StorageTek Automated Cartridge System Library Software

Quick Reference

Release 8.5

E96383-01

January 2019

ORACLE

StorageTek Automated Cartridge System Library Software Quick Reference, Release 8.5

E96383-01

Contents

Preface

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

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

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

Documentation Accessibility

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

Access to Oracle Support

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

1

Quick Reference

Throughout this quick reference, underlines show valid command and keyword abbreviations. For example, `aud` is an abbreviation of the audit command. Brackets [] enclose optional parameters. A vertical bar (|) separates parameter choices.

Start and Stop Commands

The `acsss` command is used to start, stop, and monitor the status of the various services associated with ACSLS.

- `acsss enable`

  This is the default method to bring up ACSLS. It checks for dependencies and activates, in the proper order, the five ACSLS services and the ACSLS GUI. When this method is used, the services are configured to restart automatically after a system reboot.

- `acsss temp-enable`

  Same as `acsss` enable but services are not restarted after a system reboot.

- `acsss maint-enable`

  Intended for general maintenance operations not involving the ACSLS database. This option enables the GUI infrastructure allowing GUI users to remain logged in while ACSLS is disabled. This method is used in contexts of minor software patch installations. Neither the acsls nor the smce service is enable.

- `acsss disable`

  This is the default method used to halt ACSLS operation. It is not a complete shutdown and allows for the database and any GUI login sessions to remain active for maintenance operations after the acsls and smce services have been disabled. The resulting state is identical to that of `acsss maint-enable`. This is the safest method to bring down the server since ACSLS and the library are placed in an idle state before the services are disabled.

- `acsss force-disable`

  Same as `acsss disable` but the operation does not wait for an idle state before disabling acsls and smce.

- `acsss shutdown`

  This renders a complete shutdown of all ACSLS services. It is intended for contexts of software installation and de-installation, and other maintenance contexts that require the database (`acsdb`) or the GUI infrastructure (`rmi-registry` and `surrogate`) to be shutdown.

- `acsss db`

  This is the preferred control mode to use for database maintenance operations including `db_export`, `db_import`, and `acsss_config`. It enables the ACSLS database engine and disables all other ACSLS services including the ACSLS GUI.

- `acsss status`

  Provides a quick status report of the various ACSLS services.

- `acsss a-status`

  Returns the operational status of the `acsdb` service.

- `acsss d-status`

  Returns the operational status of the `acsls` service.

- `acsss g-status`

  This option displays the status of the ACSLS GUI.

- `acsss l-status`

  Provides a verbose status summary of the various ACSLS services and includes pointers to log data for further analysis in troubleshooting contexts. The logs to which it points are helpful in contexts where the service failed to start up or shutdown.

- `acsss p-status`

  Similar to `acsss status`, this report includes a listing of the various process IDs that are monitored by each respective service contract.

- `acsss w-status`

  Shows the status of the weblogic service.

- `acsss timeout`

  Updates the SMF start-up timeout for acsls.

- `acsss legal`

  This option displays the ACSLS Legal Notice in English or French.

Command Identifiers

Each identifier argument with `cmd_proc` corresponds to a type and consists of one or more components separated by commas.

- `acs_id`

  acs(0-31)

- `cap_id`

  acs(0-31),lsm(0-99),cap(0-11)

  An asterisk (*) in a `cap_id` does the following:

  – acs,lsm,* — causes ACSLS to select the highest priority available CAP in the LSM.

  – acs,* — causes ACSLS to select the highest priority available CAP in the ACS

  – * — for an enter request causes ACSLS to select the CAP in the ACS with the most free cells.

  - * — for an eject request causes ACSLS to select the highest priority CAP in each ACS with a volume designated for ejection.

- `cell_id`

  acs(0-31),lsm(0-99),panel(0-50),row(0-41),column(0-23)

- `drive_id`

  acs(0-31),lsm(0-99),panel(0-50),drive(0-31)

- `drive_type`

  Up to 10 characters drive type identifier; can be any combination of numbers (0-9) or letters (A-Z).

- `lock_id`

  decimal number (0-32767)

- `lsm_id`

  acs(0-31),lsm(0-99)

- `media_type`

  Up to 10 characters media type identifier; can be any combination of numbers (0-9) or letters (A-Z). Spaces are not allowed. A common media type is the STK1R.

- `owner_id`

  volume owner

- `panel_id`

  acs(0-31),lsm(0-99),panel(0-50)

- `pool_id`

  decimal number (0-65535) Specifying an asterisk (*) for the *pool_id* reassigns a volume to its current *pool_id*

- `port_id`

  acs(0-31),port(0-15)

- `request_id`

  unique decimal number (0-65535) assigned by the ACSLS.

- `subpanel_id`

  acs(0-31),lsm(0-99),panel(0-50),startrow(0-41),startcolumn(0-23),endrow(0-41),endcolumn(0-23)

- `vol_id`

  Six-character identifier consisting of any combination of numbers (0-9), letters (A-Z, a-z, or mixed case (except for use in volrpt)), dollar sign ($), pound sign (#), and leading and/or trailing spaces ( ). Use single or double quotes to enclose *vol_ids* with leading or trailing spaces. *Do not* specify *vol_ids* with embedded spaces.

- `volrange`

  Specifies an ascending range of volumes separated by a dash.

  For volranges in query, enter, and eject commands:

If it is a numeric range, specify only the right most numeric portions of the *vol_ids* as the range. All preceding characters *must* be identical. The display commands support full alphanumeric volranges and allow wildcards '*' and '_'.

Auditing the Library

Helpful `cmd_proc` operations:

- `audit cap_id server`

  Audits the entire library; updates library configuration.

- `audit cap_id acs acs_id`

  Audits an ACS.

- `audit cap_id lsm lsm_id`

  Audits an LSM.

- `audit cap_id panel panel_id`

  Audits an LSM panel.

- `audit cap_id subpanel subpanel_id`

  Audits an LSM subpanel.

Configuration

Helpful Unix operations include:

- `acsss_config`

  Runs the configuration script.

- `dv_print`

  Displays values of dynamic options.

- `dv_config -p <variable_namme> -u`

  To prompt for and update a single variable.

- `dv_config -d`

  Displays values of dynamic and static options.

Configuration - Dynamic

Helpful Unix operations include:

ACS

- `config acs new`

  Adds a new ACS.

- `config acs acs_id`

  Reconfigures an existing ACS.

Drives

- `config drive(s) panel_id`

Reconfigures all drives on an existing drive panel. This includes adding drives, updating drive types and serial numbers for existing drives, and deleting drives that were removed from the database.

LSMs

- `config lsm lsm_id`

  Reconfigures an existing LSM and all its components, which include CAPs and panels.

  Note: Use config acs to add or delete an LSM in an ACS

Ports

- `config port(s) acs_id`

  Reconfigures port connections to an ACS.

Displaying Status

- Display CAP information

```
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 ] ]
```

- Display cell information

```
display cell cell_loc ...
[ -status cell_status ... ]
[ [ -c ] | