が成功した場合、`rpcinfo` ユーティリティは、次のメッセージを表示します。

```
program 300031 version 1 ready and waiting
```

```
program 300031 version 2 ready and waiting
```

これは、ACSLS がネットワーク全体のクライアント接続に使用できることを確認します。

ブリッジドライブで接続したファイバライブラリ内の CAP がロックされる

ブリッジドライブを介して接続されるファイバ接続ライブラリ内の CAP が、別の ACSLS インスタンスがライブラリの管理を引き継ぐときに、ロックされる場合があります。詳細とこの問題

の解決方法は、SL150 の付録で「[取り出し中に開かない CAP \(メールスロット\)](#)」を参照してください。

Oracle Support 用の診断情報の収集

Oracle Support では、サービスコールの一部として診断ログのセット全体および分析用のほかの診断情報を送信するよう依頼することがあります。このすべてのデータは単一のコマンドで収集できます。

get_diags

このユーティリティですべての情報が収集されると、データを電子メールで送信するか、手動転送を使用可能にするかを尋ねるプロンプトが表示されます。

データを直接 ACSLS マシンから電子メールで送信することにした場合は、ACSLS マシンとインターネット間で電子メール通信が可能であることを確認します。企業によっては、ターゲットマシンとの電子メールの送受信がファイアウォールによってブロックされることがあります。この場合は、情報を社内の自分宛に電子メール送信してから、Oracle Support に診断データを転送してください。

または、情報を手動で転送することも選択できます。*get_diags* ユーティリティで、転送待ち機中の tar パッケージが見つかる場所がアドバイスされます。通常、診断データ用のステージング領域は `/export/backup/diag/acsss` にあります。

ACSLS と SELinux (Security-Enhanced Linux)

SELinux は、Oracle Linux ではデフォルトで有効になっています。SELinux は、標準 Unix レベルのアクセス制御以上に、ユーザーロールおよびイミディエイトコンテキストドメインに応じて、システムリソースへのアクセスを強制します。SELinux の強制が有効になっている場合、独自の PostgreSQL データベースにアクセスする ACSLS の機能は、このようなアクセスに対してロールとコンテキストを確立する特別なポリシーがなければ妨げられます。

ACSLS の SELinux ポリシーモジュールのアンインストール

ACSLS をインストールすると、3 つの SELinux ポリシーモジュール (*allowPostgr*, *acsdb*, *acsdb1*) がカーネルにロードされます。これらのモジュールでは、SELinux の強制がアクティブなときに独自のデータベースおよびその他のシステムリソースにアクセスするために ACSLS で必要となる定義および強制例外が提供されます。これらのモジュールがインストールされていると、SELinux の強制を無効にすることなく、通常の ACSLS 操作 (*bdb.acsss*, *rdb.acsss*, *db_export.sh*, *db_import.sh* などのデータベース操作を含む) を実行できます。

迅速なソフトウェアアップグレードのために、ACSLs によってロードされた SELinux ポリシーモジュールは、ACSLs パッケージをアンインストールするときに自動的に削除されません。それらを手動で削除するには、ACSLs モジュールのリストを取得します。

```
# semodule -l | grep acsdb
```

```
# semodule -l | grep allowPostgr
```

モジュールごとに、次の方法でモジュールを削除します。

```
# semodule -r <module name>
```

SELinux の強制を管理する

ACSLs のインストール後、従来のファイルアクセス許可設定が有効であると思われるのに、システムから「Permission denied」と返されるようなアクセス関連の問題が発生した場合、そのアクセス拒否の原因が SELinux にある可能性があります。

SELinux の強制が有効になっているかどうかを確認するには、`sestatus` コマンドを実行します。

```
# sestatus
SELinux status:   enabled
Current mode:     enforcing
```

`setenforce` コマンドを使用して一時的に SELinux の強制を無効にできます。

```
# setenforce Permissive
```

SELinux の強制が許可モードであれば、障害の発生したリソースへのアクセスを復元できるかどうかを確認できます。許可モードにあるが強制モードにはない承認されたユーザーが、必要なリソースを利用できる場合、SELinux ポリシーの更新が必要であることを示しています。

SELinux のセキュリティーを永続的に (ブートにわたって) に無効にするには:

1. `/etc/selinux/config` のファイルを編集します
2. `SELINUX=enforcing` を `SELINUX=permissive` に変更します

SELinux の強制をふたたび有効にするには、`root` が `sysadm_r` ロールを持っている必要があります。

```
# newrole -r sysadm_r
```

```
# setenforce enforcing
```

SELinux が明らかに制限の原因であったことを検証し終わったら、SELinux の監査ログを調べることによって、必要なリソースへのアクセスを許可しなかった実際のルールを確認できます。

```
# vi /var/log/audit/audit.log
```

audit.log は、SE の強制が成功または失敗したそれぞれのアクセス試行に関するサマリーを提供します。失敗したイベントを探す必要があります。ACSLs の場合、特にユーザーの *acsss* および *acsdb* に関連したイベントを調べます。

どのファイルまたはディレクトリに関連付けられた SELinux コンテキスト属性でも表示できます。

```
# ls -Z <file name>
```

secon のコマンドを使用して、所定のプロセスのコンテキスト属性または現在のシェルのコンテキスト属性を表示できます。*chcon* コマンドを使用して、ファイルまたはディレクトリのコンテキスト属性を変更できます。これらの操作については、マニュアルページを参照してください。

audit.log で見つかった失敗した操作に応じてポリシーモジュールを作成できます。

```
# cd /var/log/audit
# audit2allow -a -M <ModuleName>
```

これは、SELinux によって記録された障害を評価し、ポリシーモジュールファイル *<ModuleName>.pp* を作成します。このファイルは、ブロックされていた操作を許可するために、Linux カーネルにロードできます。

```
# semodule -i <ModuleName>.pp
```

audit2allow は、*audit.log* で識別されるすべての制限を許可するポリシーを作成するため、*audit.log* には特に許可する操作しか含まれていないように確認することをお勧めします。元の *audit.log* を保存して、新しいログを作成できます。

```
# mv audit.log audit1.log
# touch audit.log
```

ポリシーモジュールを作成する前に、取得する操作を続行します。

SELinux に関する詳細は、マニュアルページを参照してください。

```
# man selinux
```

GUI が動作していることの検証

`checkGui.sh` ユーティリティーは一般的な要素を確認して、GUI が動作しているかどうかを評価します。GUI が動作していない場合、このユーティリティーは、問題の可能性の高い原因をユーザーに示すことができます。

このユーティリティーは次の点を確認します。

- Weblogic がシステム上で有効になっていますか。
- 疑似プロセスまたは古いプロセスが Weblogic 操作を妨げている可能性がありますか。
- SlimGUI アプリケーションが正常に配備されていますか。
- Weblogic および GUI は、localhost に送信される HTTP リクエストに応答できますか。
- Weblogic は、ホストのインターネットアドレスに送信される HTTP リクエストに応答できますか。
- ファイアウォールサービスがサーバー上で有効になっていますか。その場合、Weblogic ポート 7001 および 7002 への受信リクエストを受け入れるポリシーはありますか。

Linux システムでは、`iptables` と呼ばれるファイアウォールがデフォルトで有効になっている場合があります。`iptables` を完全に無効にすることも、ポート 7001 および 7002 への受信トラフィックを受け入れるポリシーを追加することもできます。

1. これらのポートを (`root` として) 有効にするには、ファイル `/etc/sysconfig/iptables` を編集します。次の 2 行を追加します。

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

これらのルールは絶対に、調べられる前に受信パケットを照合する別のルールのあとに挿入しないようにしてください。たとえば、`REJECT all` を行うルールのあとの `iptables` チェーンの末尾に付加しないでください。

`iptables` コマンドを使用してこれらのルールを追加する場合:

- テーブルを一覧表示 (`iptables -L`) または出力 (`iptables -S`) します。

- ルールを追加します。

前のルールによって新しいルールが入力を照合できなくなる場合があるため、チェーンの末尾にルールを追加するだけでは (*iptables -A*) 目的の結果が生成されない可能性があります。

たとえば、*ruIenum* によってルールを挿入します (*iptables -I*)。

- 変更後、テーブルを一覧表示 (*iptables -L*) または出力 (*iptables -S*) して、既存のルールによってポート 7001 および 7002 の新しいルールが調べられないようになっていないことを確認します。

これにより、新しいルールは受信パケットを照合できるようになります。

checkGui.sh ユーティリティは、ポート 7001 および 7002 で入力を受け入れるルールが存在するかどうかを確認します。これは、これらのルールが右側の *iptables* チェーン内にあることも、新しいルールが実際に処理されることも検証しません。言い換えれば、*checkGui.sh* は、新しいルールの検査を妨げる前のルールが存在しないことを検証しません。

2. *iptables* を再起動します。

```
service iptables restart
```

Solaris での同等のサービスは *ipfilter* であり、通常はデフォルトで有効になっていません。

GUI のトラブルシューティングのヒント

次の表では、GUI のトラブルシューティングのヒントについて説明します。

表J.1 GUI のトラブルシューティングのヒント

問題	解決方法
ブラウザに <code>https://<hostname></code> と入力しましたが、応答ページには「Unable to connect」と表示されます。	正しい URL は <code>https://hostname.domain:7002/SlimGUI/faces/Slim.jsp</code> です
ACSL S GUI ページが不完全です。一部のフレームが不完全であったり、セクション全体が見つかりません。	ブラウザのリフレッシュボタンをクリックします。

問題	解決方法
有効なことがわかっているユーザー ID とパスワードが、Java WebLogic で拒否されます。ログインできません	<p>ローカルの ACSLS 管理者に問い合わせてください。管理者である場合、<code>userAdmin.sh</code> ユーティリティーを使用して、ユーザーの一覧表示、ユーザーの追加、またはユーザーパスワードの変更を行なってください。</p> <p>まだユーザーにログインの問題がある場合は、システムに十分なメモリーがあるかを確認してから、<code>userAdmin.sh</code> のオプション 5 を使用して ACSLS GUI を再起動してください。または、<code>svcadm disable weblogic</code> および <code>svcadm enable weblogic</code> を使用して Weblogic を再起動できます。</p>
Java エラースタックトレースが、GUI の 1 つ以上のウィンドウに表示されます。	<p>ブラウザのリフレッシュボタンを押します。</p> <p>問題が解決しない場合は、<code>acsss status</code> を使用して ACSLS サービスが実行されていることを検証します。</p> <p>サービスが実行していない場合は、<code>acsss enable</code> を使用してサービスを起動します。</p> <p>ACSL S サービスが実行している場合、<code>userAdmin.sh</code> を使用して GUI を再起動します。または、<code>svcadm disable weblogic</code> および <code>svcadm enable weblogic</code> を使用して Weblogic を再起動できます。</p> <p>システムに <code>root</code> アクセスできない場合は、<code>acsss shutdown</code> ですべてのサービスをシャットダウンしてから、<code>acsss enable</code> で再起動できます。GUI がこのプロセスによって再起動されません。</p>
インデックスツリーフレームで「論理ライブラリ」の選択肢が見つかりません。	<p>最初に論理ライブラリを作成する必要があります。「Configuration and Administration」->「Logical Library Configuration」->「Create Logical Library」を選択します。</p>
「Tape Library Operations」または「Tape Libraries & Drives」の下の「Volumes」ページにボリュームが一覧表示されていません。	<p>これは、初期監査がライブラリに対して行われていないことを示します。「Tape Library Operations」->「Audit」を選択します。</p>
「Logical Libraries」の下の「Volumes」ページにボリュームが一覧表示されていません。	<p>これは、ボリュームがボリュームライブラリにまだ割り当てられていないことを示しています。「Logical Library Configuration」->「Assign Volumes」を選択します。</p>

問題	解決方法
GUI の応答時間が遅くなっています。	GUI マストヘッドの「Preferences」ボタンの下にある「Alert Update Interval」を増やします。
GUI ユーザーの追加、GUI ユーザーのパスワードの変更、または <i>acs1s_admin</i> パスワードの設定を行う必要があります。	「userAdmin.sh」 を参照してください。 このユーティリティーを使用すると、ユーザーの追加とユーザーのパスワードの変更を行え、 <i>acs1s_admin</i> パスワードをリセットする方法がわかります。
ブラウザがセキュリティ証明書を必要としています。	「HTTPS 用の自己割り当てデジタル証明書の構成」 を参照してください。

付録K ACSLS クライアントアプリケーションのインストール時の問題

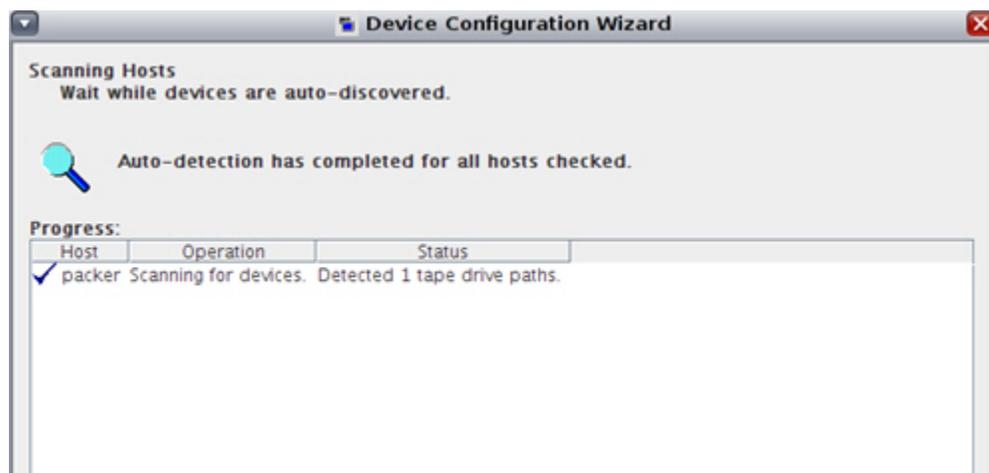
この付録では、ACSLs と通信するクライアントアプリケーションをインストールしているときの既知の問題と、それらの正常な解決方法について要約します。

Solaris 11 での ACSAPI クライアントのインストール

Solaris 11 には、クライアントアプリケーションが RPC を使用して ACSLS サーバーと通信するために必要なすべてのパッケージが含まれているわけではありません。

問題:

この例では、クライアントは ACSLS ライブラリを検出できませんでした。



解決方法:

1. /etc/hosts ファイルを変更し、ACSAPI クライアントサーバーをリポートします。

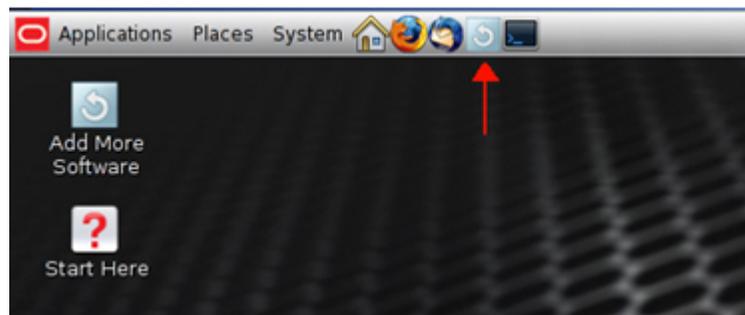
Solaris 11 でのデフォルト: (ループバック IP アドレスしかありません)

```
:::1 <client_hostname> localhost
127.0.0.1 <client_hostname> localhost loghost
```

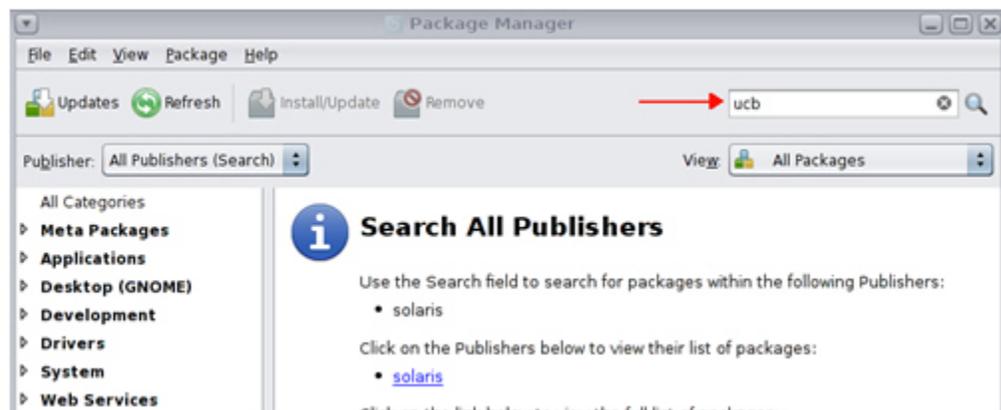
Solaris 11 での更新された `/etc/hosts` ファイル: (ACSAPI クライアントの実際の IP アドレスが含まれています)

```
:::1 localhost
127.0.0.1 localhost
<client_IP_addr> <client_hostname> localhost
```

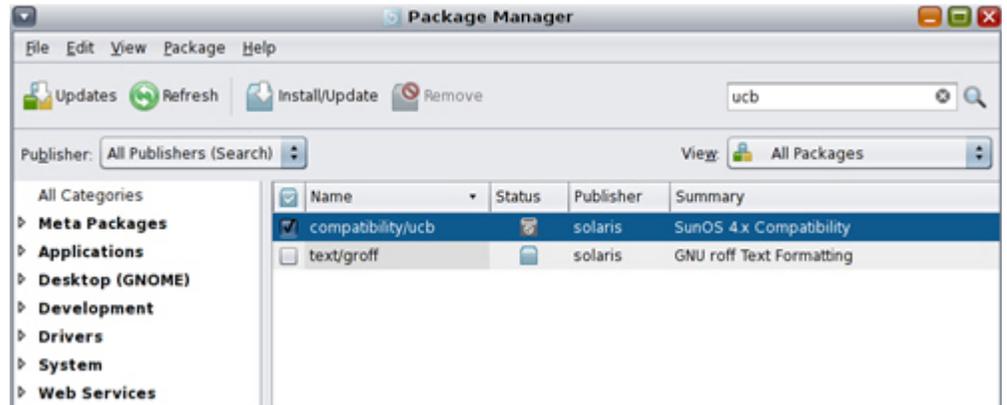
2. パッケージマネージャーを使用して、ACSL通信に必要な `compatibility/ucb` パッケージをインストールします。
 - a. 「パッケージマネージャー」アイコンをクリックします。



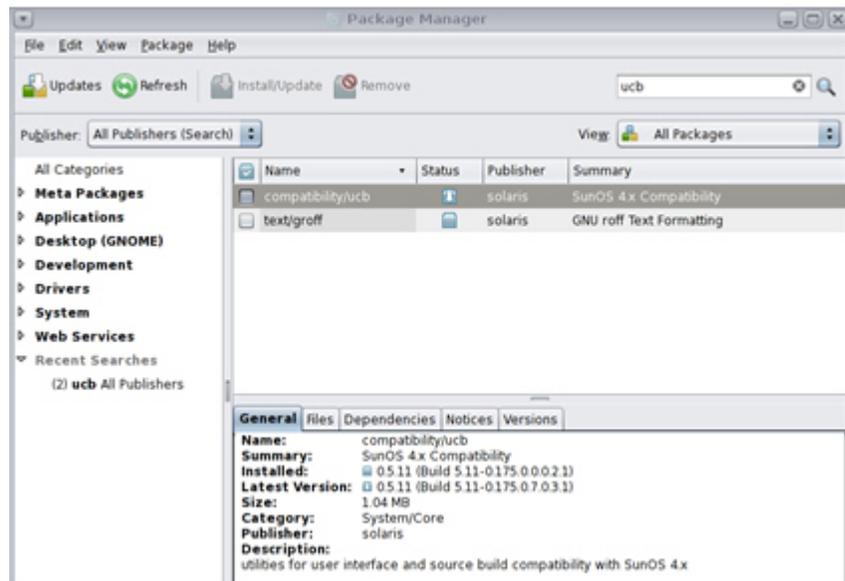
- b. 「ucb」を検索します。



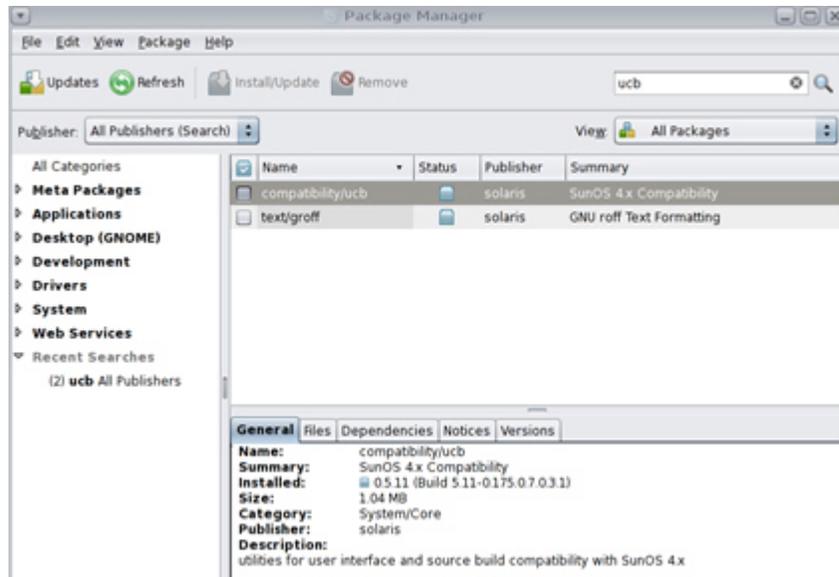
- c. 「`compatibility/ucb`」パッケージにチェックマークを付け、「インストール/更新」をクリックします。



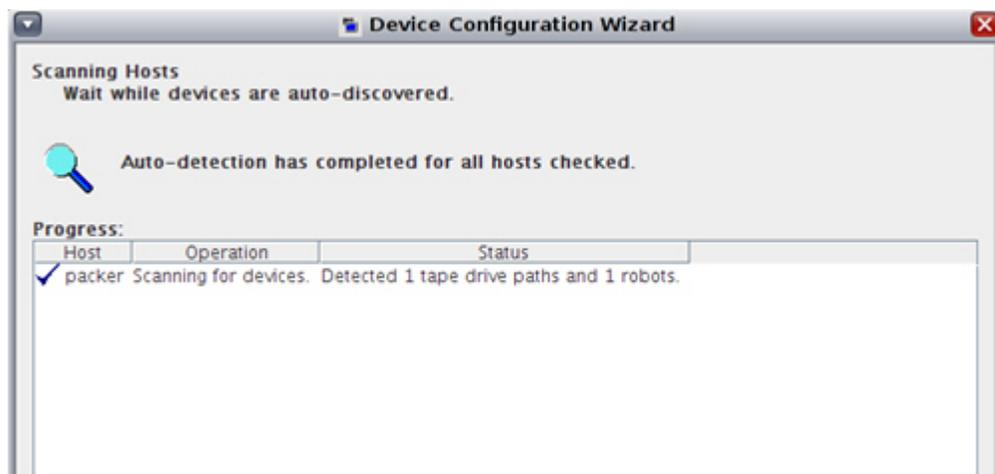
d. 最新バージョンがインストールされたことを確認します。



e. 新しいバージョンが使用可能な場合は、再度「インストール/更新」を実行します。



3. クライアントを再構成します。



付録L ライブラリのパフォーマンス

この章では、ライブラリストレージモジュール (LSM) 間のパススルーアクティビティー中のライブラリのパフォーマンスを最大化する方法について説明します。ACSLs は、いくつかの方法でパススルーアクティビティーを最小限に抑えるように機能します。これらの ACSLS 機能を使用してパススルーアクティビティーを最小限に抑えることにより、パフォーマンスを向上させることができます。それらの機能を次に示します。

- 十分な並行マウントおよびマウント解除リクエストの送信
- 複数の SL8500 への接続
- LSM 間のパススルーアクティビティーの最小化
- マウント解除中のカートリッジのフロート
- ライブラリへのカートリッジの挿入
- ライブラリからのカートリッジの取り出し
- ライブラリ内の空のストレージセルの保持
- ACSAPI リクエストと ACSLS コマンドを使用したパススルーアクティビティーの最小化
- これらの ACSAPI リクエストおよび ACSLS コマンドの使用

十分な並行マウントおよびマウント解除リクエストの送信

一部のクライアントアプリケーションは、1 回につき 1 つのリクエストのみを処理する単純な SCSI メディアチェンジャーライブラリ用に設計されています。ただし、ACSLs と SL8500 と SL3000 ライブラリは、多数のリクエストを同時に処理するように設計されています。各ライブラリのキューには少なくとも 40 から 50 の並行リクエストが含まれており、ACSLs キューは基本的に無制限です。

クライアントは、ACSLs に数百の並行マウントおよびマウント解除リクエストを送信できます。ライブラリのパフォーマンスを最大化するには、ライブラリ内のロボットやその他のすべてをビジー状態に維持するために、クライアントは ACSLS に十分な数のリクエストを送信する必要があります。

クライアントが ACSLS に限られた数の並行リクエストしか送信できない場合は、ACSLs に複数のクライアントを接続できます。たとえば、特定のクライアントアプリケーションが ACSLS

に 1 回につき 1 つのマウントまたはマウント解除リクエストしか送信できない場合は、ACSL S に SL8500 あたり 12 から 16 のクライアントアプリケーションを接続することによって、ACSL S に SL8500 内のすべてのロボットをビジー状態に維持するための十分な並行マウントおよびマウント解除リクエストを送信できます。

次に、ライブラリのロボットをビジー状態に維持するために並行リクエストがいくつ必要になる可能性があるかを理解するための詳細を示します。

- ライブラリは、ドライブがロードされて準備が完了するまで、ACSL S にマウント応答を返しません。このようにして、ACSL S は、そのマウントが成功したかどうかを認識します。
- 約 3000 のストレージセルを備えた小規模な SL8500 では、ロボットがカートリッジをドライブに移動するまでに約 10-15 秒かかります。
- ただし、LTO ドライブがカートリッジをロードして準備を完了するには約 19 秒かかります。
- レール上の 1 つのロボットをビジー状態に維持するのにさえ、そのレールに対する少なくとも 3 つの重複したマウントリクエストが必要です。

これにより、カートリッジを受信する最初のドライブがロードして準備を完了している間に、ロボットがほかのマウントを処理できるようになります。レールあたり 2 つのロボットの場合は、レールあたり少なくとも 4 つの並行マウントまたはマウント解除が必要になる可能性があります。

- SL8500 には 4 つのレールがあるため、SL8500 内のすべてのロボットをビジー状態に維持するには、SL8500 あたり 12 から 16 の並行マウントおよびマウント解除リクエストが必要です。

これらの並行リクエストのすべてが 1 つの ACSAPI クライアントから来ることも、それぞれが 1 回につき 1 つのリクエストを発行し、次のリクエストを発行する前にその応答を待機するだけの 12 から 16 の異なるクライアントから来ることもあります。

複数の SL8500 への接続

ACSL S サーバーが、パススルーポート経由で接続された SL8500 の文字列を管理している場合は、その文字列内の各 SL8500 に接続します。これにより、ACSL S はステータスに応答するか、またはカートリッジの移動を開始したライブラリにリクエストをルーティングすることによって、ライブラリのパフォーマンスを最適化できます。

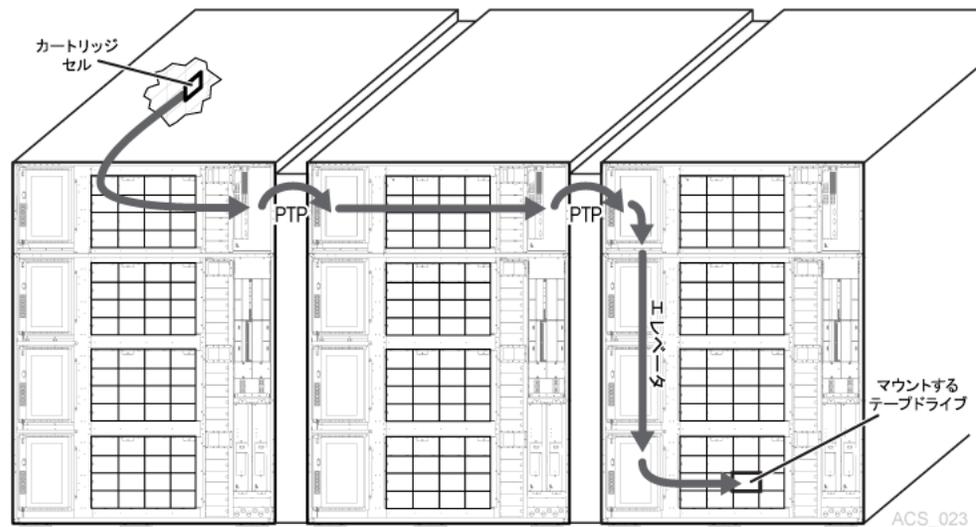
文字列内の複数の SL8500 への接続の詳細は、「[マルチ TCP/IP サポート](#)」を参照してください。

LSM 間のパススルーアクティビティの最小化

テープライブラリのパフォーマンスを向上させるには、あるライブラリから別のライブラリへのパススルーアクティビティを最小限に抑えることが重要です。カートリッジとドライブが接続済みのライブラリ内にある場合は、ドライブとカートリッジの両方が同じ LSM (ライブラリ内のレール) 内にあるときにテープマウントが発生することが常に望まれます。マウントに必要なパススルーが少ないほどパフォーマンスが向上します。

図L.1「LSM 間のパススルーアクティビティの最小化」は、カートリッジとドライブが別の LSM 内にあるときに必要になる可能性のあるパススルーを示しています。

図L.1 LSM 間のパススルーアクティビティの最小化



詳細は、「エレベータおよび PTP アクティビティを最小限に抑える」、「ワークロードをサポートするためのテープドライブの構成」、および「カートリッジの場所の管理」を参照してください。

マウント解除中のカートリッジのフロート

カートリッジがマウント解除されると、ACSL はそのカートリッジの古いホームセルがドライブとは別の LSM 内にある場合は常に新しいホームセルを割り当てることによって、LSM 間のパススルーアクティビティを回避しようとします。これは ACSL のデフォルトの動作です。この機能を無効にするには、LSM の「拡張ストア」機能を使用します。マウント解除されているカートリッジは、LSM 間の移動を回避するために新しいホームセルに「フロート」されます。

LSM 内の空のストレージセルの保持

すべてのストレージセルがいっぱいになると、ACSL S は、マウント解除でカートリッジを LSM に「フロート」できません。同様に、いっぱいになった LSM にカートリッジが挿入された場合は、そのカートリッジを空のセルを含む LSM にパススルーする必要があります。

パフォーマンスを最適化するには、StreamLine アクセスのためにライブラリ内に保持する必要のないカートリッジを識別し、それらのカートリッジをライブラリから取り出し (eject) ます。また、いっぱいになった LSM から十分な空のセルを含むほかの LSM にカートリッジを移動して、すべての LSM に空のストレージセルが含まれるようにすることもできます。「[アクティブな LSM からの、もっとも長い間アクセスされていないカートリッジの移動](#)」を参照してください。

注:

ライブラリ内のカートリッジの管理は、そのライブラリがマウントリクエストでビジー状態になっていないときに実行するようにしてください。

マウントおよびマウント解除のタイムアウト

ACSL S とクライアントアプリケーションの間でマウントおよびマウント解除のタイムアウトを調整する必要があります。クライアントのタイムアウトは、SL8500 のレールとライブラリ間のパススルーを必要とするマウントおよびマウント解除が完了できるように、十分に長くする必要があります。また、リクエストがパススルーポートを待機する可能性がある場合は特に、複数の並行リクエストが完了できるように十分に長くすることも必要です。

ACSL S はまた、ライブラリまたはテープドライブの一時的な停止中はリクエストを自動的にキューに入れ、そのライブラリまたはドライブが使用可能になったら再試行します。これにより、冗長電子デバイスの切り替え中、ライブラリまたはドライブがリブートされたとき、またはライブラリのアクセスドアが開かれているときのリクエストの失敗が回避されます。

マウントおよびマウント解除をキューに入れるデフォルトの時間 (`MOUNT_RETRY_TIME_LIMIT`) は 20 分です。顧客は、これを 5 から 80 分までの任意の時間に設定できます。ACSL S はまた、ライブラリがキューに入れられたリクエストを処理できるかどうかを `MOUNT_RETRY_DELAY` ごとにチェックします。これらはどちらも動的変数、つまり、ACSL S の実行中に変更できる変数であるため、その変更はただちに有効になります。

ACSAPI クライアントのタイムアウトを、大規模なライブラリコンプレックス内のリクエストが完了できるように調整してください。また、`MOUNT_RETRY_TIME_LIMIT` を ACSAPI クライアントのマウントおよびマウント解除のタイムアウトと調整してください。

ライブラリへのカートリッジの挿入

カートリッジがライブラリに挿入されると、各カートリッジには、CAP にもっとも近い LSM 内のホームセルが割り当てられます。CAP を含む LSM に空のセルがある場合、ホームセルは、この LSM 内に割り当てられます。CAP を含む LSM がいっぱいである場合、ホームセルは、空のセルを含むもっとも近い LSM 内に割り当てられます。

パフォーマンスを最適化するには、カートリッジを、そのカートリッジと互換性があるテープドライブを含む LSM に挿入 (*enter*) します。たとえば、9940 カートリッジは T9840B テープドライブのみを含む LSM ではなく、T9940B テープドライブを含む LSM に挿入 (*enter*) します。そうでない場合は、9940 カートリッジを T9940B テープドライブを含む LSM にパススルーする必要があります。

ライブラリからのカートリッジの取り出し

カートリッジがライブラリから取り出される場合は、そのカートリッジを指定された CAP に移動する必要があります。

パフォーマンスを最適化するには、カートリッジを、そのカートリッジが配置されている LSM 内の CAP に取り出し (*eject*) ます。カートリッジが複数の LSM 内に配置されている場合は、大部分のカートリッジに近い CAP を選択します。

ACSAPI リクエストと ACSLS コマンドを使用したパススルーアクティビティーの最小化

これらの ACSAPI リクエストおよび ACSLS *cmd_proc* コマンドは、LSM 間の不必要なパススルーアクティビティーを回避するのに役立ちます。

次は、クライアントから ACSLS への ACSAPI リクエストと、*cmd_proc* を使用して入力された ACSLS コマンドの両方に適用されます。「リクエスト」という用語は、ACSAPI リクエストと *cmd_proc* コマンドの両方に使用されます。

特定のカートリッジのマウント

これは、データカートリッジか、または選択した特定のスクラッチ (空の) カートリッジのどちらかのマウントに適用されます。次のコマンドを使用します。

- `query mount`
- `mount`

query mount

- 指定されたカートリッジ (`vol_id`) について、そのカートリッジのメディアタイプと互換性があるドライブのリスト (そのカートリッジに近い順) を返します。(もっとも近い LSM 内の互換性のあるドライブが最初に表示され、以降も同様です。)
- ACSLS 7.3 以降のリリースでは、カートリッジから同じパススルー距離にあるドライブは、もっとも以前に使用された順になります。

例: カートリッジの LSM 内の互換性のあるドライブのうち、カートリッジがマウント解除されてからもっとも長い時間が経過したものが最初で、次に長い時間が経過したドライブが 2 番目になります。以降も同様です。

コマンド: `query mount vol_id`

mount

特定のカートリッジをマウントし、リストの先頭にできるだけ近い使用可能な (使用中でない) ドライブを選択します。

ACSLS によって選択されたスクラッチカートリッジのマウント

次のセクションでは、ACSLS によって選択されたスクラッチカートリッジのマウントについて説明します。

query mount scratch (query mount * と呼ばれます)

指定されたスクラッチプールについて、指定されたスクラッチプール内のカートリッジのメディアタイプと互換性があるすべてのドライブのリストが返されます。特定のメディアタイプを指定すると、ドライブを指定したメディアタイプと互換性があるものだけに制限できます。

返されるドライブのリストは、もっとも密度の高いスクラッチプールにもっとも近いドライブが最初に表示されるように順序付けられます。

コマンド:

- ACSAPI
- `cmd_proc`
- `query mount scratch`

```
query mount * pool_id [media media_type]
```

特定のメディアタイプか、または *ALL_MEDIA_TYPE* (*cmd_proc* コマンド内の *media **) のどちらかを指定できます。*ANY_MEDIA_TYPE* はサポートされていません。

注:

ALL_MEDIA_TYPE が指定されている場合は、もっとも密度の高いスクラッチプールを決定するために、ドライブと互換性があるメディアを含むカートリッジが選択されます。

mount scratch (query mount * とも呼ばれます)

指定されたドライブについて、スクラッチ (空の) カートリッジを選択してマウントします。オプションで、指定されたスクラッチプールのカートリッジまたは指定されたメディアタイプのカートリッジ、あるいはその両方を選択します。ACSLs は、ドライブにもっとも近い LSM 内の互換性のあるスクラッチカートリッジを選択します。すべてのカートリッジ間で使用をローテーションするために、選択された LSM 内のもっとも以前のアクセス日付を持つ互換性のあるカートリッジが選択されます。

コマンド:

- ACSAPI

cmd_proc

mount scratch

*mount * drive_id [pool_id] [media media_type]*

- ACSAPI リクエストの場合は、*drive_id*、*pool_id*、および *media_type* を指定する必要があります。(*media_type* には、特定のメディアタイプ、*ALL_MEDIA_TYPE*、*ANY_MEDIA_TYPE* のいずれかを指定できます。)
- *cmd_procmount ** コマンドでは、*pool_id* が指定されていない場合、デフォルトで共通プール (プール 0) になります。

特定のメディアタイプを識別できます。*media ** が指定されている場合は、*ANY_MEDIA_TYPE* が選択されます。メディアが指定されていない場合は、*ALL_MEDIA_TYPE* が選択されます。

ANY_MEDIA_TYPE と *ALL_MEDIA_TYPE* の特殊な *media_type* 値は、次のように処理されます。

- *ALL_MEDIA_TYPE* が指定されている場合は、ドライブと互換性があるメディアタイプを持つカートリッジが選択されます。(これは *media_compatibility* ファイルに基づきます。)

- `ANY_MEDIA_TYPE` が指定されている場合は、`scratch_preferences` ファイルによって、ドライブにマウントするメディアの優先リストが識別されます。

これらの ACSAPI リクエストおよび ACSLS コマンドの使用

このセクションでは、テープライブラリのパフォーマンスを向上させるための ACSAPI リクエストおよびコマンドを使用する方法について説明します。

次の説明で、「リクエスト」という用語は、ACSAPI リクエストと `cmd_proc` コマンドの両方に使用されます。

特定のカートリッジをマウントする場合 (ここで、`vol_id` は既知です):

- `mount` リクエストの前に `query mount` リクエストを発行します。
- 最初の「使用可能な」ドライブを選択し、`mount` リクエストでこのドライブを指定します。

特定のドライブにスクラッチカートリッジをマウントする場合は、次の 2 つのオプションがあります。

- 特定のドライブにもっとも近いスクラッチカートリッジを選択するには:

ドライブ、およびオプションでスクラッチプールを指定した `mount scratch` リクエストを使用します。

ACSAPI リクエストの場合は、次のいずれかを指定します。

- `ALL_MEDIA_TYPE` (互換性のあるメディアが選択されます)
- `ANY_MEDIA_TYPE` (スクラッチ優先順位リストを使用します)。

`cmd_proc` コマンドの場合は、次のいずれかを選択します。

- メディアタイプを指定しません (互換性のあるメディアが選択されます)
- `media *` を指定します (スクラッチ優先順位リストを使用します)。
- ドライブのリストから特定のドライブを選択したあと、もっとも近いスクラッチカートリッジを選択するには:

ドライブを選択してから、`scratch tape:` を選択します。これにより、そのドライブが確実にスクラッチカートリッジに近くなります。

- 目的のスクラッチプール内の大部分のスクラッチメディアにもっとも近い使用可能なドライブを識別するには、`query mount scratch` リクエストを入力します。
- 選択されたドライブにスクラッチカートリッジをマウントするには、`mount scratch` リクエストを入力します。オプションで、スクラッチプールを指定します。

ACSAPI リクエストの場合は、次のいずれかを指定します。

- *ALL_MEDIA_TYPE* (互換性のあるメディアが選択されます)、または
- *ANY_MEDIA_TYPE* (スクラッチ優先順位リストを使用します)。

cmd_proc コマンドの場合は、次のいずれかを指定します。

- メディアタイプを指定しません (互換性のあるメディアが選択されます)、または
- *media ** を指定します (スクラッチ優先順位リストを使用します)。

付録M ファイアウォールセキュリティーオプション

ファイアウォールセキュリティーオプションは、クライアントソフトウェアがファイアウォールをまたがってリクエストを発行している間、ACSLS をそのファイアウォールの背後で動作できるようにします。

ファイアウォールセキュリティーは ACSLS クライアントにも提供され、これらのクライアントをそれぞれ独自のファイアウォールの背後で動作できるようにします。これは、Oracle によって、その独立系ソフトウェアベンダー (ISV) パートナーでも使用可能になります。特定のクライアントごとの最新のステータスを確認するには、使用しているクライアントソフトウェアコンポーネントの ISV にお問い合わせください。

ファイアウォールの背後での ACSLS の実行

このファイアウォールセキュリティーソリューションには、次の利点があります。

- ACSLS をファイアウォールの背後で動作できるようにします (つまり、ACSLS はファイアウォールのセキュアな側にあり、クライアントは反対側にあります)。
- ACSLS クライアントを独自のファイアウォールの背後で動作できるようにします (つまり、クライアントはセキュアな側にあり、ACSLS はファイアウォールの反対側にあります)。

注:

ISV は、提供された変更を、自身のクライアント側のソフトウェアコンポーネント内に実装する必要があります。

- 現在の ACSLS クライアント実装との互換性が保持されるため、これらのクライアントを引き続きファイアウォールソリューションで ACSLS とともに実行できます。
- 現在の ACSAPI/クライアントの機能とパフォーマンスが保持されます。これには、ファイアウォール以外の環境で使用可能なすべての機能が含まれます。

完全なソリューションには、上の最初の 2 つの機能の組み合わせが含まれます。これにより、ACSLS と ACSLS クライアントはそれぞれ独自のファイアウォール (つまり、ACSLS とクライアントの間に 2 つのファイアウォール) の背後で動作しながらも、ファイアウォール以外の環境内と同じ通信パフォーマンスを維持できます。

セキュリティ領域への対処

ACSL S は、次のセキュリティの問題に対処しています。

RPC

ACSL S 内での RPC の使用は、ファイアウォール環境内で動作させようとしている一部のサイトでは問題になります。現在のインストール済みクライアントベースとの互換性を保持すると、RPC を ACSLS から完全に削除する機能が除外されます。

ACSL S のファイアウォールセキュリティ機能は、次に示すような、RPC に固有の問題に対処しています。

- 外部の (信頼できない) 第三者が、無制限の範囲のポート (1024-65535) にわたって、信頼できるホストへの接続を開始できるようにする必要性。
- 既知のポート 111 上で実行されている portmap (または *rpcbind*) デーモン経由での、プラットフォーム上の使用可能なサービスのマッピングの公開。

セキュリティ

ファイアウォールソリューションでは、基本的なセキュリティは、セキュアでない側から信頼できる (セキュアな) 側へのアクセスを制限することから得られます。いずれの場合も、通信を実行してデータ交換を許可するには、一部の制限され、かつ制御されたアクセスを許可する必要があります。目標は、適切に定義され、かつ制限された一連のエントリポイント内のそのデータ交換を許可することにより、それらのアクセスポイントとそれに対応する通信を制御できるようにすることです。この目標は、このソリューションによって満たされます。

注:

IPv4 ベースのエッジファイアウォールを使用している場合は、インターネットホストがいずれかの IPv6-over-IPv4 トンネル化トラフィックを使用して内部ホストに到達することを防止するために、そのファイアウォールを、すべてのアウトバウンド IPv4 プロトコル 41 パケットおよび UDP ポート 3544 パケットを削除するように構成してください。

通信コンポーネント

ACSL S とクライアントの通信は、クライアントプラットフォームと ACSLS プラットフォームの間のネットワーク通信の処理を 2 つのネットワークインタフェースコンポーネントに依存しています。ACSL S のクライアントまたはプロキシサーバーとして機能しているソフトウェアは、ACSL S プラットフォームや既存のクライアントとの互換性を維持するために、これらの 2 つのコンポーネントのうちの 1 つを実装します。クライアントプラットフォーム上に存在するコ

コンポーネントは SSI と呼ばれ、ACSL S プラットフォーム上に存在するコンポーネントは CSI と呼ばれます。すべての変更を (ACSL S プラットフォームなどの) 1 つの側で実装することが望ましい一方で、クライアントの互換性を維持し、すべてのファイアウォールセキュリティー機能を提供するには、各側で対応する変更を行なって利点を得ることが必要です。長所として、各側が独立して機能を実装し、自身の側でファイアウォールセキュリティーの利点を実現できる点があります (たとえば、ACSL S の変更によって、ACSL S プラットフォームがセキュリティー保護されたファイアウォールの背後で動作できます)。

ファイアウォールセキュリティーオプションの利点

このセクションでは、ファイアウォールセキュリティーオプションの利点について説明します。

ACSL S サーバー側

サーバー側のコンポーネントのみの変更 (このファイアウォールセキュリティーソリューション内で提供されます) では、次の利点があります。

- ACSL S 通信の受信接続を、すべての登録済みプログラム番号について 1 つの TCP ポートに制限します (ACSL S CSI の登録済みプログラム番号は 2 つ存在し、その両方が 1 つのポートによって処理されます)。
- ユーザーがそのポートの識別情報を指定し、自身のファイアウォールを対応する方法で構成できるようにします。
- ユーザーが UDP ポートへの ACSL S 通信をオフにできるようにします。
- ユーザーが ACSL S サーバーによるクライアント側のポートマッパー* (UDP/TCP ポート 111) への通信をすべて無効にできるようにします。このポートマッパーは、クライアント側のコードとの互換性を保持するために、引き続きクライアントプラットフォーム上で実行されている必要があります。ただし、サーバーによって開始されたネットワーク通信には使用されないため、クライアントのファイアウォールをそれに対するアクセスを禁止するように構成できます。
- ACSL S サーバー側からクライアントへの送信接続は、現在のパフォーマンスを保持するために使用されるサーバー側のポートに対して無制限です。これは、セキュリティーコミュニティによって広く受け入れられている習慣に従っています。

ACSL S サーバーポートの制限

このファイアウォールソリューションでは、外部の第三者がネットワーク通信を開始するときに使用できる受信ポートの数が制限されます。これらのポートは、ACSL S 受信リクエストのために顧客が指定した 1 つのポートに加え、可能性のある 2 つのポートマッパーポート (TCP および UDP ポート 111) の 1 つまたは 3 つに制限されます。

注:

ACSLS サーバーのポートマッパーへのクライアントアクセスを禁止する (そのため、UDP および TCP ポート 111 へのアクセスを禁止する) には、クライアントソフトウェアコンポーネントを変更する必要があります。下のクライアント側の説明を参照してください。

ソリューションのサーバー側 (上記) は、ACSLS 内に完全に実装されています。

クライアント側 (CSC)

CSC に対して行われる変更によって、クライアント側のプラットフォームに上で説明したのと同じ制限が適用されます。これにより、CSC に、独自のセキュアなファイアウォールの背後に存在できるという同じ機能が提供されます。このソリューションには、次の利点があります。

- CSC への通信 (応答) の受信接続を、登録済みプログラム番号ごとに 1 つの TCP ポートに制限します。ACSLS SSI の登録済みプログラム番号は 1 つ存在します。
- エンドユーザーは TCP ポートの識別情報を指定し、自身のファイアウォールを同様に構成できます。
- UDP ポートへのクライアント側の通信をオフにします。
- クライアントによる ACSLS サーバーのポートマッパー (UDP/TCP ポート 111) への通信をすべて無効にします。このポートマッパーは、ACSLS コードとの互換性を保持するために、引き続き ACSLS プラットフォーム上で実行されたままにする必要があります。ただし、クライアントのネットワーク通信はこのポートマッパー経由で開始されません。そのため、ACSLS サーバーのファイアウォールをそれに対するアクセスを禁止するように構成できます。
- クライアント側から ACSLS サーバーへの送信接続は、現在のパフォーマンスを保持するために使用されるクライアント側のポートに対して無制限です。

クライアントポートの制限

このソリューションでは、外部の第三者がネットワーク通信を開始するときに使用できる受信ポートの数が制限されます。これらのポートは、クライアント受信応答のために顧客が指定した 1 つのポートと、可能性のある 2 つのポートマッパーポート (TCP および UDP ポート 111) の 1 つまたは 3 つに制限されます。

注:

クライアントのポートマッパーへの ACSLS サーバーのアクセスを禁止する (そのため、UDP および TCP ポート 111 へのアクセスを禁止する) には、**ACSLS** サーバーソフトウェアコンポーネントを変更する必要があります (上の ACSLS サーバー側の説明を参照してください)。

このソリューションの実装は、次の 2 段階から成ります。

- Oracle StorageTek が、CSC 開発者ツールキット 2.3 (以降) のソースコードに対して必要なコード変更を行なっています。
- このセキュリティを自身のクライアントプラットフォームに提供することを希望する ACSLS のクライアントは、これらの変更を自身のクライアント側の SSI コードに統合し、製品を再構築して、自身の Client System Component (CSC) を ACSLS に対して再度認証する必要があります。

メリット

ソリューションのクライアント側の部分とサーバー側の部分が独立しています。そのため、2 つの側のどちらか一方のみがファイアウォールの背後に存在する場合は、もう一方の側に関して、その側でしかソフトウェア変更を実装する必要がありません。さらに、1 つの側だけの変更によって、すでに存在する現在のすべてのクライアントおよびサーバー実装との互換性や、CSI/SSI インタフェースを使用するほかのソフトウェアコンポーネントとの互換性が維持されます。

注:

これには、現在の Oracle StorageTek 製品との互換性が含まれます。

このソリューションは、クライアント/サーバー通信の現在のパフォーマンスには影響を与えません。

ファイアウォールセキュリティ機能をオンにして変数を設定する

ACSLS サーバーをファイアウォールの背後で、さらにオプションで ACSLS クライアントをファイアウォールの背後で実行するには、ACSLS サーバーとクライアントシステムの両方で、それらがファイアウォールの背後に存在するときに変数を設定します。これらの変数を使用すると、受信通信を 1 つのポートに制限したり、オプションでポートマッパーを無効にしたりできます。

ACSLS 変数

CSI_TCP_RPCSERVICE- TCP プロトコルを使用した RPC の CSI サポートを有効にします。

- **機能:** CSI を TCP RPC サーバーとして動作できるようにします。いずれかのクライアントが TCP 経由で ACSLS と通信する場合は、このオプションを TRUE に設定します。
- **有効なオプション:** TRUE または FALSE (デフォルトは TRUE)
 - TRUE によって、クライアントの CSI への TCP アクセスが有効になります。

- FALSE によって、クライアントの CSI への TCP アクセスが無効になります。
- その他の詳細: このオプションを有効にするには、ACSLs 製品を再起動する必要があります。

CSI_UDP_RPCSERVICE- UDP プロトコルを使用した RPC の CSI サポートを有効にします。

- 機能: このオプションを選択すると、CSI が UDP RPC サーバーとして動作できるようになります。いずれかのクライアントが UDP 経由で ACSLS と通信する場合は、このオプションを TRUE に設定します。
- 有効なオプション: TRUE または FALSE (FALSE を推奨)
 - TRUE によって、クライアントの CSI への UDP アクセスが有効になります。
 - FALSE によって、クライアントの CSI への UDP アクセスが無効になります。
- その他の詳細: このオプションを有効にするには、ACSLs 製品を再起動する必要があります。ファイアウォールセキュリティの CSI は、TCP 通信でのみサポートされます。ACSLs サーバーとともにファイアウォールの内部でレガシークライアントアプリケーションを使用していないかぎり、*CSI_UDP_RPCSERVICE* は FALSE に設定します。

CSI_USE_PORTMAPPER- ポートマッパーを有効にします。

- 機能: このオプションを選択すると、クライアントに応答を送信できない場合、CSI はポートマッパーを調べようとします。クライアント上のポートマッパーへのアクセスを許可しない場合は、このオプションを ALWAYS に設定します。
- 有効なオプション: ALWAYS、NEVER、または IF_DUAL_LAN_NOT_ENABLED
 - ALWAYS は、CSI がクライアントに応答を送信できない場合、ポートマッパーが常に調べられることを示します。
 - NEVER は、CSI がクライアントに応答を送信できない場合、ポートマッパーが決して調べられないことを示します。このオプションは、クライアントがポートマッパーをサポートしていない場合を選択するようにしてください。
 - IF_DUAL_LAN_NOT_ENABLED は、デュアル LAN のサポートが有効になっていない場合にのみポートマッパーが調べられるように指定します。デュアル LAN のサポートが有効になっている場合は、クライアントがポートマッパーをサポートしていないと見なされません。IF_DUAL_LAN_NOT_ENABLED は、下位互換性のためのデフォルトです。
- その他の詳細: このオプションを有効にするには、ACSLs 製品を再起動する必要があります。

CSI_FIREWALL_SECURE - CSI をファイアウォールの背後で (ユーザー定義のインバウンドポートで) 使用できるようにします。

- **機能:** このオプションは、ACSL S サーバーをセキュリティ保護されたファイアウォールの背後で動作できるようにします。ACSL S によって使用されるインバウンドポートを指定し、アクセスを 1 つのポートに制限します。そのポートを除くすべてのポート上の受信 ACSL S トラフィックを拒否するようにファイアウォールを構成します。これにより、そのポートのみが、ACSL S との通信を開始することを希望する外部のクライアントで使用されるために公開されるようになります。

ポートアクセスを制限するには、次の手順を完了して、指定したポートのためのセキュアなファイアウォールを設定します。

- このオプションを TRUE に設定します。
- 受信 ACSL S リクエストが許可される、CSI によって使用されるポートを指定します。(CSI_INET_PORT で指定されます。)
- 固定ポート RPC をサポートしていない一部のレガシークライアントアプリケーションの場合は、そのクライアントからのポートマッパー照会リクエストをサポートするために、ファイアウォールで UDP/TCP ポート 111 を開くことが必要になる可能性があります。
- ファイアウォールセキュリティの CSI は、TCP 通信でのみサポートされます。

ACSL S サーバーとともにファイアウォールの内部でレガシークライアントアプリケーションを使用していないかぎり、CSI_UDP_RPCSERVICE は FALSE に設定します。

- 前に指定したポート上で外部クライアントが通信を開始したり、受信したりできるように、ACSL S サーバーが背後に存在するファイアウォールを構成します。開かれたファイアウォールポートを最小限に抑えるために、クライアントアプリケーション上の固定ポートを同じポートで設定することを忘れないでください。
- ACSL S を再起動します。
- **有効なオプション:** TRUE または FALSE (デフォルトは TRUE)
 - TRUE – ACSL S サーバーへのアクセスを、クライアントからの受信リクエストのために 1 つのポートだけを使用するように制限します。
 - FALSE – ACSL S サーバーへのクライアントリクエストに使用されるポートを制限しません。
- **その他の詳細:** このオプションを有効にするには、ACSL S 製品を再起動する必要があります。

CSI_INET_PORT - 受信 ACSL S リクエストを受信するために CSI によって使用されるポート番号。

- **機能:** このオプションは、クライアントからの受信 TCP リクエストのために CSI によって使用されるポートを指定します。

- 有効なオプション: 1024 から 65535 までの数値 (50003 を除く)。(デフォルトは 30031)
- その他の詳細: このオプションを有効にするには、ACSL S 製品を再起動する必要があります。この変数は、ファイアウォールセキュリティの CSI が `CSI_FIREWALL_SECURE` で有効になっており、TRUE に設定されている場合にのみ使用されます。

ACSL S 変数の表示および設定

ACSL S の静的および動的変数を表示したり設定したりするには、ACSL S `acsss_config` ユーティリティまたは `dv_config` ユーティリティを使用します。

- `dv_config -d`

ACSL S のすべての静的および動的変数とその設定を表示します。

- `dv_config -p <variable_name> -u`

変数を変更するよう求めるプロンプトを表示し、それが動的変数である場合は ACSL S グローバル共有メモリーを更新します。変数の完全な説明を表示するには、このプロンプトで「?」と入力します。完全な説明が表示されたあと、再度プロンプトが表示されます。

ACSAPI クライアントシステム変数

ACSL S クライアントシステムでファイアウォールセキュリティ動作を有効にするには、そのクライアントシステムが ACSL S CSC ツールキット 2.3 (以降) またはそれ以降のリリースで構築されている必要があります。

ACSL S クライアント上のファイアウォールセキュリティ動作を有効にするための環境変数は 4 つあります。これらの変数を特定の値に設定する必要があります。SSI プロセスを起動するには、これらの各変数を設定し、SSI の環境にエクスポートしておく必要があります。これらは、SSI によって、次に示すように解釈および使用されます。

CSC でスクリプトを使用して SSI を起動している場合は、そのスクリプト内からこれらの変数を設定およびエクスポートすることをお勧めします。さらに、クライアント開発者によって、エンドユーザーが CSC や動作環境に基づいてそれらの変数を適切に構成するための方法が提供されている可能性があります。

`CSI_UDP_RPCSERVICE` – UDP がネットワーク通信に使用されるかどうかを決定します。

- 機能: SSI ネットワーク通信のための基礎となるネットワークトランスポートレイヤーとしての UDP の使用を有効または無効にします。
- 有効なオプション: TRUE または FALSE

- **その他の詳細:** ファイアウォールセキュリティの CSC の場合は、この環境変数を FALSE に設定する必要があります。ファイアウォールセキュリティの ACSLS アプリケーションのパケットはすべて、TCP ネットワークトランスポートを使用して送信されます。

CSI_TCP_RPCSERVICE – TCP がネットワーク通信に使用されるかどうかを決定します。

- **機能:** SSI ネットワーク通信のための基礎となるネットワークトランスポートレイヤーとしての TCP の使用を有効または無効にします。
- **有効なオプション:** TRUE または FALSE
- **その他の詳細:** ファイアウォールセキュリティの CSC の場合は、この環境変数を TRUE に設定する必要があります。ファイアウォールセキュリティの ACSLS アプリケーションのパケットはすべて、TCP ネットワークトランスポートを使用して送信されます。

CSC ツールキット 2.3 での新しい変数

SSI_INET_PORT – 受信応答のための固定ポート番号。

- **機能:** SSI が受信 ACSLS 応答に使用するポートを指定します。
- **有効なオプション:** 0 または 1024–65535 (50001 と 50004 を除く)。
 - 0 は、ポートの動的な割り当てを可能にする以前の動作を有効なままにすべきであることを示します。
 - 1024–65535 は、その数値を SSI が ACSLS 応答を受け入れる TCP ポートとして使用するべきであることを示します。
 - 50001 または 50004 は、*mini_e1* および SSI によって使用されるため指定しないでください。
- **その他の詳細:** この環境変数を 0 以外の値に設定すると、SSI は、受信 ACSLS 応答にこのポートを使用するようになります。つまり、SSI で ACSLS 応答が受信されるようにするには、ファイアウォールがそのポート上の受信リクエストを許可する必要があります。これは、ACSL S が CSC の SSI との接続を開始する唯一のポートです。

注:

この値は、CSC プラットフォームを保護するファイアウォールで構成され、このポート上の接続への受信リクエストを許可している値に一致している必要があります。

CSI_HOSTPORT – ACSLS サーバー上のポートマッパーへの照会を排除します。代わりに、リクエストを ACSLS サーバー上のこのポートに送信します。

- **機能:** SSI がその ACSLS リクエストを送信する先の ACSLS サーバー上のポートを指定します。ACSL S CSI が、CSC からのインバウンド ACSLS リクエストを受け入れるためにこ

のポートを (つまり、この同じ値に設定されたファイアウォールセキュリティの固定ポートに) 使用している必要があります。

- **有効なオプション:** 1024–65535 (50003 を除く) および 0 (この値は、インバウンドパケットのために CSI によって使用されるポート用に ACSLS サーバー上で設定された値に一致している必要があります)
 - 0 は、ACSL S サーバー上のポートマッパーに照会する以前の動作を引き続き使用することを示します。
 - 1024–65535 は、この値が受信リクエストのために CSI によって使用されることを示します。
 - 50003 は、acslm によって使用されるため指定しないでください。
- **その他の詳細:** この環境変数を設定すると、SSI から ACSLS サーバーのポートマッパーへの照会が排除されます。この変数の値は、SSI がその送信 ACSLS リクエストを送信する先の ACSLS サーバー上のポート番号を指定します。これにより、ファイアウォールで保護された ACSLS サーバーは、そのファイアウォールにあるポートマッパーへのアクセスを禁止できます。ポートマッパー照会では以前、SSI がその ACSLS リクエストを送信する先のポート番号が提供されました。

注:

この値は、受信リクエストの受け入れおよび処理のために CSI によって使用されるポートの値に一致している必要があります。このポートを指定可能な値に確実に固定されたままにするには、ACSL S にファイアウォールセキュリティ機能を適用する必要があります。不一致が存在すると、CSC と ACSLS の間の通信が失われます。

クライアント上の環境変数の表示および設定

クライアントでは、環境変数を設定するために使用されるコマンドはシェルや OS によって異なります。

- UNIX および Linux では、次のコマンドを使用して環境変数を表示します。

```
echo $<variable-name>
```

- ksh および bash シェルでは、次のコマンドを使用して環境変数を設定できます。

```
<environment_variable> = <value>
```

```
export <environment_variable>
```

ファイアウォールセキュリティソリューションのシナリオ

次の図は、ファイアウォールをまたがって使用された場合の ACSLS コンポーネントの動作、ポートの使用法、および関係の考えられるシナリオを示しています。これらは、先に (上で) 示したテキストによって理解されることを目的にしています。次の図の「SSI」は、通信のクライアント側で動作する ACSLS のネットワークインタフェースコンポーネントです。CSI は、ACSLS プラットフォームで動作する ACSLS のネットワークインタフェースコンポーネントです。

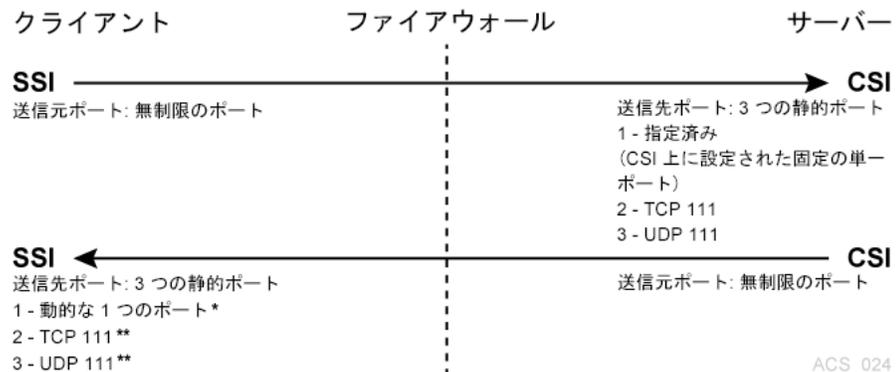
注:

これらのシナリオをサポートするには、ACSLC CSC 開発者ツールキット 2.3 (以降) および新しい環境変数が必要です。

ACSLC サーバー側のみでのファイアウォールセキュリティ

この例では、ファイアウォールセキュリティは ACSLS サーバー側 (CSI) のみに実装されています。このシナリオをサポートするために、CSC ツールキット 2.3 (以降) および新しい環境変数は必要ありません。

図M.1 ACSLS サーバー側のみでのファイアウォールセキュリティ



この例で、動的とは、そのポートが起動時に SSI によって 1024-65535 の範囲から選択されることを示します。このポートはユーザーによって指定されるわけではなく、通常は SSI の新しい実行をはさんで (つまり、SSI 実行中プロセスのあるインスタンスから次のインスタンスにかけて) 同じポートでもなくなります。

SSI 側のポートマッパー 111 ポートが CSI によって照会されることはまれにしかありません。これは、SSI からリクエストパケットで提供された戻りポート番号がその SSI に応答パケットを送り返すために機能しない (つまり、ネットワークインタフェースの障害が発生した場合に、CSI によってアクセスされるだけです。この場合は、再試行メカニズムとして、CSI は使用するポート (これは、ポートマッパーの SSI のプログラム番号の下に登録されています) を SSI 側のポートマッパーに照会します。

ファイアウォールの背後に存在する ACSLS をセキュリティー保護するには、次の設定が必要です。

- ACSLS: 変更を行なったら、ACSLS を再起動する必要があります。

- `CSI_TCP_RPCSERVICE = TRUE`
- `CSI_UDP_RPCSERVICE = FALSE`

(ただし、UDP を使用して ACSLS と通信するクライアントが存在する場合、これは TRUE である必要があります)

- `CSI_USE_PORTMAPPER = ALWAYS` (これは `IF_DUAL_LAN_NOT_ENABLED` でもかまいません)
- `CSI_FIREWALL_SECURE = TRUE`
- `CSI_INET_PORT = <1024-65535 (50003 を除く)>` デフォルトは 30031
- クライアントの SSI 設定 - クライアントをファイアウォールの背後で動作できるようにする環境変数。

- `CSI_TCP_RPCSERVICE = TRUE`
- `CSI_UDP_RPCSERVICE = TRUE` (FALSE でもかまいません)
- `SSI_INET_PORT = 0`

これは ACSLS CSC 開発者ツールキット 2.3 での新しい環境変数であり、このシナリオには必要ありません。

- `CSI_HOSTPORT = 0` または `<1024-65535 (50003 を除く)>` デフォルトは 30031

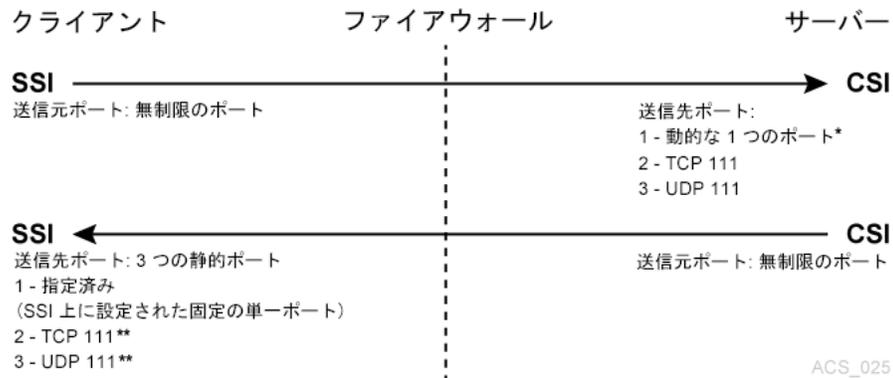
これは、ACSLS サーバー上でポートマッパーを使用している場合は必要ありません。定義されているときに 0 でない場合は、ACSLS サーバー上の `CSI_INET_PORT` に一致している必要があります。これは ACSLS CSC 開発者ツールキット 2.3 (以降) での新しい環境変数であり、このシナリオには必要ありません。

クライアントが (ACSLS サーバー上の) `CSI_INET_PORT` および (クライアント上の) `CSI_HOSTPORT` で指定されたポートを使用して ACSLS サーバーにリクエストを送信できるようにファイアウォールを構成します。クライアントが ACSLS サーバー上のポートマッパー (ポート 111) にアクセスでき、ACSLS がクライアント上のポートマッパー (111) にアクセスできるようにします。

クライアント側のみでのファイアウォールセキュリティ

この例では、ファイアウォールセキュリティはクライアント側 (SSI) のみに実装されています。このシナリオをサポートするには、CSC ツールキット 2.3 (以降) および新しい環境変数がが必要です。

図M.2 クライアント側のみでのファイアウォールセキュリティ



この例で、動的とは、そのポートが起動時に CSI によって 1024-65535 の範囲から選択されることを示します。このポートはユーザーによって指定されるわけではなく、通常は CSI の新しい実行をはさんで (CSI 実行中プロセスのあるインスタンスから次のインスタンスにかけて) 同じポートでもなくなります。

SSI 側のポートマッパー 111 ポートが CSI によって照会されることはまれにしかありません。これは、SSI からリクエストパケットで提供された戻りポート番号がその SSI に応答パケットを送り返すために機能しない (つまり、ネットワークインタフェースの障害が発生した場合)、CSI によってアクセスされるだけです。この場合は、再試行メカニズムとして、CSI は使用するポート (これは、ポートマッパーの SSI のプログラム番号の下に登録されています) を SSI 側のポートマッパーに照会します。

ファイアウォールの背後に存在するクライアントシステムをセキュリティ保護するには、次の設定が必要です。

- ACSLS: 変更を行なったら、ACSLs を再起動する必要があります。
 - `CSI_TCP_RPCSERVICE = TRUE`
 - `CSI_UDP_RPCSERVICE = FALSE`

UDP を使用して ACSLS と通信するクライアントが存在する場合、これは TRUE である必要があります。

- `CSI_USE_PORTMAPPER = ALWAYS` (これは `IF_DUAL_LAN_NOT_ENABLED` でもかまいません)
- `CSI_FIREWALL_SECURE = FALSE`
- `CSI_INET_PORT = 0`
- クライアントの SSI 設定 - クライアントをファイアウォールの背後で動作できるようにする環境変数。
 - `CSI_TCP_RPCSERVICE = TRUE`
 - `CSI_UDP_RPCSERVICE = FALSE` (TRUE でもかまいません)
 - `SSI_INET_PORT = <1024-65535 (50001 と 50004 を除く)>`

これは ACSLS CSC 開発者ツールキット 2.3 (以降) での新しい環境変数であり、このシナリオには必要ありません。

- `CSI_HOSTPORT = 0` または `<1024-65535 (50003 を除く)>` デフォルトは 30031

ACSLs サーバー上でポートマッパーを使用している場合は必要ありません。定義されているときに 0 でない場合は、ACSLs サーバー上の `CSI_INET_PORT` に一致している必要があります。これは ACSLS CSC 開発者ツールキット 2.3 (以降) での新しい環境変数であり、このシナリオには必要ありません。

次のことを可能にするようにファイアウォールを構成する必要があります。

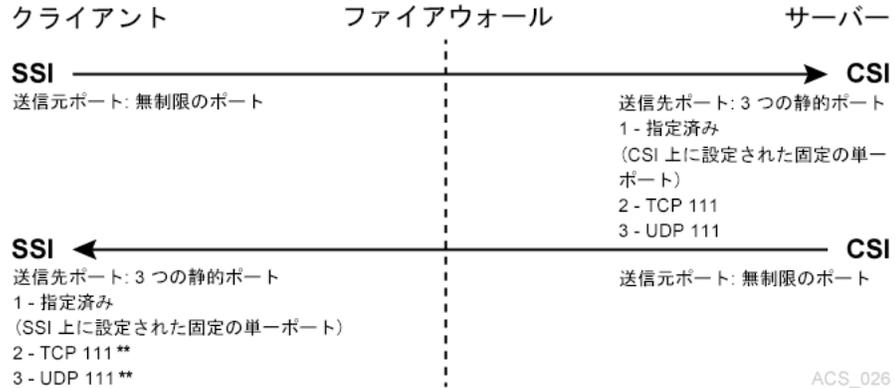
- クライアントが (ACSLs サーバー上の) `CSI_INET_PORT` および (クライアント上の) `CSI_HOSTPORT` で指定されたポートを使用して ACSLS サーバーにリクエストを送信できるようにします。
- クライアントが ACSLS サーバー上のポートマッパー (ポート 111) にアクセスできるようにします。
- ACSLS サーバーがクライアント上の `SSI_INET_PORT` で指定されたポートを使用してクライアントに応答を送信できるようにします。
- ACSLS サーバーがポート 111 を使用してクライアント上のポートマッパーに照会できるようにします。

ポートマッパーを使用した ACSLS サーバー側とクライアント側の両方でのファイアウォールセキュリティ

この例では、クライアント (SSI) と ACSLS サーバー (CSI) の両方がファイアウォールセキュリティ API を実装しています。クライアントとサーバーは、ポートの識別を引き続きポートマッ

パーに依存しています。このシナリオをサポートするには、CSC ツールキット 2.3 (以降) および新しい環境変数が必要です。

図M.3 ポートマッパーを使用した ACSLS サーバー側とクライアント側の両方でのファイアウォールセキュリティ



SSI側のポートマッパー 111ポートがCSIによって照会されることはまれにしかありません。これは、SSIからリクエストパケットで提供された戻りポート番号がそのSSIに応答パケットを送り返すために機能しない(つまり、ネットワークインタフェースの障害が発生した)場合に、CSIによってアクセスされるだけです。この場合は、再試行メカニズムとして、CSIは使用するポート(これは、ポートマッパーのSSIのプログラム番号の下に登録されています)をSSI側のポートマッパーに照会します。

ACSLSサーバーとクライアントの両方を保護しているファイアウォールの場合は、次の設定が必要です。

- ACSLS: 変更を行なったら、ACSLSを再起動する必要があります。

- `CSI_TCP_RPCSERVICE = TRUE`
- `CSI_UDP_RPCSERVICE = FALSE`

ただし、UDPを使用してACSLSと通信するクライアントが存在する場合、これはTRUEである必要があります。

- `CSI_USE_PORTMAPPER = ALWAYS` (これは `IF_DUAL_LAN_NOT_ENABLED` でもかまいません)
- `CSI_FIREWALL_SECURE = TRUE`
- `CSI_INET_PORT = <1024-65535 (50003を除く)>` デフォルトは 30031
- クライアントのSSI設定 - クライアントをファイアウォールの背後で動作できるようにする環境変数。
 - `CSI_TCP_RPCSERVICE = TRUE`

- `CSI_UDP_RPCSERVICE = FALSE`
- `SSI_INET_PORT = <1024-65535 (50001 と 50004 を除く)>`
- `CSI_HOSTPORT = <1024-65535 (50003 を除く)>` デフォルトは 30031

これは、ACSL S サーバー上の `CSI_INET_PORT` に一致している必要があります。

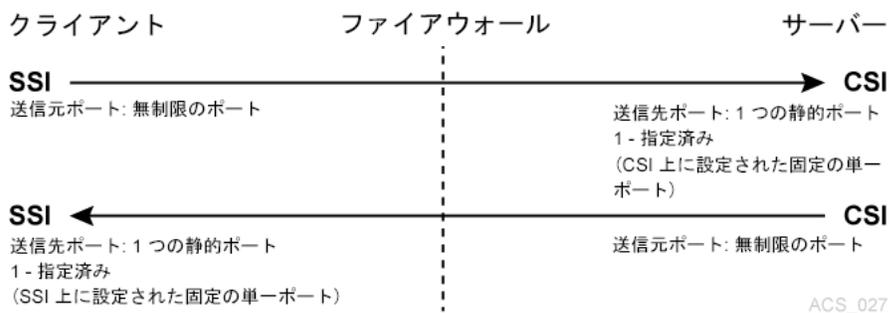
次のことを可能にするようにファイアウォールを構成する必要があります。

- クライアントが (ACSL S サーバー上の) `CSI_INET_PORT` および (クライアント上の) `CSI_HOSTPORT` で指定されたポートを使用して ACSL S サーバーにリクエストを送信できるようにします。
- クライアントが ACSL S サーバー上のポートマッパー (ポート 111) にアクセスできるようにします。
- ACSL S サーバーがクライアント上の `SSI_INET_PORT` で指定されたポートを使用してクライアントに応答を送信できるようにします。
- ACSL S サーバーがポート 111 を使用してクライアント上のポートマッパーに照会できるようにします。

ポートマッパーを使用しない ACSL S サーバー側とクライアント側の両方でのファイアウォールセキュリティ

この例では、クライアント (SSI) と ACSL S サーバー (CSI) の両方がファイアウォールセキュリティ機能を実装しています。クライアントとサーバーでは、「ポートマッパーなし」機能が有効になっています。このシナリオをサポートするには、CSC ツールキット 2.3 (以降) および新しい環境変数が必要です。

図M.4 ポートマッパーを使用しない ACSL S サーバー側とクライアント側の両方でのファイアウォールセキュリティ



もっともセキュアな構成の場合は、次の設定が必要です。

- ACSL S: 変更を行なったら、ACSL S を再起動する必要があります。

- `CSI_TCP_RPCSERVICE = TRUE`
- `CSI_UDP_RPCSERVICE = FALSE`
- `CSI_USE_PORTMAPPER = NEVER`
- `CSI_FIREWALL_SECURE = TRUE`
- `CSI_INET_PORT = <1024-65535 (50003 を除く)>` デフォルトは 30031
- クライアントの SSI 設定 - クライアントをファイアウォールの背後で動作できるようにする環境変数。
 - `CSI_TCP_RPCSERVICE = TRUE`
 - `CSI_UDP_RPCSERVICE = FALSE`
 - `SSI_INET_PORT = <1024-65535 (50001 と 50004 を除く)>`
 - `CSI_HOSTPORT = <1024-65535 (50003 を除く)>` デフォルトは 30031

これは、ACSL S サーバー上の `CSI_INET_PORT` に一致している必要があります。

次のことを可能にするようにファイアウォールを構成する必要があります。

- クライアントが (ACSL S サーバー上の) `CSI_INET_PORT` および (クライアント上の) `CSI_HOSTPORT` で指定されたポートを使用して ACSL S サーバーにリクエストを送信できるようにします。
- ACSL S サーバーがクライアント上の `SSI_INET_PORT` で指定されたポートを使用してクライアントに応答を送信できるようにします。

ACSL S サーバーでファイアウォールセキュリティをオンにする

ファイアウォールセキュリティオプションをオンにするには、`acsss_config` ユーティリティを使用して、いくつかの変数を設定する必要があります。

1. `acsss` としてログインします。
2. ACSL S サーバーを停止します

注:

新しいファイアウォールセキュリティ変数を有効にするには、ACSL S サーバーを停止する必要があります。

`acsss disable`

3. 構成スクリプトを実行するには、次のコマンドを入力します。

`acsss config`

ACSL S 機能の構成画面が表示されます。

4. 「option 1 - Set CSI tuning variables」を選択します

次のものを除き、すべての変数のデフォルト値を受け入れます。

a. 次のプロンプトで、その値を *TRUE* に設定します。

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].

変数: *CSI_TCP_RPCSERVICE*

TCP をオンにすると、ACSL S のクライアントが TCP プロトコルをネットワーク通信に使用できるようになります。ACSL S のファイアウォールセキュリティー機能では TCP のみがサポートされるため、クライアントは、このプロトコルを使用してネットワーク通信を実行してください。

b. 次のプロンプトで、その値を *FALSE* に設定します。

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].

変数: *CSI_UDP_RPCSERVICE*

注意:

この UDP プロトコルに依存している ACSLS クライアントが存在しないことを確認してください。ファイアウォールセキュリティーの ACSLS は、TCP でのみ動作します。

UDP をオフにすると、このプロトコルを使用してサーバーにアクセスするクライアントは存在しなくなります。これにより、ACSL S プラットフォームへのすべての一般的な UDP アクセスをファイアウォールで禁止し、それによって環境内で特に必要なアクセスのみを許可できます。

クライアントがファイアウォールセキュリティー機能を実装していないかぎり、これらのクライアントのポートマッパーアクセスのための UDP および TCP ポート 111 へのアクセスを許可し、これらのクライアントの ACSLS ポートマッパーへの照会を明確にオフにします。

c. 次のプロンプトで、その値を *NEVER* に設定します。

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].

変数: *CSI_USE_PORTMAPPER*

NEVER は、ACSL S のクライアントが、これらのクライアントプラットフォーム上のポートマッパーへの外部アクセスを禁止できるようにします。

重要: これにより、ACSL S プラットフォーム上のポートマッパーへの外部アクセスをオフにすることはできません。それを行うには、ACSL S のクライアントがクライアントソフトウェアコンポーネント内でファイアウォールセキュリティーの変更を採用している必要があり、かつクライアントソフトウェアコンポーネント内でこの機能がオンになっている必要があります。

この機能によって、ACSL S サーバーがクライアントプラットフォーム上のポートマッパーに照会することはなくなります。これにより、クライアントを保護しているどのファイアウォールでもポートマッパーへのアクセスを禁止できます。

- d. 次のプロンプトで、その値を *TRUE* に設定します。

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

変数: *CSI_FIREWALL_SECURE*

TRUE を使用すると、ACSL S がインバウンドクライアント通信 (TCP 接続) を受け入れるために使用する 1 つのポートを指定できます。この変数は、この機能を有効にするだけです。特定のポートは、次の変数で指定されます。

- e. 次のプロンプトで、その値を ACSL S サーバー上の使用可能な固定ポートに設定します。

Port number used by the CSI to receive incoming ACSL S requests.

変数: *CSI_INET_PORT*

これは、受信ネットワーク接続を受け入れるために ACSL S CSI コンポーネントによって使用されるポートです。1024-65535 の範囲内のポート (ポート 50003 を除く) を指定します。

重要: このポート上の受信接続を許可するようにファイアウォールを構成します。これにより、そのポートのみが、ACSL S との通信を開始することを希望する外部のクライアントで使用されるために公開されるようになります。このポートを除くほかのすべての受信ポートおよび UDP/TCP ポート 111 上の接続を禁止できます (ただし、

ポート 111 は、クライアントが ACSLS ポートマッパーへの照会を排除する機能を実装している場合を除きます。その場合は、ファイアウォールでもポート 111 を禁止できません。このポートの推奨されるデフォルト値は 30031 です。ほとんどのシステムで、このポートがほかのプロセスによって使用される可能性は低くなります (ただし、不可能ではありません)。ポートの競合が発生している場合に実行する手順については、付録 J「トラブルシューティング」を参照してください。

5. 「E」を選択して `acsss_config` を終了します。

変更が保存されます。

6. 次のコマンドを入力して ACSLS を再起動します。

```
acsss enable
```

ACSL S サーバーでファイアウォールセキュリティーをオフにする

ファイアウォールセキュリティー機能をオンにするために上で使用された一部の変数は、その機能をオフにする操作にも関連しています。ファイアウォールセキュリティー動作をオフにするには、下の手順を実行するだけで済みますが、サイトによってはほかの変数も変更する場合があります。

1. `acsss` としてログインします。
2. ACSLS サーバーを停止します

注:

新しいファイアウォールセキュリティー変数を有効にするには、ACSL S サーバーを停止します。

```
acsss disable
```

3. 構成スクリプトを実行するには、次のコマンドを入力します。

```
acsss_config
```

4. 「option 1 - Set CSI tuning variables」を選択します

ファイアウォールセキュリティー機能を構成したときに設定された次の値を変更します。次の変数を変更します。

- a. 次のプロンプトで、その値を `ALWAYS` に設定します。

```
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].
```

変数: `CSI_USE_PORTMAPPER`

- b. 次のプロンプトで、その値を `FALSE` に設定します。

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

変数: `CSI_FIREWALL_SECURE`

5. 「E」を選択して `acsss_config` を終了します。

変更が保存されます。

6. 次のコマンドを入力して `ACSLs` を再起動します。

```
acsss enable
```

ファイアウォールセキュリティの構成

以降の操作では、`ACSLs` が背後に存在するネットワークファイアウォールの構成に関連した知識が求められます。ファイアウォールはすべて「サードパーティー」ソフトウェアであり、ネットワーク環境を保護するためのそれらの正しい設定に関連したさまざまな詳細が必要になります。以降の説明は、ファイアウォールセキュリティポリシーの推奨事項というより、むしろ `ACSLs` 製品に関してファイアウォールで実行する必要があること、および実行できることのための一連の役立つ手順としてのみ示されています。その他のセキュリティの詳細については、システム管理者に確認してください。

次に、`ACSLs` プラットフォーム用にファイアウォールを設定するための詳細の一覧を示します。

- UDP 受信および送信接続を禁止するための全体的な規則を設定します。
- TCP 受信接続を禁止するための全体的な規則を設定します (TCP 送信接続は開いたままにしておく必要があります)。
- `ACSLs` で使用するように指定したポート上の受信 TCP 接続を許可するための特定の規則を設定します。**重要:** このポートは、`acsss_config` で構成したポートに一致している必要があります。そうでないと、`ACSLs` サーバーでクライアント通信が受信されなくなります。

すべてのクライアントがファイアウォールセキュリティ機能を実装しており、`ACSLs` プラットフォームのポートマッパーへの照会を行わない場合、作業は完了です。クライアントが引き続き `ACSLs` プラットフォーム上のそのポートマッパーを使用する場合は、次を追加する必要があります。

- 既知のポートマッパー TCP および UDP ポート 111 上の受信および送信接続を許可するための特定の規則を設定します。

例:

次は、上のすべての規則を設定するために iptables ベースのファイアウォールに対して設定された規則の例です。

注:

これらは、特定のファイアウォールに対して構成されるその他の規則に追加されます。

```
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
```

ファイアウォールセキュリティ通信のトラブルシューティング

ACSLS プラットフォームやクライアントを含む (現在では仲介ファイアウォールも含まれます) ネットワーク通信インタフェースのトラブルシューティングには、複数の手順が必要になることがあります。ACSLS とそのクライアントの間のパスにファイアウォールを導入することにより、ネットワーク通信の障害の原因がさらに増える可能性があります。また、ほかのコンポーネント内の設定に対応する方法で構成する必要のあるコンポーネントも増えているため、これらの設定が一致しない場合はネットワーク通信に影響を与えます。ACSLS、そのクライアント、ファイアウォールに対するすべての構成作業を完了したかどうか、およびネットワーク通信が機能していないかどうかをチェックして試行すべき事項の一覧を次に示します。

1. ACSLS プラットフォームのチェック:

- ACSLS が起動して動作していますか。そうでない場合は、*acsss_event.log* をチェックして、考えられる原因または考えられる原因の手がかりを見つけてください。
- CSI が ACSLS によって正常に起動されましたか。そうでない場合は、*acsss_event.log* 内に原因の手がかりとなる通知メッセージが存在していることがあります。何らかの *configuration parameters* の不適切な値または *port conflict* が考えられる原因である可能性があります。
- *acsss_event.log* 内に CSI の障害の原因となる *port conflict* が報告されていますか。その場合は、「*netstat*」または同様のシステムユーティリティを使用してシ

システム上で使用されているポートを調べ、使用可能なポートを使用するように ACSLS を構成してください。ファイアウォールを一致するように再構成することを忘れないでください。

- CSI が予期したポートで登録されていますか。コマンド「`rpcinfo -p`」を使用して portmap テーブルを参照します。CSI は、プログラム番号 300031 の下に登録されています。そのプログラム番号の下に登録されているポートが予期したポートであることを確認してください (デフォルトポートは 30031 であり、プログラム番号より 0 が 1 つ少なくなっています)。
2. ACSLS と CSI が起動して動作し、かつ正しく登録されている場合は、次の手順として、ファイアウォールをまたがる ACSLS プラットフォームへのアクセスをチェックします。
- 基本的な RPC 経由で ACSLS に到達できますか。「`rpcinfo -t <hostname> <program-number> <version-number>`」コマンドを使用して、CSI に単純な RPC リクエストを送信します。(`rpcinfo` コマンドとその使用に関する詳細情報を取得するには、システム上で「`man rpcinfo`」を使用します。)これを、ACSLS を含むファイアウォールの内部にあるマシン (たとえば、その ACSLS プラットフォーム自体) から、およびファイアウォールの外部から実行します。内部からは機能するが、外部から機能しない場合は、ファイアウォールが ACSLS リクエストを遮断しています。ファイアウォールと ACSLS ポートの構成を二重にチェックします。ファイアウォール経由でポートマップパーにアクセスできることを確認します (ポートマップパーへのアクセスが禁止されている場合、このテストをファイアウォールの外部から使用することはできません)。
 - ACSLS とファイアウォールに対して構成されている各ポートが一致していますか。これらのパラメータを二重にチェックします。これは、ネットワーク通信での障害の原因である可能性があります。構成されている値は別にして、上で説明した「`rpcinfo -p`」コマンドを実行して、CSI が実際に予期したポート番号で登録されていることを確認します。そうでない場合は、`acsss_event.log` を参照して原因に関する情報を見つけてください。
 - ACSLS がリクエストを受信しているが、応答を送り返せない状態ですか。`acsss_event.log` をチェックして、CSI が多数の削除されたネットワークパケットまたはネットワーククライアントとの通信の障害を報告していることを見つけた場合は、クライアントリクエストは受信されているが、応答が送信されていません。この場合も、それらがファイアウォールによってブロックされていることを示しています。
3. 問題がまだ解決されない場合。

上に示した方法は、調べるべきいくつかのレベルの事項に対処しています。これらによって特定の答えが得られない場合は、次に、通信がどこで切断されているかを見つけるために何らかのより低レベルのチェックを行います。最善の方法として、ネットワークパケット

トスニファー機能 (Solaris の下での「snoop」など) の使用があります。*snoop* コマンドとその使用に関する詳細情報を取得するには、Solaris ベースのシステム上で「man snoop」を使用します。

ネットワークに接続されたほかのシステムでも、同様のパケットトレース機能が使用できます。

- これを使用するには、パケットがどこに向かっていて、どこで失われているかを示す場所からパケットスニフングを実行する必要があります。これは、ファイアウォールの内部と外部の両方からである可能性があります。
- さらに、パケットデータを調べると、何らかの情報が得られます。どちらかの側がポートマッパーの使用を許可している場合は、いくつかの PORTMAP パケットが表示される可能性があります。
- また、ACSL S とそのクライアントの間を通過している RPC パケットも表示されることがあります。
- 最後に、トランスポートレベルの TCP 接続を調べると、接続の各側で使用されている特定のポートがわかります。これは、多くの場合、通信がどこで停止されているかを見つけるための重要な情報です。

これらの操作の実行に関するさらに詳細な説明は、このマニュアルの範囲を超えています。ただし、システム管理者は、この領域における何らかの支援を提供できます。

よくある質問

- ACSLS のファイアウォールセキュリティーソリューションが必要な理由は何ですか。

ファイアウォールセキュリティーソリューションを使用すると、ACSL S をファイアウォールの背後で効率的に実行でき、さらにセキュリティーが大幅に強化されるようにそのファイアウォール上のポートを制限できます。

- どのリリースの ACSLS がファイアウォールセキュリティー機能をサポートしていますか。

ACSL S 7.0.0 以降のみがこの機能をサポートしています。

- このファイアウォールセキュリティー機能を使用する場合、開いたままにしておく必要のあるポートの最大数はいくつですか。

受信ネットワーク接続を許可することが必要になる可能性のあるポートの最大数は、ACSL S ネットワークインタフェースに 1 つと、ポートマッパー (UDP および TCP 111) に 2 つの合計 3 つです。送信ポートは、受け入れられている業界のセキュリティー習慣に従って無制限です。

- 開いたままにしておくことができるポートの最小数はいくつですか。

最小数は 1 です。これは、クライアント (ISV ソフトウェア) が自身のクライアントにもファイアウォールセキュリティー機能を実装しており、ACSL S プラットフォーム上に存在するポートマップへの照会を行わない場合に可能です。その場合、受信接続に対して開く必要のある唯一のポートは、ACSL S ネットワークインタフェースによって使用されるユーザー指定の 1 つの TCP ポートです。

- この機能ではなぜ、ある範囲のポートを使用しないのですか。

ある範囲のポートを使用することにアーキテクチャー上のメリットはなく、セキュリティー上のデメリットがいくつかあります。ファイアウォールセキュリティー以外の ACSL S では、いずれかの特定のプラットフォーム上で使用可能な動的ポートの範囲全体で構成されるある範囲のポートを使用します。これは、サイトのセキュリティーに対する潜在的な危険として正しく認識されています。これを ACSL S のパフォーマンスに悪影響を与えることなくできるだけ制限することが、その危険を解消するための目標です。ACSL S ネットワークインタフェースでは常に 1 つの受信ポートのみを使用するため、そのポートが ACSL S プラットフォームで ACSL S 専用になっているかぎり、1 つのポートを超えて範囲を拡張する理由はありません。

- 選択したポートが、システムでのそのポートの別の使用方法と競合する場合はどうしたらよいですか。

これは、そのポートがユーザー指定可能になっている理由の 1 つです。使用可能な特定のポートは、顧客サイトによって異なります。ユーザーが 0-1023 の既知の予約ポートのいずれかを使用することは許可されません。30031 のデフォルトポートは登録済みポートの範囲内にあります。これにより、動的ポートを使用している別のアプリケーションがそれを使用する可能性は低くなります (ただし、不可能ではありません)。これは登録済みポートの範囲内にありますが、それを使用するように登録されているアプリケーションは存在しないため、それが妥当なデフォルトの選択になっています。

- この機能を使用すると ACSL S サーバーをファイアウォールで保護できますか。

はい。この機能が設定されていると、ACSL S サーバーをファイアウォールの信頼できる側に配置でき、クライアントは反対 (信頼できない) 側または同じ側からそのサーバーにアクセスします。

- この機能を使用すると ACSL S クライアント (ISV コンポーネント) をファイアウォールで保護できますか。

可能性はありますが、単独ではできません。このシナリオを実現するには、使用しているクライアントソフトウェアコンポーネント (ACSL S のクライアント) が、StorageTek ACSL S

Client System Component 開発者ツールキットで入手可能なファイアウォールセキュリティ機能を採用している必要があります。そのステータスに関する最新の更新については、使用しているクライアントソフトウェアのプロバイダにお問い合わせください。

- クライアントをファイアウォールで保護できるようにする場合はどうしたらよいですか。

使用しているクライアントソフトウェアのプロバイダにお問い合わせください。それらのプロバイダは、その CSC (クライアントソフトウェアコンポーネント) でファイアウォールセキュリティの変更を採用しているかどうかを伝えることができます。

- ポートマッパーについてはどうでしょうか。ポートマッパーへのアクセスを完全に禁止することはできますか。

クライアントがファイアウォールセキュリティの変更を採用している場合は、ACSL S プラットフォームのポートマッパーへのクライアントの照会を停止できる可能性があります。その場合は、ACSL S プラットフォームを保護するファイアウォールでポートマッパーへのアクセスを禁止できます。ほかのいずれの場合も、クライアントは ACSLS ネットワークインタフェースとの接続を確立するために ACSLS サーバー側のポートマッパーに依存するため、クライアントがそのポートマッパーを使用できるようにしておく必要があります。

- ACSLS サーバーのファイアウォールが ACSLS プラットフォームのポートマッパーへのアクセスを停止するために、クライアントがいくつかの変更を実装する必要があるのはなぜですか。

ACSL S プラットフォームのこれらの照会を行なっているのはクライアントであるためです。クライアントが引き続きこれらの照会を行う場合は、その照会を成功させるために、ACSL S プラットフォームは引き続きポートマッパーのサービスを提供する必要があります。

- ポートマッパーが悪いのではないかと思います。それをなぜ完全に削除しなかったのですか。

ポートマッパーは、レガシークライアントに重要なサービスを提供しています。これを完全に削除すると、これらのクライアントが依存しているインタフェースが無効になります。つまり、新しいポートマッパー以外のインタフェースの再コーディング、再テスト、および再認証なしでは、どのレガシークライアントも機能しません。このファイアウォールセキュリティソリューションでは、ACSL S からクライアントへと、クライアントから ACSLS への両方のポートマッパーへの照会を削除する機能を提供しましたが、クライアントソフトウェアをこれに強制的に準拠させることはできません。そのため、サイトのクライアントがファイアウォールセキュリティ機能を採用し、ポートマッパーサービスを使用しなくなるまでは、ポートマッパーを少なくともオプションサービスとして使用可能なままにしておく必要があります。

- クライアントの一部がファイアウォールセキュリティ機能を採用し、一部はまだ採用していません。これを利用するにはどうしたらよいですか。

これらの機能を採用しているクライアントを、それぞれ独自のファイアウォールの背後で保護できます。さらに、ポートマッパーの既知のポートへのアクセスをファイアウォールで制限したあと、それを必要としているクライアントからのポートマッパーへのアクセスのみを許可するように構成できます。これらの詳細および機能は、そのサイトで使用されている特定のファイアウォールによって異なります。

- RPC が悪いのではないかと思います。それをなぜ完全に削除しなかったのですか。

ACSL S ネットワークインタフェースは、ACSL S の最初のリリースから RPC に基づいていました。これは効率的で、安定した、さらに信頼できるメカニズムであり、ネットワーク通信のレイヤーでさまざまなメリットを提供することが実証されています。ただし、そのポートの一般的な動的割り当てやポートマッパーの使用のために、RPC を使用するプラットフォームのセキュリティ保護がより困難になる場合もあります。このファイアウォールセキュリティソリューションでは、これらの領域の両方に対処しています。これにより、顧客はファイアウォールを制限された方法で効率的に構成しながら、ファイアウォールを設定する目的であるセキュリティの利点を得ることができます。

さらに、RPC を ACSL S ネットワークインタフェースから完全に削除すると、現在のすべての (レガシー) ACSL S クライアントが無効になるため、その CSC (クライアントソフトウェアコンポーネント) の再コーディング、再テスト、および再認証なしでは、どの ACSL S クライアントも ACSL S と通信できなくなります。

- ファイアウォールセキュリティ機能は、ACSL S クライアントと ACSL S サーバーの間のネットワーク通信のパフォーマンスやタイミングにどのような影響を与えますか。

新しいファイアウォールセキュリティ機能によるパフォーマンスへの影響はありません。ファイアウォールの使用状況がパフォーマンスに影響を与える可能性はありますが、これは特定の顧客ごとのファイアウォール実装の運用特性に基づきます。パフォーマンスへの影響を無視できるファイアウォールの場合、ACSL S とそのクライアントは引き続き、ファイアウォールセキュリティ機能のインストール前と同様に動作します。また、ファイアウォールによる遅延を適切に処理できるように、ACSL S ネットワークインタフェースの許容範囲を構成することもできます。

- ファイアウォールセキュリティ機能は、ACSL S 操作の残りの部分にどのような影響を与えますか。

ファイアウォールセキュリティソリューションのインストールによる ACSL S 操作のほかの部分への効果や影響はありません。

- ファイアウォールセキュリティ機能は、クライアントが (ACSAPI 経由で) 使用する ACSLS 機能にどのような影響を与えますか。

ACSAPI 経由で提供されている (また、ACSLS クライアントが現在 ACSLS とインタフェースするために使用している) すべての機能が、ファイアウォールセキュリティ機能の下で、この機能がインストールされていない場合と同様に動作します。特に、このファイアウォールセキュリティ機能はアクセス制御だけでなく、ACSLS 製品に追加されたすべての新機能もサポートしています。ACSAPI のすべての機能が、この機能によって引き続きサポートされます。

- ファイアウォールセキュリティ機能は、ACSLS HA (高可用性) ソリューションと連携して動作しますか。

ファイアウォールセキュリティ機能が HA 操作に悪影響を与えることはありません。ただし、HA ソリューションはファイアウォールをまたがって (つまり、各 HA サーバーがファイアウォールの反対側にある状態で) 動作するように設計されていません。HA ソリューションにはポートマッパーへのリモートアクセスが必要であるため、各サーバーをファイアウォールの反対側で実行しようとする試みが行われたとしても、ファイアウォールはそのアクセスを禁止できません。ファイアウォールをまたがった実行には、HA 設定に悪影響を与える可能性のあるその他の詳細が存在します。これを実行しないようにすることを強くお勧めします。

HA サーバーがファイアウォールの同じセキュリティ保護された側に設定されている場合は、一連の HA サーバーをファイアウォールセキュリティ機能とともに設定できます。そのとき、ファイアウォールの反対側にあるクライアントはこれらのサーバーとファイアウォールをまたがって同じパフォーマンスで対話したり、ファイアウォールセキュリティ以外の HA ソリューションに対する場合と同様に動作したりできます。

- このファイアウォールセキュリティ機能は、ほかの StorageTek ソフトウェア製品と連携して動作しますか。

ほかの StorageTek 製品およびパートナー製品 (つまり、ACSLS と通信するクライアントソフトウェアコンポーネント) との相互運用性は完全に保持されています。これらの製品は変更なしで引き続き動作し、ACSLS サーバーがセキュリティ保護されたファイアウォールの背後で動作している状態で、または (現在と同様に) これらの製品と同じ環境内で ACSLS サーバーと通信します。

- ほかに StorageTek ソフトウェア製品には同じファイアウォールセキュリティ機能がありますか。

ほかの StorageTek 製品は、単純にファイアウォールセキュリティーの ACSLS と同じ環境で使用されるという理由から、ファイアウォールセキュリティーの利点を備えていません。各製品はファイアウォールでセキュリティー保護された ACSLS と連携して動作できますが (前の質問を参照してください)、これらの各製品をそれぞれ独自のファイアウォールの背後に配置することは、その特定の製品自体で対処する必要のある課題です。一部の StorageTek 製品にはすでに、これらの製品が動作するプラットフォームを保護するために使用されるファイアウォールで何らかの制限を許可する組み込みのポリシーが含まれています。さらに、ACSLS に対するクライアントとして機能する製品はすべて、ACSLS に対して行われ、ACSLS CSC 開発者ツールキット 2.3 (以降) の一部として提供されているファイアウォールセキュリティーの変更を採用するオプションを備えています。

この章では、CSCI (Client Server Communications Interface) コンポーネントについて説明します。

この章で扱う内容は、次のとおりです。

- CSCI の概要
- エラーメッセージ
- 環境変数

CSCI のインストールは、オペレーティングシステムのインストールおよび ACSLS のインストールと構成のあとに行うようにしてください。

注:

ACSLs が正しく機能するには、START_CSCI_PROCESS 構成変数が TRUE に設定されている必要があります。これは、acsss_config のオプション 1 で実行します。6章「[ACSLs の動作を制御する変数の設定](#)」を参照してください。

CSCI の概要

CSCI (Client Server Communications Interface) は、クライアントシステムとサーバーシステム間のリクエストおよび応答パケットを渡す役割を果たすスタンドアロンのコンポーネントです。CSCI は、MVS または RMLS/CSC クライアントが ACSLS と通信できるようにするオプションコンポーネントです。CSCI を使用しない場合、デフォルトでは CSI を使用して ACSLS と通信します。

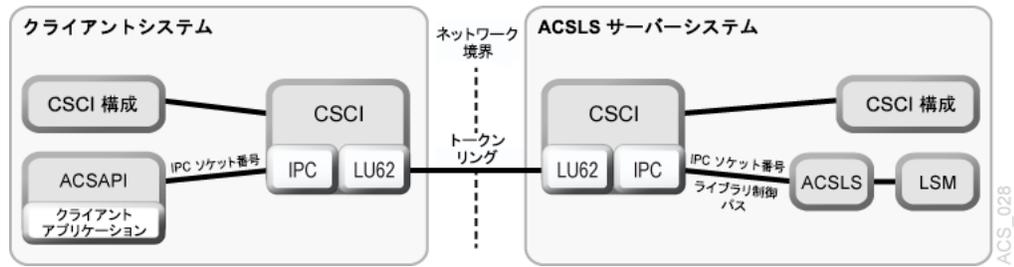
CSCI 接続

CSCI は、ACSLs サーバーを MVS クライアントまたは RMLS/CSC クライアントにリンクする IPC および TCP/IP 接続を提供します。

アーキテクチャーおよびサブシステム

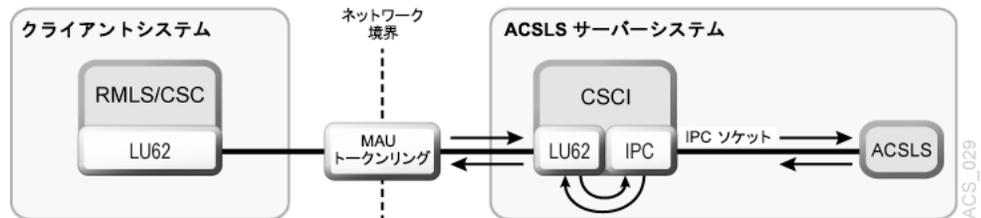
次の図は、全体的な CSCI アーキテクチャーおよびサブシステムを示しています。

図N.1 CSCI アーキテクチャーおよびサブシステム



次の図は、CSCI システムがネットワークインタフェースに接続する方法を示しています。サーバー側では、データは、ネットワークハブから TCP/IP インタフェース経由で CSCI インタフェースに流れます。ここから、データは IPC インタフェース経由で ACSLS ソフトウェアに両方向に、ただし半二重モードで (つまり、一度には一方向に) 流れます。

図N.2 トークンリングインタフェースを使用した CSCI システム



エラーメッセージ

CSCI のインストールおよび構成エラーメッセージは、次の状態に関連しています。

- 正しくない所有権
- 正しくないアクセス権
- 未定義の環境変数
- その他のさまざまな状態
- 正しくない CSCI および TCP/IP 構成

一般的なエラーメッセージ

Communication failure, verb = cmroutine, rc = xx, errno = yyy IPC failure on Socket CSCIZzzzz

説明: 指定された接続で読み取りに失敗しました。

変数:

- `xx` は、IBM CPI リターンコードです
- `yyy` は、`/usr/include/sys/errno.h` 内の UNIX の `errno` 値です
- `CSCIzzzzz` は、使用中であったソケット番号です
- `cmroutine` は、失敗した IBM CPI ルーチン名です

Read failed for connection: connection_name

説明: 指定された接続で読み取りに失敗しました。

変数: `connection_name` は、読み取りに失敗した接続の名前です。

Write failed for connection: connection_name

説明: 指定された接続で書き込みに失敗しました。

変数: `connection_name` は、書き込みに失敗した接続の名前です。

環境変数

CSCI は、ACSLs 実行可能ファイルが存在する `/$ACS_HOME/bin` ディレクトリ内のトランザクションプログラム名 (TPN) である `csciTcpServer.sh` によって呼び出されます。このシェルスクリプトは、CSCI 環境変数を定義します。

次の表は、サーバーサブシステムの CSCI 環境変数と、ACSLs/CSCI ソフトウェアの配布パッケージで提供されているものと同様のシェルスクリプトについて説明しています。このシェルスクリプトは、サーバー上の TCP/IP によって開始されます。

表N.1 CSCI サーバーサブシステムの環境変数

名前	説明
<code>START_CSCI_PROCESS</code>	ACSLs の起動時に CSCI を自動的に起動します (TRUE または FALSE)
<code>CSCI_TRACE</code>	この変数は、CSCI トレースが ON または OFF のどちらであることを示します。
<code>CSCI_SLEEP_TIME</code>	この変数は、接続テーブルを使用した各ラウンドロビンループの最後の PER スリープ時間 (ポーリングタイムアウト) を示します。デフォルトは 100 ミリ秒です。
<code>CSCI_INITIAL_CONNECTION</code>	この変数は、最初に関する CSCI の初期接続名を示します。これは、CSCI の初期起動時の I/O に必要な接続サービス名です。通常、

名前	説明
	クライアントは最初に IPC を開き、サーバーは最初に TCP/IP を開きます。
<i>CSCI_SERVER_NAME</i>	この変数は、CSCI サーバーの名前を示します。これは、1 つの CSCI サーバーに対して定義されているすべての CSCI クライアントで同じに設定されている必要があります。このサーバー名は、その CSCI クライアントの <i>CSCI_SERVER_NAME</i> に一致している必要があります。サーバーとクライアントの <i>CSCI_SERVER_NAME</i> の一致により、マシン間のエンドツーエンドの論理リンクが提供されます。最後に、この名前は、あとでほかの環境変数内の「connectionname」に使用されます。 <i>CSCI_SERVER_NAME</i> に割り当てられた値を、以降の変数の connectionname 部分にレプリケートする必要があります。
<i>CSCI_connectionname_NET_TYPE</i>	この変数は、この CSCI が CIF サブシステムから構成されているネットワークタイプを示します。 <i>NETTYPE</i> は、この CSCI によって使用されるトランスポートレイヤーを定義します。有効な割り当ては LU62 または IPC です。
<i>CSCI_connectionname_INPUT_SOCKET</i>	この変数は、この CSCI への入力として使用する入力ソケット番号を示します。この番号は、サーバーまたはクライアントアプリケーションの出力ソケット番号に一致します。
<i>CSCI_connectionname_CON_TYPE</i>	この変数は、この CSCI のタイプを示します。この <i>CON_TYPE</i> は、この CSCI を SERVER または CLIENT として定義します。
<i>CSCI_connectionname_TRANSLATE_TYPE</i>	この変数は、実行中に使用されるこの CSCI の XDR タイプを示します。これは、XDR または NONE に割り当てることができます。
<i>CSCI_connectionname_DESTINATION_SOCKET</i>	この変数は、CSCI サーバーによって使用される宛先または出力ソケット番号を示します。このソケット番号は、CSCI サーバーアプリケーションへの入力ソケット番号に一致します。

付録O メディア管理

この章では、異なる密度で記録された同じタイプのメディアを管理する方法と、古い密度で書き込まれたカートリッジと新しい密度で書き込まれたカートリッジとを区別する方法について説明します。

概要

この概要では、次の内容を理解できます。

- 遭遇する可能性のある制限事項と問題
- 異なる密度で同じメディアに記録するテープドライブの例
- 推奨の解決方法

制限事項

既存のテープメディアを使用しながらより高い密度で記録する新しいテープドライブが導入される場合があります。これらの新しいドライブは、多くの場合、古い密度で記録されたテープを読み取れますが、古い密度で書き込むことはできません。古いテープドライブでは、高密度で読み取ることも書き込むこともできません。

このような制限のために、次の問題が発生することがあります。

- 新しい密度で書き込まれたテープが古いドライブにマウントされたときに、古いドライブでこのテープを読み取ることができません。
- ストレージ管理アプリケーションが、あとからファイルを追加することで、一部が使用されているテープに書き込もうとしたときに、新しいテープドライブで読み取れるが書き込めない密度とは異なる密度でテープが書き込まれていた場合、この書き込みは失敗します。

ライブラリ内に古いテープドライブと新しいテープドライブの両方が混在している場合は、メディアタイプが同じテープカートリッジを管理する必要があります。

例

次の例には、同じメディアに別々の密度で記録するテープドライブの組み合わせが示されています。

- T10000A と T10000B のテープドライブで使用される T10000 メディア

T10000B テープドライブは、T10000A と同じメディアを使用しますが、T10000A の倍の密度でデータを書き込みます。T10000B は、T10000A メディアを読み取り、T10000B 密度のデータの書き込みにそのメディアを再利用する (テープの先頭から書き込む) ことができますが、以前に書き込まれた T10000A メディアにデータを追加することはできません。

T10000A ドライブは、T10000A 密度のデータを書き込むために T10000B カートリッジを再利用できますが、T10000B カートリッジに対してデータを読み取ることも、データを追加することもできません。

- T9840A、T9840B、T9840C および T9840D のテープドライブで使用される 9840 メディア

T9840A、T9840B、T9840C、T9840D のドライブの任意の組み合わせが存在する場合、次の理由から 9840 メディアを管理します。

- T9840A と T9840B

T9840A と T9840B のテープドライブは同じメディアを使用し、同じ密度で記録します。

- T9840C

T9840C は、T9840A および T9840B と同じメディアを使用しますが、倍の密度で記録します。

- T9840D

T9840D は、T9840C ドライブのほぼ倍の密度で書き込みます。

T9840A、T9840B、T9840C、T9840D のドライブの任意の組み合わせが存在する場合、9840 メディアを管理します。

- T9940A と T9940B のテープドライブで使用される 9940 メディア

- T9940A

T9940A ドライブは、単密度データの書き込みに T9940B カートリッジを再利用できますが、T9940B カートリッジに対してデータの読み取りも、データの追加も行えません。

- T9940B

T9940B テープドライブは、T9940A と同じメディアを使用しますが、T9940A の倍の密度でデータを書き込みます。T9940B は、T9940A メディアを読み取り、倍密度データの書き込み用にこのメディアを再利用できます。ただし、以前に書き込まれた T9940A カートリッジへのデータの追加はできません。

解決方法

ACSLS には、2 種類以上のドライブで書き込めるが記録密度に互換性がない一般的なメディアを管理するために使用できる複数のツールがあります。クライアントアプリケーションは、データの読み取り/追加の非互換性を管理するために、これらのツールを使用する必要があります。

ACS 内で、一般的なメディアに異なる密度で記録するドライブを管理する場合、次のいずれかの方法を使用します。

- ACS 内のすべての古いドライブを同時に新しいドライブに交換する。

これは、もっとも簡単でもっとも安全な方法です。この方法を使用して、異なる密度を使用したドライブの組み合わせの管理によって生じる問題を回避します。これを実施できない場合は、2 番目の項目で説明するように、徐々に古いドライブを新しいドライブに交換できます。

注:

古いドライブを新しいドライブに交換したあとで、古い記録密度で書き込まれたテープにファイルを追加しないでください。Veritas NetBackup では、これはテープを一時停止して行われます。

- 徐々に古いドライブを新しいドライブに置き換える。

この場合、異なる密度で記録された一般的なメディアを管理する必要が生じます。これは次のようにして行えます。

- カートリッジが SL8500 および SL3000 ライブラリ内でマウント解除されるときに返される記録形式情報を使用してメディアを管理する。これについては、次のセクションで説明します。
- 形式ごとに別々の ACSLS プールを作成する。
- バックアップアプリケーション (Veritas NetBackup、Legato NetWorker、IBM Tivoli、CA BrightStor など) の機能を使用してメディアプールを管理する。

ACSLS によって報告される記録密度を使用した一般的なメディアの管理

最新のライブラリ内の最新のテープドライブからカートリッジをマウント解除するとき、カートリッジの記録形式が ACSLS に報告されます。ACSLS は記録形式をそのデータベースに保存します。`display volume` コマンドを使用して、この情報を表示できます。

記録形式は次の装置から報告されます。

- ライブラリ:
 - SL3000
 - SL8500 (4.10 以降のファームウェア)
- テープドライブ:
 - すべての T10000 テープドライブ (1.38 以降のファームウェア)
 - T9840A、T9840C および T9840D (T9840B を除くすべての T9840 テープドライブ) (1.4 .2 以降のファームウェア)
 - T9940A および T9940B テープドライブ (1.4.2 以降のファームウェア)

次の例では、T10000A と T10000B の両方の形式で記録された T10000 メディアを管理する方法について説明します。次の表を使用して、自身の具体的な事例に手順を適用してください。

古い形式のドライブ	新しい形式のドライブ
T10000A	T10000B
T9940A	T9940B
T9840A/T9840B	T9840C または T9840D
T9840C	T9840D

注:

以降の説明では、コマンドおよびユーティリティーに次の構文規則を使用します。

- そのまま入力されるコマンドとユーティリティーは太字で示されます。
- 変数 (正しい値を入力する必要があります) は斜体で示されます。

手順

T10000B ドライブを取り付けたあとで、記録形式情報を使用して、互換性のあるテープドライブにカートリッジをマウントします。繰り返しになりますが、T10000A の密度で書き込まれたカートリッジは、T10000A ドライブまたは T10000B ドライブで読み取ることができませんが、T10000A の形式で書き込まれたデータを追加できるのは T10000A ドライブだけです。T10000B ドライブだけが、T10000B 形式で書き込まれたカートリッジに対して読み取りや追加を行えます。

スクラッチカートリッジをマウントするときに特別な処理は必要ありません。テープの開始位置からカートリッジに書き込む場合は、以前の記録形式はどれでもかまいません。

データがすでに記録されているカートリッジに対して読み取りまたは追加を行うドライブを選択するには、この手順に従います。次のコマンドは、カートリッジの記録形式を読み取りおよび書き込みできるドライブを識別します。

非スクラッチカートリッジのマウント

1. カートリッジの記録形式を表示します。

```
display volume vol_id -f recording_format_family recording  
_format_model
```

このボリュームの記録形式が表示されます。

2. 次の `query` コマンドを使用して、互換性のあるドライブを識別します。

```
query mount vol_id
```

カートリッジと互換性のあるドライブが表示されます。

- カートリッジをマウントする前に、正しいドライブタイプを選択します。

この `query` コマンドでは、T10000A と T10000B の両方のドライブが返されます (どちらのドライブタイプも T10000 メディアと互換性があるため)。

- ドライブタイプを使用して、T10000A と T10000B を区別します。

リスト内の互換性のある最初のドライブを選択すると、パススルーが最小限に抑えられ、ライブラリのパフォーマンスが向上します。

3. 次のコマンドを使用して、記録形式と互換性のあるドライブにカートリッジをマウントします。

```
mount vol_id drive_id
```

新しい記録形式への移行

- カートリッジ上のすべてのデータが期限切れになると、新しい記録形式でテープの開始位置からカートリッジに再度書き込むことができます。

期限切れのカートリッジの詳細は、「[LSM の設定](#)」を参照してください。

- すべての T10000A ドライブが T10000B ドライブに変換または交換されたあとで、T10000A 形式で書き込まれたカートリッジにデータが追加されていないければ、すべての T10000 メディアを T10000B ドライブで使用できます。

ACSL S プールを使用した、異なる密度で記録された一般的なメディアの管理

次の例では、T9940A と T9940B の両方の密度で記録された 9940 メディアを管理する方法について説明します。次の手順は、別々のテープドライブが異なる密度で一般的なメディアに記録するすべての場合に適用されます。次の表を使用して、自身の具体的な事例に手順を適用してください。

古い形式のドライブ	新しい形式のドライブ
T10000A	T10000B
T9940A	T9940B
T9840A/T9840B	T9840C または T9840D
T9840C	T9840D
SDLT 220	SDLT 320

手順

1. ACS 内のすべての 9940 データ (非スクラッチ) カートリッジを特定し、それを T9940A プールに割り当てます。これは、T9940A ドライブを含む ACS 内に T9940B ドライブを取り付ける前に行なってください。

これで、9940 スクラッチカートリッジを T9940A または T9940B プールに割り当てることができます。

- a. 次のコマンドを使用して、T9940A および T9940B のメディアプールを定義します。

```
define pool pool_id
```

- b. すべての 9940 メディアを報告します。

次に示すように、*display volume* コマンド (オプション 1) またはカスタムの *volrpt* (オプション 2) のどちらかを使用して、これを行えます。

結果がファイルに書き込まれます。

- オプション 1

display volume コマンドを使用してファイルに結果を書き込みます。

```
display volume * -media STK2P > filename
```

ここでは:

すべての 9940 テープカートリッジ (メディア STK2P) が報告されます。

`filename` は、出力が書き込まれるファイル名です。カートリッジが配置されている **ACS ID** も一覧表示されます。

必要に応じて特定の ACS からカートリッジを選択して、出力を読み取ります。

1 つの ACS のカートリッジが必要な場合は、`-home` オペランドを使用して、この ACS のカートリッジだけを選択します。

```
display volume * -home acs_id.*,*,*,* -media STK2P> filename
```

- オプション 2

ACS 内のすべてのボリュームに対してカスタムの `volrpt` を使用します。結果がファイルに書き込まれます。

```
volrpt -d -f custom_volrpt_file -a acs_id > filename
```

ここでは:

`custom_volrpt_file` は、カスタムの `volrpt` で報告されるフィールドを指定したファイル名です。次のフィールドが報告されます。

```
VOLUME_ID      6  2
```

```
MEDIA_TYPE     7  2
```

```
VOLUME_TYPE    4  2
```

`acs_id` は、管理している ACS の ID です。

`filename` は、出力が書き込まれるファイル名です。

メディアタイプが STK2P のボリュームだけを選択して、出力を読み取ります。

- c. 選択されたボリュームを、適切なプールに割り当てます。

次のコマンドを使用して、すべての非スクラッチ (`VOLUME_TYPE = D`) カートリッジを T9940A プールに割り当てます。

```
set scratch off pool_id vol_id
```

次のコマンドを使用して、スクラッチカートリッジ (`VOLUME_TYPE = S`) を T9940A または T9940B のどちらかのプールに割り当てます。

```
set scratch pool_id vol_id
```

2. T9940B ドライブを取り付けたあとで、プールを使用して、互換性のあるテープドライブにカートリッジをマウントします。

繰り返しになりますが、T9940A の記録密度で書き込まれたカートリッジは、T9940A ドライブまたは T9940B ドライブのどちらでも読み取ることができますが、9940A メディアにデータを追加できるのは T9940A ドライブだけです。T9940B ドライブだけが、9940B の記録密度で書き込まれたカートリッジに対して読み取りや追加を行えます。

次のコマンドを使用して、T9940A または T9940B ドライブに適したプールからカートリッジを識別し、選択し、マウントできます。プールを使用して、確実に正しいドライブにカートリッジをマウントします。

注:

正しいドライブタイプにスクラッチカートリッジをマウントする場合は、手順 A に従います。データがすでに記録されている非スクラッチカートリッジをマウントする場合は、手順 B に従います。

手順 A - スクラッチカートリッジのマウント

- a. ドライブに照会してドライブタイプを識別します。

```
query drive drive_id | all
```

- b. カートリッジを目的のドライブにマウントするために、正しいプール (メディアタイプ) を識別します。
- c. 指定のプールのドライブにスクラッチカートリッジをマウントします。

```
mount * drive_id pool_id
```

手順 B - 非スクラッチカートリッジのマウント

- a. カートリッジと互換性のあるドライブのステータスを表示します。

```
display volume vol_id -f pool
```

このボリュームのプールが表示されます。

- b. 照会コマンドを使用して、互換性のあるドライブを特定します。

```
query mount vol_id
```

カートリッジと互換性のあるドライブが表示されます。

カートリッジをマウントする前に、正しいドライブタイプを選択します。この照会コマンドでは、T9940A と T9940B の両方のドライブが返されます (どちらのドライブタイプも 9940 メディアと互換性があるため)。

ドライブタイプを使用して、T9940A と T9940B を区別します。

- c. 選択したドライブにカートリッジをマウントします。

```
mount vol_id drive_id
```

3. カートリッジ上のすべてのデータが期限切れになると、スクラッチカートリッジを新しい記録密度のプールに移行できます。T9940A ドライブと T9940B ドライブはそれぞれの密度でカートリッジを再初期化できるため、スクラッチカートリッジを別のプールに再度割り当てることができます。

```
set scratch pool_id vol_id
```

4. すべての T9940A ドライブが T9940B ドライブに変換または交換されたあとで、データが T9940A 記録密度で書き込まれたカートリッジに追加されていないならば、すべての 9940 メディアを T9940B ドライブで使用できます。

付録P XAPI クライアントインタフェース

この章では、ACSL S への XAPI クライアントインタフェースについて説明します。

ACSL S サーバーへの XAPI クライアントインタフェース

XML API (XAPI) は、TCP/IP 上の一般的なプロトコルを使用して StorageTek クライアントおよびサーバーが通信できるようにする API です。ACSL S 8.4 以降のリリースは、XAPI のサポートで構成できます。

MVS ベースサーバーを使用するようにリクエストされたクライアントは (現実のテープには HSC、仮想テープには VTCS)、ACSL S を使用して StorageTek ライブラリを管理できるようになりました。

XAPI および XCMD の構成、管理、およびオペレータコマンドと、サポートされる XAPI ユーザーリクエスト、ACSL S XAPI 制御変数の詳細は、ELS ドキュメントセット内の ACSLS サーバーへの *StorageTek Enterprise Library Software XAPI* クライアントインタフェースのドキュメントを参照してください。

ACSL S XAPI サービス

ACSL S XAPI サービスのインストールおよびアンインストールの手順については、『*StorageTek ACSLS 8.4 インストールガイド*』を参照してください。

XAPI コンポーネントは、インストールされると、ACSL S を使用して起動および停止されます。

- XAPI サービスとともに ACSLS を起動するには、Unix コマンドプロンプトで次のように入力します。

```
acsss enable
```

- ACSLS と XAPI サービスを停止するには、Unix コマンドプロンプトで次のように入力します。

```
acsss disable
```

- ACSLS、XAPI サービス、ACSL S データベース、およびすべての ACSLS コンポーネントを停止するには、Unix コマンドプロンプトで次のように入力します。

```
acsss shutdown
```

XAPI の変数

ACSL S の `acsss_config` または `dv_config` ユーティリティーを使用して、ACSL S XAPI の静的変数を表示および設定します。

変更を有効にするには、ACSL S を再起動する必要があります。

- `dv_config`
 - `dv_config -d` はすべての ACSLS 変数を表示します。
 - `dv_config -p <variable_name>` は XAPI 変数を更新します。

[6章「ACSL S の動作を制御する変数の設定」](#)を参照してください。

- `acsss_config`

XAPI 変数を表示および変更するには、オプション 9 を選択します。このオプションは、XAPI サーバーがインストールされている場合にのみ表示されます。

```
ACSL S 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:

XAPI の変数は次のとおりです。

- **XAPI_PORT**

プロンプト: *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]:*

このオプションは、クライアントからの受信 TCP リクエストに XAPI サーバーで使用されるポートを指定します。XAPI サーバーによって使用されるポートを定義する、1024 から 65535 の数値を入力します。ポート 50003 は指定しないでください。

この変数を有効にするには、XAPI サーバーを再起動する必要があります。

- **XAPI_WORK_PATH**

プロンプト: *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]:*

XAPI サーバーの作業ファイルが置かれるディレクトリを選択します。インストールされると、XAPI サーバーは、`$ACS_HOME/log/xapi` ディレクトリに情報を記録します。通常の使用では、この変数の値は変更されません。`$ACS_HOME` が含まれているファイルシステムでディスク容量の問題がある場合は、代替パスを指定できます。指定するパスは絶対パス（または `$ACS_HOME` で始まるパス）にする必要があります。

この変数を有効にするには、XAPI サーバーを再起動する必要があります。

- **XAPI_LOG_SIZE**

プロンプト: *Changes to the XAPI log size will not take effect until the xapi server is restarted. Maximum XAPI log size in Mbytes. [20]:*

このオプションは、M バイト（ここでは「1048576 バイト」と定義）で表される、XAPI ログのしきい値サイズを指定します。負でない数を入力します。20 がこのオプションのデフォルト値です。

この変数を有効にするには、XAPI サーバーを再起動する必要があります。

- `XAPI_LOG_FILE_NUM`

プロンプト: *Number of XAPI Log archive files to retain. [10]:*

このオプションは、保持するアーカイブ XAPI ログファイルの数を指定します。現在の `vlog.file` サイズがしきい値サイズを超えると、0 から `n` の接尾辞が付加されてログファイルの名前が変更します。0 はもっとも新しいファイルで、`n` はもっとも古いファイルです。アーカイブファイルは、`XAPI_WORK_PATH` ディレクトリに保存されます。指定された数のアーカイブログに達すると、新しいファイルがアーカイブディレクトリに追加されるごとに最も古いファイルがそのディレクトリから削除されます。1 つ以上 99 以下のアーカイブファイルを保持できます。1 から 99 の数値を入力して、保持するアーカイブログファイルの数を指定します。

この変数を有効にするには、XAPI サーバーを再起動する必要があります。

- `XAPI_TRACE_SIZE`

プロンプト: *Changes to the XAPI trace size will not take effect until the xapi server is restarted. Maximum XAPI trace size in Mbytes. [50]:*

このオプションは、M バイト (ここでは「1048576 バイト」と定義) で表される、XAPI トレースのしきい値サイズを指定します。負でない数を入力します。50 がこのオプションのデフォルト値です。

この変数を有効にするには、XAPI サーバーを再起動する必要があります。

- `XAPI_TRACE_FILE_NUM`

プロンプト: *Number of XAPI Trace archive files to retain. [10]:*

このオプションは、保持するアーカイブ XAPI トレースファイルの数を指定します。現在の `vtrace.file` サイズがしきい値サイズを超えると、0 から `n` の接尾辞が付加されてトレースファイルの名前が変更します。0 はもっとも新しいファイルで、`n` はもっとも古いファイルです。アーカイブファイルは、`XAPI_WORK_PATH` ディレクトリに保存されます。指定された数のアーカイブログに達すると、新しいファイルがアーカイブディレクトリに追加されるごとに最も古いファイルがそのディレクトリから削除されます。1 つ以上 99 以下のアーカイブファイルを保持できます。1 から 99 の数値を入力して、保持するアーカイブログファイルの数を指定します。

この変数を有効にするには、XAPI サーバーを再起動する必要があります。

- `XAPI_STARTUP_FILE`

プロンプト: *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]:*

このオプションは、XAPI 起動ファイルの名前を指定します。このファイルは、XAPI `_WORK_PATH` ディレクトリに存在し、XAPI 起動パラメータを含みます。

この変数を有効にするには、XAPI サーバーを再起動する必要があります。

- `XAPI_TAPEPLEX_NAME`

プロンプト: *Changes to the XAPI Tapeplex name will not take effect until the xapi server is restarted.XAPI Tapeplex の名前.[]:*

このオプションは、XAPI Tapeplex の名前を指定します。この変数を有効にするには、XAPI サーバーを再起動する必要があります。

長さが 8 文字を超えないテキストを入力してください。

付録Q ACSLS でのアクセシビリティ機能

オラクル社は、障害のあるお客様にも使用していただける高品質の情報テクノロジーを提供するように努力しています。ここでは、ACSLS での具体的なアクセシビリティ機能について要約しています。

ユーザーインターフェース

このセクションでは、ACSLS ユーザーインターフェースについて説明します。

ACSLS (テキストのみ) コマンド行インターフェース (CLI)

ACSLS のプライマリ CLI は、`cmd_proc` と呼ばれます。ACSLS オペレータは、`cmd_proc` からすべての物理ライブラリ操作を行うことができます。これには、次が含まれます。

- マウント、マウント解除、挿入、取り出し、および移動操作などのテープ移動アクティビティ。
- テープドライブおよびテープボリュームの位置とステータス、LSM ステータス、LMU ステータス、および CAP ステータスに対するクエリー操作。
- すべての物理ライブラリリソースをオンラインおよびオフラインに変更する制御操作、および複数のライブラリインターフェース間で切り替える機能 (LMU の切り替え)。
- テープの所有権、プールの管理、ジョブ制御のためのロック機能、クリーニングカートリッジの管理などのテープ管理アクティビティ。

CLI からアクセスできない管理機能には、次のものが含まれます。

- 論理ライブラリの管理: 論理ライブラリを作成したり、テープドライブを割り当ておよび割り当て解除したり、テープボリュームを論理ライブラリに割り当ておよび割り当て解除する機能。
- 論理ライブラリ操作: 論理ボリュームを構成されたドライブにマウントおよびマウント解除する機能。

2 番目の CLI である `lib_cmd` は、論理ライブラリを管理するために ACSLS GUI の代わりに使用されます。`lib_cmd` を使用すると、次のことが可能です。

- 論理ライブラリを作成または削除します。

- 論理ライブラリの属性を編集します。
- ボリュームまたはドライブを論理ライブラリに割り当ておよび割り当て解除します。
- クライアントアプリケーションと論理ライブラリ間のマッピングを定義します。
- 論理ライブラリのステータスを表示します。
- 物理または論理ライブラリ内のボリュームまたはドライブのステータスを表示します。
- 論理ライブラリまたは割り当てられたドライブをオンラインまたはオフラインに変更します。

プライマリ管理インタフェース: Unix シェル機能

ACSLs をインストールして構成したり、ACSLs データベースをバックアップおよび復元したり、補助ユーティリティを操作したりするプライマリ機能には、Solaris の下で標準の UNIX シェル (sh, csh, ksh, および bash) 経由でアクセスできます。

ACSLs グラフィカルユーザーインタフェース (GUI)

ACSLs GUI は、次の 3 つのフレームに分かれています。

- 上部フレームには常に、5 つのタイトル付きナビゲーションボタンを含む ACSLS GUI マストヘッドと、ユーザー ID、サーバー ID、およびダッシュボードレベルのステータスを含むサマリー情報が表示されます。
- いちばん左のフレームには常に、展開可能なナビゲーションツリーが表示されます。
- 右側のフレームに表示される内容は、ユーザーがナビゲーションツリーまたはマストヘッドの「Preferences」ボタンから内容の選択を表明すると動的に変更されます。

Web ベースの GUI を使用すると、ユーザーはすべての論理ライブラリ操作および大部分の物理ライブラリ操作を実行できます。GUI ライブラリ操作には、次のものが含まれます。

- 論理ライブラリを作成したり、ボリュームを論理ライブラリに割り当てまたは割り当て解除したり、ボリューム所有権またはクリーニング属性を設定したりする管理機能。
- マウント、マウント解除、挿入、取り出しなどの物理的および論理的なテープ移動アクティビティ。
- 物理テープドライブおよびテープボリュームの位置とステータス、LSM ステータス、LMU ステータス、および CAP ステータスに対するクエリー操作。
- 論理ライブラリに割り当てられている物理ライブラリリソースに対するクエリー操作。
- 物理および論理ライブラリリソースをオンラインおよびオフラインに変更 (vary) する制御操作。

GUI からアクセスできない管理機能には、次のものが含まれます。

- 複数のライブラリインタフェース間で切り替える機能 (LMU の切り替え)。
- プールの管理や、ジョブ制御のためのロック機能を含む特定のテープ管理アクティビティ。

これらのタスクは、`cmd_proc` インタフェース経由で処理されます。

ACSL S GUI でのアクセシビリティのプログラミング

- 内容のフレームに表示されるすべてのページは、ページの上部にあるタイトルで識別されます。
- GUI のすべてのグラフィカル要素に、テキストと同等のツールチップが用意されています。代替テキストを表示するには、マウスのポインタを対応するアイコンの上に置きます。
- 内容のフレームに表示されるすべてのデータテーブルは、列および行ヘッダーで識別されます。
- 展開可能なナビゲーションツリーは、デフォルトで完全に展開されるように構成できます。
- マストヘッドには、障害のあるステータス状態を示す色分けされたダッシュボード要素が含まれています。色覚障害をお持ちのユーザーは各ステータスアイコンの横を参照できます。数値が 0 を超えている場合は常にここに同じメッセージが示され、同時にこのエラントステータスを共有するライブラリリソースの個数も表示されます。

ACSL S でアクセシビリティモードを構成するために必要な特殊な手順

このセクションでは、ACSL S でアクセシビリティモードを構成するために必要な特殊な手順について説明します。

GUI ツリーメニュー

ACSL S GUI は、展開または縮小されたツリーメニューを備えています。ツリーメニューを縮小することによって、メニューの複雑さが、メニュー部分が選択されると展開される少数の初期選択の後ろに隠されます。メニューのキーボードナビゲーションまたはオーディオサマリーに依存している可能性のある視覚に障害をお持ちのユーザーの場合は、縮小されたツリーメニューによって、GUI の大部分がビューから隠されたままになることがあります。展開されたツリーメニューを有効にするには、次の手順に従います。

1. ACSL S GUI マストヘッドの右上隅にある 3 つのボタンのうちの最初の「Preferences」ボタンを見つけます。「Preferences」ボタンを表明すると、GUI の右側のフレームに「Preferences」ページが表示されます。

2. 「Preferences」フレームの「Per-user Preferences」部分で、「Default Tree Menu」ドロップダウンメニューを見つけます。
3. ドロップダウン矢印を表明し、「Expanded」を選択します。
4. 展開された選択肢が選択されたら、「Preferences」フレームの右上隅にある 2 つのボタンを見つけて「Set」ボタンを表明します。

用語集

アップグレードインストール	既存の顧客サイトで新しいバージョンの ACSLS をインストールするときに実行されます。
移動中のカートリッジ	その元の位置と移動先の位置の間にあるカートリッジ。カートリッジは、パススルーポート、ロボットハンド、またはプレイグラウンド内にある場合に移動中と見なされます。
イベントロガー	「ACS イベントロガー」を参照してください。
イベントログ	ライブラリと ACSLS イベントが記載されたメッセージを含む、ACSEL によって保持されているファイル。
インライン診断	サブシステムコンポーネント内の機能マイクロコードとタイムシェアリングベースで動作している間にサブシステムのコンポーネントをテストするルーチン。
大きな CAP (LCAP)	それぞれ 10 セルの 4 つの取り外し可能マガジンにストレージセルが配置された 40 カートリッジの CAP。これらのマガジンは、ホストソフトウェアには 40 セルの 1 つの列として表示されます。
カートリッジ	一定の長さのデータ記録テープを含むプラスチック製カバー。テープは、トランスポートにロードされると自動的に装着されます。自動装着のために、テープにはプラスチック製リーダーブロックが取り付けられています。カートリッジの背には、ボリューム ID が一覧表示された OCR/バーコードラベルを付けることができます。
カートリッジアクセスポート (CAP)	データまたはクリーニングカートリッジの手動挿入または自動取り出しを可能にする、LSM のドアパネルに組み込まれた双方向のポート。
カートリッジテープ I/O ドライバ	カートリッジサブシステムにコマンド (読み取り、書き込み、巻き戻しなど) を発行するオペレーティングシステムソフトウェア。
カートリッジドライブ (CD)	2 つまたは 4 つのカートリッジドライブとそれに関連付けられた電源、および空圧源を含むデバイス。
カートリッジトランスポート	テープをカートリッジから、テープのデータの書き込み/読み取りを行うヘッドまで移動する電気機械式デバイス。トランスポートは、その動作に必要な電気と空気を供給する電源および空圧源とは区別されます。「カートリッジドライブ」を参照してください。
外部ラベル識別子	物理テープカートリッジを識別するために使用される、カートリッジの外側の端にある 6 文字の英数字ラベル。これは、A - Z の大文字、0 - 9 の数値、\$、#、および空白で構成できます。
開放型相互接続 (OSI)	国際標準化機構 (International Organization for Standardization) のソフトウェアアーキテクチャーモデル。OSI モデルは、データ処理システムの相互接続のための標準を提供します。

仮想ラベル	カートリッジの物理ラベルがないか、または読み取れないときに、そのカートリッジに割り当てることができる論理ボリューム ID (volser)。
監査	ライブラリのすべてまたは一部に含まれる内容の物理的なインベントリ。
クライアントアプリケーション	テープカートリッジの内容を管理するソフトウェアアプリケーション。これらは、ACSL S と対話することによってテープカートリッジにアクセスします。クライアントシステム上に任意の数のクライアントアプリケーションが常駐できます。
クライアントコンピューティングシステム	コンピュータ、およびオペレーティングシステムの実行可能イメージ。
クライアントシステムインタフェース (CSI)	ACS Library Manager と Client System Component の間のメッセージを変換およびルーティングするソフトウェアコンポーネント。
クライアントソフトウェア	このソフトウェアはテープカートリッジの内容を管理し、カートリッジへのリクエストを生成し、さらにカートリッジとの間のデータの送受信を行います。クライアントソフトウェアは、ACSL S には含まれません。
検証エラー	cmd_proc によって実行される形式および構文検証から発生するエラー。
構造化照会言語 (SQL)	データベース内のデータの定義、アクセス、および更新のために使用される言語。
高速ロード	高速ロードが有効になっている場合は、操作が検証されて ACSLS によって受け入れられるとマウント操作を発行した FC イニシエータは正常な応答を受信しますが、それはカートリッジの移動が開始される前です。
コマンドアクセス制御	コマンドへのアクセスを制限します。
コマンドプロセッサ (cmd_proc)	ACSSA の画面インタフェース。cmd_proc を使用すると、第 7 章で説明されているコマンドを入力できます。
コマンド領域	リクエストを入力したり、応答を受信したりする、cmd_proc インタフェースのいちばん下の領域。
コンプレックスパーティション分割	ライブラリコンプレックス内のパススルーポート経由で接続された複数の SL8500 ライブラリにわたるパーティション分割。これには、最大 16 のパーティションのサポートも含まれます。
サーバーシステム	ライブラリの一部であり、ACSL S が存在する場所です。現在は、ライブラリ制御システムと呼ばれます。ライブラリ制御システムは、ライブラリとクライアントシステムの間インタフェースとして機能します。
サーバーシステムユーザー	ACSL S コマンド、ユーティリティー、またはサーバーシステムに関する手順を起動するユーザー。サーバーシステムユーザーは一般に、サイトおよび保守担当者 (ライブラリオペレータ、テープライブラリアン、システム管理者、CSE、システム担当者など) です。

サーボ機構	フィードバックを使用してプロセスを制御するシステム。
サイクルエラーメッセージ	ライブラリまたは ACSLS の障害を示すメッセージ。
再実行ログファイル	ACSLS データベースを復元するために使用されるバックアップファイル。
サイロ	LSM に対して一般的に使用される用語。「ライブラリストレージモジュール」を参照してください。
システムユニット	ライブラリ制御プラットフォーム。
システムリソース変数	ACSLS によって使用されるシステムリソースの量を制御するために使用されます。
自動カートリッジシステム (ACS)	パススルーポート経由で接続された 1 つまたは複数のライブラリから成るライブラリサブシステム。
自動ライブラリ	「ライブラリ」を参照してください。
ジャーナル	最後のチェックポイント以降にデータベースに対して行われた変更の連続したログ。
冗長電子デバイス	オプションの SL8500 冗長電子デバイス (RE) 機能は、エンタープライズライブラリ内のフェイルオーバー保護を提供します。RE では、2 セットのライブラリコントローラカードを使用します。常に、1 つのセットがアクティブであり、もう一方のセットがスタンバイです。アクティブなライブラリコントローラは、ACSLS または SL コンソールからのコマンドに応答して、スタンバイにフェイルオーバーできます。ライブラリカードの障害が発生した場合は、ライブラリで自動フェイルオーバーを開始できます。
初期プログラムロード (IPL)	マシンリセットをアクティブ化し、EPROM の起動時診断を開始したあと、機能コードをロードするプロセス。
処理エラー	処理またはネットワーク通信障害から発生するエラー。
シリアル接続	「HLI 接続」を参照してください。
スクラッチ	空白であるか、または有効なデータが含まれないことを示すテープカートリッジの属性。
ストレージサーバーインタフェース (SSI)	クライアントアプリケーションと CSI の間のメッセージを変換およびルーティングする、クライアントシステム上に常駐するソフトウェアコンポーネント。
制御バスアダプタ	クライアントコンピューティングシステムの制御プロトコルを StorageTek ライブラリ制御システムの制御プロトコルに変換するハードウェアデバイス。
制御ユニット (CU)	チャンネルと最大 15 のカートリッジトランスポートの間で論理的に位置決めされるマイクロプロセッサベースのユニット。CU はチャンネルコマンドをトランスポートコマンドに変換し、チャンネルにトランスポートステータスを送信します。

セル	カートリッジが格納される LSM 内の容器。
チャンネル	ホストとメインストレージを入出力制御ユニットに接続するデバイス。
データパス	クライアントアプリケーションからテープカートリッジへの読み取り/書き込みアクセスを可能にするネットワークパス。
データパスアダプタ	クライアントコンピューティングシステムのデータプロトコルを StorageTek 制御ユニットのデータプロトコルに変換するハードウェアデバイス。
データベース	互いに関連するデータレコードのコレクション。「ACSLS データベース」も参照してください。
テープの開始位置 (BOT)	書き込まれたデータが開始するテープ上の位置。
テープの終端 (EOT)	書き込まれたデータが終了するテープ上の位置。
テープライブラリ管理システム (TLMS)	クライアントアプリケーションのタイプ。
デュアル TCP/IP	ホストソフトウェア (ACSLS または HSC) とライブラリの間の 2 つの個別のホスト接続を提供します。
電氣的に消去できるプログラム可能な読み取り専用メモリー (EPROM)	消去して再プログラミングできる特殊なメモリーチップ。
動的構成	ACSLS がオンラインのまま実行中に ACSLS ライブラリ (およびコンポーネント) への構成の変更を実装できるようにします。
取り出し済みカートリッジ	ライブラリから取り出されたカートリッジ。0 以外の保持期間が設定されている場合、カートリッジステータスは STATUS_VOLUME_EJECTED に変更されます。
ネットワークアダプタ	ネットワークと特定の取り付けられた装置の間の電気および論理インタフェースを提供する装置。
ネットワークインタフェース (NI)	ネットワーク接続を維持し、メッセージの交換を制御する、サーバーシステムとクライアントシステム間のインタフェース。NI は、サーバーシステムと各クライアントシステム上に常駐します。
パーティション	ACSLS により個別の ACS として管理される、物理パーティションに割り当てられたライブラリセル、カートリッジ、ドライブ、および CAP のパーティション。
パススルーポート (PTP)	複数の LSM を備えた ACS 内で、カートリッジをある LSM から別の LSM に渡すことができるようにするメカニズム。
バッキングライブラリ	論理ライブラリの作成元の物理ライブラリを識別します。

発光ダイオード (LED)	少量のエネルギーを使用し、主にオン/オフ状態を示すために使用される発光デバイス。
標準 CAP (SCAP)	7つの固定されたセルの3つの列にストレージセルが配置された21カートリッジのCAP。
表示領域	ライブラリのステータスに関連したメッセージを収集する、cmd_proc インタフェースのいちばん上の領域。
プール	1つまたは複数の類似した機能または属性を持つテープカートリッジのコレクション (スクラッチテープのプールなど)。
複数 TCP/IP	複数のライブラリへの TCP/IP 接続を使用して、ホストソフトウェア (ACSL5 または HSC) と SL8500 ライブラリコンプレックスの間の冗長な通信パスを提供すること。
不在カートリッジ	データベース内に存在するが、カートリッジの記録されたすべての位置がカタログ化されたときに見つからなかったカートリッジ。0 以外の保持期間が設定されている場合、ボリュームステータスは STATUS_VOLUME_ABSENT に変更されます。
フルインストール	新しい顧客サイト、または新しいライブラリがインストールされた既存のサイトに必要な完全なソフトウェアインストール。
プレイグラウンド	電源投入時および LSM の初期化が完了する前に移動中であることが検出された診断カートリッジやカートリッジを格納するために使用される (LSM 内の) 特殊なセルの予約された領域。
ホーム位置	特定のカートリッジに関連付けられたセル。
ホストソフトウェアコンポーネント (HSC)	複数のライブラリをライブラリサーバーとして制御する、IBM メインフレーム上で動作するソフトウェア。
ボリューム	データまたはクリーニングカートリッジ。
ボリュームアクセス制御	ボリュームへのアクセスを制限します (通常はクライアントによって実行されます)。
ボリューム識別子	データまたはクリーニングカートリッジをデータベースに対して一意に識別する 6 文字の文字列。
ボリュームシリアル番号 (volser)	外部ラベル識別子の同義語。
見つからないカートリッジ	データベース内に存在するが、見つからなかったカートリッジ。カートリッジの記録された推定される位置を、オフラインの LSM または通信していないドライブのために検査できなかった場合、そのカートリッジは「ABSENT」ではなく「MISSING」とマークされます。カートリッジステータスは STATUS_VOLUME_MISSING に変更されます。

メディア検証	T10000C および T10000D ドライブの指定されたプールを使用して T10000 カートリッジを検証する機能。この検証では、テストされるテープカートリッジごとに「成功」または「失敗」の結果が出力されます。
ユーザー	選択可能な機能およびオプションの変数 - ユーザーが選択可能なさまざまな機能およびオプションを定義するために使用されます。
優先 CAP (PCAP)	カートリッジの優先挿入および取り出しに使用される 1 つのカートリッジ CAP。
要求外メッセージ	エラーを示すか、または特定のルーチンアクションを実行できることを通知するメッセージ。
ライブラリ	1 つのライブラリは、1 つまたは複数の ACS、装着されたテープドライブ、ACS 内のボリューム、および ACS を制御および管理する ACSLS ソフトウェアで構成されています。
ライブラリエラー	ライブラリがオフラインであるか、ハードウェア障害が発生したか、使用できないなどのために発生するエラー。
ライブラリ管理ユニット (LMU)	ACS の一部であり、LSM を管理し、そのリソースを割り当て、さらに ACSLS と通信します。
ライブラリ構成オプション	顧客はライブラリ内の ACS の数、および各 ACS とサーバーシステムの間の接続を指定できます。
ライブラリストレージモジュール (LSM)	カートリッジ、カートリッジドライブ、CAP、およびロボットの移動に必要なストレージ領域を提供する ACS 構造。
ライブラリ制御コンポーネント	ACS 内のカートリッジのマウントおよびマウント解除を制御するソフトウェア。
ライブラリ制御システム	ライブラリ制御ソフトウェア (ACSL) とともにロードされるライブラリ制御プラットフォーム。
ライブラリ制御ソフトウェア	ライブラリ制御コンポーネント、クライアントシステムインタフェース、およびライブラリユーティリティを含む ACSLS のソフトウェアコンポーネント。
ライブラリ制御プロセッサ	適切なソフトウェアの追加によってライブラリ制御ソフトウェアの操作をサポートする、正しく構成されたコンピュータハードウェア。
ライブラリ制御ユニット	LSM の一部であり、データおよびクリーニングカートリッジのピックアップ、マウント、マウント解除、および交換を制御します。
ライブラリドライブ	クライアントシステムに接続されて制御される、LSM に取り付けられたカートリッジトランスポート。ライブラリドライブは、テープカートリッジ自動マウントおよびマウント解除操作中に LCU と対話します。ライブラリドライブは、テープデータ転送操作中にクライアントアプリケーションと対話します。ライブラリドライブは、ACSLM から個別にアドレス可能であり、クライアントアプリケー

	ションから個別にアクセス可能です。「カートリッジトランスポート」を参照してください。
リレーショナルデータベース	データ項目間の関係に従って構成およびアクセスされるデータベース。関係は、テーブルによって表されます。
ローカルエリアネットワーク (LAN)	ネットワーク内のすべてのコンポーネントがほかのすべてのコンポーネントにアクセスできるコンピュータネットワーク。これは、LMU と取り付けられた LSM の間のインターフェースのタイプです。
ACS	「自動カートリッジシステム」を参照してください。
ACS ID	ACS の一意の識別子。
ACS Library Manager (ACSLM)	ライブラリリクエストおよび応答を検証してルーティングするソフトウェアコンポーネント。
ACS イベントロガー (ACSEL)	ほかの ACSLS コンポーネントからメッセージを受信し、それをイベントログに書き込むソフトウェアコンポーネント。
ACS システム管理者 (ACSSA)	コマンドプロセッサとシステムの残りの部分の間のインターフェース。
ACS ライブラリ	1 つのライブラリは、1 つまたは複数の ACS、装着されたテープドライブ、および ACS 内に存在するカートリッジで構成されています。
ACS ライブラリソフトウェア (ACSL)	複数のライブラリの内容を管理し、ACS テープドライブ上のカートリッジをマウントおよびマウント解除するライブラリハードウェアを制御します。
ACS ライブラリハンドラ (ACSLH)	ACSLM の一部であり、LMU と直接通信します。
ACSEL	「ACS イベントロガー」を参照してください。
ACSLH	「ACS ライブラリハンドラ」を参照してください。
ACSLM	「ACS Library Manager」を参照してください。
ACSL	「ACS ライブラリソフトウェア」を参照してください。
ACSL データベース	データまたはクリーニングカートリッジの位置とステータスに関する情報を含む ACSL データベース。この情報には、セル位置やスクラッチステータスなどが含まれます。
ACSL データベース	ライブラリ構成や、ライブラリ内のすべてのデータまたはクリーニングカートリッジの位置と ID を追跡するために ACSL によって使用されるデータベース。
ACSL プラットフォーム	ACSL のための適切な環境を提供するサーバーハードウェアおよびソフトウェア。

ACSSA	「ACS システム管理者」を参照してください。
ADI	Application Data Interchange (アプリケーションデータ交換)。
BOT	「テープの開始位置」を参照してください。
CAP	「カートリッジアクセスポート」を参照してください。
CAP ID	CAP の位置の一意的識別子。CAP ID は、ACS ID、LSM 番号、および CAP 番号で構成されています。
CCI	「クライアントコンピューティングシステム」を参照してください。
CD	「カートリッジドライブ」を参照してください。
Client System Component	クライアントコンピューティングシステムのオペレーティングシステムと ACSLS の間のインタフェースを提供するソフトウェア。
CSC	Client System Component (クライアントシステムコンポーネント)。
CSE	Customer Services Engineer (顧客サービスエンジニア)。
CSI	「クライアントシステムインタフェース」を参照してください。
CSI 変数	CSC と CSI の間の通信を微調整するさまざまなオプションを定義するために使用されます。これらの変数は、acsss_config プログラムで変更します。
CU	「制御ユニット」を参照してください。
EOT	「テープの終端」を参照してください。
EPO	Emergency Power Off (非常電源切断)。
EPROM	「電氣的に消去できるプログラム可能な読み取り専用メモリー」を参照してください。
HLI	Host/Library Interface (ホスト/ライブラリインタフェース)。ACSLS がライブラリと通信する 1 つの方法。
HLI 接続	HLI 経由で ACSLS に接続されたライブラリ。これらのライブラリは、シリアルインタフェース (シリアル接続) または TCP/IP インタフェース (TCP/IP 接続) 経由で接続できます。
I/O	Input/Output (入出力)。
ID	識別子または識別。
IPC	プロセス間通信。
IPL	「初期プログラムロード」を参照してください。

LAD	Lock Access Door (ロックアクセスドア)。
LAN	「ローカルエリアネットワーク」を参照してください。
LCAP	「大きな CAP」を参照してください。
LCU	「ライブラリ制御ユニット」を参照してください。
LED	「発光ダイオード」を参照してください。
LibAttach	クライアントアプリケーションと CSI の間のメッセージを変換およびルーティングする、Windows NT クライアントシステム上に常駐するソフトウェアコンポーネント。
LMU	「ライブラリ管理ユニット」を参照してください。
LSM	「ライブラリストレージモジュール」を参照してください。
LSM ID	LSM の一意の識別子。LSM ID は、ACS ID と LSM 番号で構成されています。
NI	「ネットワークインタフェース」を参照してください。
OCR	Optical Character Recognition (光学文字認識)。
ONC	Open Network Computing (オープンネットワークコンピューティング)。
OSI	「開放型相互接続」を参照してください。
OSLAN	Open Systems Local Area Network (オープンシステムローカルエリアネットワーク)。
PCAP	「優先 CAP」を参照してください。
POST	Power-On Self-Test (電源投入時自己診断)。
PROM	Programmable Read-Only Memory (プログラム可能な読み取り専用メモリー)。
PTP	「パススルーポート」を参照してください。
RDBMS	Relational Database Management System (リレーショナルデータベース管理システム)。
ROM	Read-Only Memory (読み取り専用メモリー)。
RPC	Remote Procedure Call (リモート手続き呼び出し)。
SCAP	「標準 CAP」を参照してください。

SCSI	Small Computer Serial Interface。
SIMM	Single Inline Memory Module (シングルインラインメモリーモジュール)。
SLOT	「セル」を参照してください。
SQL	「構造化照会言語」を参照してください。
SRN	「サービスリクエスト番号」を参照してください。
SSI	「ストレージサーバーインタフェース」を参照してください。
SSR	Software Support Representative (ソフトウェアサポート担当者)。
StorageTek ライブラリコンソール	StreamLine ライブラリのために使用されるオペレータパネルソフトウェアアプリケーション。
TCP	Transmission Control Protocol (伝送制御プロトコル)。
TLMS	「テープライブラリ管理システム」を参照してください。
TOD	Time Of Day (時刻)。
UDP	User Datagram Protocol (ユーザーデータグラムプロトコル)。
UNIX	最初はベル研究所 (現在の UNIX Systems Laboratories, Inc.) によって開発され、さまざまなコンピュータシステムによって使用されているオペレーティングシステム。
UOC	Usable On Codes (使用可能コード)。
venter	Virtual enter (仮想挿入)。仮想ラベルの付いた、ラベルのないカートリッジの挿入。
volser	Volume Serial Number (ボリウムシリアル番号)。
WTM	Write Tape Mark (テープマーク書き込み)。
XDR	External Data Representation (外部データ表現)。
XML	Extensible Markup Language (拡張マークアップ言語)。Web 上の構造化されたドキュメントまたはデータ、あるいはその両方のための汎用形式。

索引

あ

- アーキテクチャー、CSCI, 621
- アクセシビリティ, 641
- アクセス制御
 - 有効化
 - ownership.assignments ファイル, 123
- アクセス制御情報、再作成, 107
- アクセス制御の有効化、ownership.assignments ファイル, 123
- アクセス制御の呼び出し、ownership.assignments ファイル, 123
- アドレスの変換, 471, 509
- アドレス、変換, 471, 509
- イベント通知, 140
- イベント通知の設定, 107
- イベント通知への登録, 140
- イベントログ, 90
- イベントログ、検索、greplog の使用, 189
- インポート
 - db_import.sh, 233
 - ディスクファイルからのデータベース, 195
- インポートされたデータベースの検証, 201
- エクスポート
 - ディスクファイルへのデータベース, 192
 - テープへのデータベース, 193

か

- カートリッジ
 - 回復, 177
 - 期限切れ, 180
 - スクラッチ, 169
 - ステータスレポート, 176
 - 挿入, 153
 - 取り出し, 155
 - 取り出し済み, 175
 - 不在, 175
 - ポリシーの適用, 160
 - 無効になっている LSM へのロード, 187
- カートリッジアクセスポート
 - カートリッジの挿入/取り出し, 481
- カートリッジの挿入, 153
- 回復
 - 障害の発生したサーバーから, 206

- 拡張ストア, 128, 583
- 拡張ストア機能, 128
- 仮想テープライブラリ, 531
 - ACSLS に対する構成, 533
 - サポートされている構成, 531
 - 動作, 532
- 監査、実行するタイミング, 126
- キーボードショートカット、cmd_proc, 47
- 期限切れのカートリッジ, 180
- クラスタ化
 - カートリッジ, 482
- クリーニングカートリッジ, 373
 - チェック, 341
- クリーニングテープのパス、手動モード, 165
- 構成
 - CSI チューニング変数の設定, 86
 - アクセス制御情報の再作成, 107
 - アクセス制御変数の設定, 104
 - 一般的な製品動作変数の設定, 94
 - イベント通知設定の定義, 107
 - イベントロギング変数の設定, 90
 - 更新, 74
 - 自動バックアップ変数の設定, 106
 - テープドライブ
 - レイアウト, 472
 - メニュー, 84
 - ライブラリハードウェアの構成, 108
- 高速ロード, 544
- コマンド構文, 281
- コマンドプロセッサ
 - 終了, 46
 - 出力のリダイレクト, 49
 - 出力ファイル, 48
 - 出力ファイルの使用, 49
 - 使用, 44
 - ショートカットキー, 47
 - 入力および出力のリダイレクト, 48
 - 入力ファイル, 48
 - バッチユーザーインタフェース機能, 48
 - 出力のリダイレクト, 48
- コマンド、ACSLs, 279
- 混合メディアライブラリの管理, 130
- 混在メディア
 - mount, 330
 - SCSI 接続 LSM の制限事項, 131
 - スクラッチ優先順位, 132

ファイル, 131
スクラッチ優先順位, 132
メディアラベル, 396
ライブラリ、管理, 130

さ

サーバー
idle, 318
開始, 382
障害回復, 206
サーバー状態, クエリー, 364
サーバーユーティリティー、一般情報, 215
最適化ガイドライン
カートリッジ, 480, 481
テープカートリッジ, 480, 481
メディア, 480, 481
サポート
get_diags, 570
自動データベースバックアップ, 202
手動の挿入, 313
手動ボリューム削除ユーティリティー, 180, 234
冗長電子装置, 463, 463
query lmu 内, 345
switch lmu 内, 383
スクラッチ
カートリッジ最適化, 482
スクラッチ, クエリー, 363
スクラッチ情報の表示, 170
スクラッチプール
削除, 301
スクラッチのマウント, クエリー, 353
定義, 299
表示, 422
リバランス, 171
スクラッチプールの削除, 172
スクラッチプールのリバランス, 171
スクラッチボリュームの管理, 169
スクラッチボリュームの追加, 171
スクラッチボリュームのマウント, 173
静的変数
acsss_config の使用, 85
CSI の設定, 86
一般的な製品動作の設定, 94
イベント通知の設定, 107
イベントロギングの設定, 90
挿入

カートリッジ, 480, 481
最適化, 481
その他の ACSLS ファイル, 回復, 206
その他の ACSLS ファイルの復元, 206

た

チューニング変数, 86
データベース
UNIX ファイルへのバックアップ, 204
インストール, 190
インポート, 194, 447
エクスポート, 190
回復, 204
回復と復元, 204
検証, 201
検証とライブラリ構成, 201
最新のバックアップへの復元, 205
指定されたテープデバイスへのバックアップ, 203
自動バックアップ, 202
テープへの手動のバックアップ, 202
ボリュームの削除, 234
データベースのインポート, 194, 447
データベースのエクスポート, 190
データベースの復元
最新のバックアップへの, 205
テープドライブの移動、追加、または取り外し時の
ACSLs の再構成, 143
デュアル LAN クライアント構成, 137
デュアル LMU, 345
管理, 135
デュアル LMU 構成, 135
動的構成
config acs, 225
config drives, 228
config lsm, 229
config ports, 231
動的構成ユーティリティー, 223
動的変数
CSI の設定, 86
アクセス制御の設定, 104
一般的な製品動作の設定, 94
イベント通知の設定, 107
イベントロギングの設定, 90
自動バックアップの設定, 106
ドライブ
clear lock, 297

- display, 409
- lock, 321, 349, 375
- unlock, 386
- 構成, 228
- 状態, 343
- ステータス, 343
- トラブルシューティング
 - ACSLS イベントログ, 561
 - GUI, 574
 - pinglmu, 567
 - クライアントの接続, 569
 - クリーニングされていないドライブ, 167
 - データベースの回復, 204
 - ライブラリの接続, 566
- トランスポート
 - SCSI 接続 LSM のクリーニング, 166
 - クリーニング, 161
 - 手動でクリーニング, 165
 - 表示, 409
- トランスポートのクリーニング, 161
 - 手動, 165
- 取り出し済みカートリッジ, 175, 179
- 取り出し済みボリューム, 155, 179
- な**
 - 内部アドレス
 - 説明, 469, 508
 - ネットワーク IP アドレス、プライマリおよびセカンダリ LAN の登録, 138
- は**
 - パーティション分割, 549
 - ガイドライン, 551
 - パーティション ID の変更, 553
 - パーティション分割されている ACS をパーティション分割されていない ACS に変更する, 555
 - ライブラリ, 553
 - ライブラリの再パーティション分割, 554
 - パススルー, 581
 - バックアップ
 - ACSLS サーバーに接続された指定されたテープデバイス, 203
 - UNIX ファイル, 204
 - 自動, 202
 - テープへの手動, 202
 - バックアップおよび回復ツール
 - db_import.sh の使用, 450
 - rdb.acsss の使用, 450
 - 自動バックアップ, 447
 - 手動でのデータベースのエクスポート, 448
 - 手動バックアップ, 448
 - 障害のシナリオ
 - サーバー障害 – 新しいハードウェアでの同じサーバーの再構築, 452
 - サーバー障害 – 新しいハードウェアでの別の ACSLS サーバーの再構築, 452
 - データベースが破損した - データベースを稼働状態に復元する必要がある, 451
 - 間違ったライブラリに対して acsss_config を実行した, 452
 - パネル, 行, 列アドレス指定, 469, 509
 - ファイアウォールセキュリティー
 - ACSLS, 591
 - RPC の問題, 592
 - オフにする, 610
 - オンにする, 607
 - シナリオ, 601
 - セキュリティーの問題, 592
 - 通信, 612
 - 通信の問題, 592
 - 変数の設定, 595
 - 利点, 593
 - 復元, 204
 - その他の ACSLS ファイル, 206
 - 不在カートリッジ, 175
 - 不在ボリューム, 177
 - フロート, 128, 253, 480, 583, 583
 - 最適化, 482
 - ポート
 - 関する情報, 424
 - 追加, 231
 - ポートの検証, 424
 - ボリューム
 - display, 426
 - 削除, 180
 - スクラッチ解除, 174
 - スクラッチ情報の表示, 170
 - スクラッチの管理, 169
 - スクラッチの追加, 171
 - スクラッチのマウント, 173
 - 取り出し, 155
 - 不在, 175, 176

ボリューム, クエリー, 366
ボリューム属性, 194
ボリューム統計, 90
ボリューム統計レポート
 ボリューム移動のロギング, 212
ボリュームの移動, 334
ボリュームのスクラッチ解除, 174
ボリュームレポート、カスタム, 210

ま

マウントおよびマウント解除のキューへの追加, 142
マウントおよびマウント解除の再試行, 142
マルチホームホスト, 138
見つからないカートリッジ, 176, 178
無効になっている LSM へのカートリッジのロード,
187
メディア管理, 625, 630
メディア検証, 145
メディアの最適化, 480, 481

や

ユーザー, 表示, 380
ユーティリティ
 acs_renumber.sh, 217
 acsss, 219
 bdb.acsss, 221
 config acs, 225
 config drives, 228
 config lsm, 229
 config ports, 231
 db_export.sh, 232
 db_import.sh, 233
 del_vol, 234
 drives_media.sh, 236
 ejecting.sh, 237
 free_cells.sh, 241
 get_license_info, 246
 getHba.sh, 244
 greplog, 246
 lib_type.sh, 248
 licensekey.sh, 248
 moving.sh, 248
 pinglmu, 567
 probeFibre.sh, 258, 568
 probescsi.sh, 568
 rdb.acsss, 259

showDevs, 568
showDrives.sh, 264
stats_report, 265
testlmu, 567
testlmutcp, 567
testports, 565, 566
userAdmin.sh, 267
volrpt, 269
watch_vols, 275
よくある質問, 614

ら

ライブラリサーバーユーティリティ、一般情報, 215
ライブラリの監査, 125
 監査間隔, 127
 監査の実行, 126
ライブラリハードウェア
 構成, 71
リクエスト, クエリー, 361
リクエスト処理の中断, 49
論理ライブラリ, 537
 コマンド行インタフェース, 435
 削除, 544
 作成, 539
 制限, 539
 利点, 537

A

ACS の状態, 338
ACSLs
 audit, 125
 SCSI 接続 LSM の混合メディアの制限事項, 131
 アイドル状態, 49
 拡張ストア機能, 128
 コマンド, 279
 混合メディアのサポート, 130
 再起動, 50
 再構成
 一般的な製品動作, 94
 ロギングオプション, 90, 104, 106
 スクラッチ優先順位, 132
 ディレクトリ構造, 50
 データベースの回復, 204
 デュアル LAN クライアント構成, 137
 デュアル LMU 構成, 135
 ユーティリティ, 215

リクエスト処理の中断, 46

ACSLs コマンド

audit, 285
cancel, 292
clear lock, 297
define pool, 299
delete pool, 301
dismount, 303
eject, 308
idle, 318
lock, 321
logoff, 323
mount *, 328
mount, 324
move, 334
query acs, 337
query cap, 339
query clean, 341
query drive, 343
query lmu, 345
query lock, 349
query lsm, 351
query mount *, 356
query mount, 353
query pool, 358
query port, 360
query, 336
query request, 361
query scratch, 363
query server, 364
query volume, 366
set cap mode, 369
set cap priority, 371
set clean, 373
set lock, 375
set owner, 377
set scratch, 378
set コマンド, 368
show, 380
start, 382
switch lmu, 383
unlock, 386
vary, 388
venter, 395
挿入, 313

ACSLs のアイドルリング, 49

ACSLs の再起動, 50

ACSLs の再構成

 ログインオプション, 90, 105, 107

ACSLs ユーティリティ

acs_renumber.sh, 217
acsss, 219
bdb.acsss, 221
config acs, 225
config drives, 228
config lsm, 229
config ports, 231
db_export.sh, 232
db_import.sh, 233
del_vol, 234
drives_media.sh, 236
ejecting.sh, 237
free_cells.sh, 241
get_license_info, 246, 246
getHba.sh, 244
greplog, 246
moving.sh, 248, 248
probeFibre.sh, 258
rdb.acsss, 259
showDevs.sh, 264
showDrives.sh, 264
stats_report, 265
userAdmin.sh, 267
volrpt, 269
watch_vols, 275

acssa

 リモートログイン, 47

acssa ユーザー ID, 43

acsss_config, 84, 621

 自動クリーニングの有効化または無効化, 167

 統計情報レポート, 266

 ドライブ構成を動的に変更する, 228

 ボリューム統計レポートのログイン, 212

 メニュー, 75, 85

 モジュールの追加、取り外し、またはスワップ後に
 実行, 521, 529

acsss コマンド, 219

acsss ユーザー ID, 43

AEM, 509

audit コマンド, 285

B

bdb.acsss ユーティリティ, 221

C

cancel コマンド, 292

CAP

cap_state, 340

set mode, 369

一括, 474

状態, 150

タイプ, 148

に関する情報の表示, 153

モード, 150

優先順位, 152

優先度の設定, 371

CAP 情報の表示, 150

CAP の使用, 148

CAP のロード, 148

cell, display, 407

checkGui.sh, 573

clear lock コマンド, 297

cmd_proc

logoff, 323

終了, 46

出力のリダイレクト, 49

出力ファイル, 48

出力ファイルの使用, 49

使用, 44

ショートカットキー, 47

入力および出力のリダイレクト, 48

入力ファイル, 48

バッチユーザーインタフェース機能, 48, 48

出力のリダイレクト, 48

入力ファイルの使用, 48

cmd_proc の開始, 47

cmd_proc の起動, 47

cmd_proc の終了, 46

cmd_proc の使用, 44

cmd_proc の中断および再開, 46

cmd_proc の入力および出力のリダイレクト, 48

config acs, 225

config drives ユーティリティ, 228

config lsm, 229

CSCI

アーキテクチャー, 621

エラーメッセージ, 622

概要, 621

環境変数, 623

D

db_export.sh ユーティリティ, 190, 232

db_import.sh ユーティリティ, 233

define pool コマンド, 299

del_vol ユーティリティ, 234

delete pool コマンド, 301

diagnostics

get_diags, 570

dismount コマンド, 303

display cap コマンド, 403

display cell コマンド, 407

display drive コマンド, 409

display lock コマンド, 415

display lsm コマンド, 416

display panel コマンド, 420

display pool コマンド, 422

display port コマンド, 424

display volume コマンド, 426

display コマンド, 400

display cap, 403

display cell, 407

display drive, 409

display lock, 415

display lsm, 416

display panel, 420

display pool, 422

display port, 424

display volume, 426

display コマンドオプションの使用, 400

display コマンドリファレンス, 399, 435

E

eject コマンド, 308

enter コマンド, 313

enter コマンド, 手動と自動, 315

G

greplog, 189

GUI

開始, 60

トラブルシューティングのヒント, 574

H

HLI-PRC, の説明, 469, 509

I

idle コマンド, 318

IP アドレス、プライマリおよびセカンダリ LAN の登録, 137

L

lib_cmd, 435, 435

LMU, スイッチ, 383

lock

clear, 297

コマンド, 321

lock コマンド, 321

logical libraries

lib_cmd, 435

logoff コマンド, 323

LSM

display, 416

query, 351

設定, 148

M

mount * コマンド, 328

mount コマンド, 324

move, 334

move コマンド, 334

moving.sh, 248

P

pool

define, 299

display, 422

query, 358

削除, 301

Q

query acs コマンド, 337

query cap コマンド, 339

query clean コマンド, 341

query drive コマンド, 343

query lmu コマンド, 345

query lock コマンド, 349

query lsm コマンド, 351

query mount * コマンド, 356

query mount コマンド, 353

query pool コマンド, 358

query port コマンド, 360

query request コマンド, 361

query scratch コマンド, 363

query server コマンド, 364

query volume コマンド, 366

query コマンド, 336

R

rdb.acsss, 259

rdb.acsss ユーティリティ, 259

S

SCSI 接続

追加, 128

SCSI 接続 LSM のトランスポートのクリーニング, 166

set cap mode コマンド, 369

set cap priority コマンド, 371

set clean コマンド, 373

set lock コマンド, 375

set owner コマンド, 377

set scratch コマンド, 378

show コマンド, 380

SL3000

ACSL S サポート, 503

ACSL S への接続, 504

CAP ID を別名処理する, 511

CAP の動作, 513

CAP 番号付け, 506

新しいカートリッジアドレスの監査, 515

新しいパネルタイプ, 508

オフラインに変更する, 514

監査, 505

構成する前に, 504

動的構成を使用する, 515

内部アドレス指定, 508

見つからないカートリッジを見つける, 513

モジュール, 507

SL3000 でのデュアル TCP/IP のサポート, 516

SL500

ACSL S 接続, 517, 523

CAP の動作, 520, 526

アドレススキーム, 518, 524

ドライブのアドレス, 518, 525

パネル, 518, 524
行番号, 524
ライブラリ構成可能設定と ACSLS, 520, 526
ライブラリの監査, 521, 528
ライブラリの相違点, 517, 523
列番号, 518, 524

SL8500

ACSLS HA, 43
CAP, 473
CAP の動作, 513
SL8500 コンポーネントが動作していることを確認する, 468, 505
エレベータおよび PTP アクティビティを最小限に抑える, 480
オフラインに変更する, 482, 514
カートリッジの場所の管理, 481
拡張, 486
 監査, 489
テープドライブの場所, 472
テープのワークロードのサポート, 480
デュアル TCP/IP, 454, 516
内部アドレスと ACSLS アドレス, 469
内部アドレスの変換, 471
パーティション分割, 550
パススルーポート, 491
 ACS の分割, 499, 499
 ACS のマージ, 497
 PTP の取り外し, 499
 新しい SL8500 の追加, 493
左側に新しい SL8500 を追加する, 493
複数の SL8500 への接続, 462, 468
右側に新しい SL8500 を追加する, 494
見つからないカートリッジ, 482, 513
SL8500 でのデュアル TCP/IP サポート, 454
Solaris
 データベースのバックアップとエクスポート, 190
Solaris のインストール、準備
 Solaris 8, 111
start コマンド, 382
stats_report ユーティリティ, 265
switch lmu コマンド, 383

T

TCP/IP のタイムアウト, 137

U

UNIX ファイルのバックアップ, 204
unlock コマンド, 386
userAdmin.sh, 267

V

vary コマンド, 388
venter コマンド, 395
volrpt ユーティリティ, 269

W

watch_vols ユーティリティ, 275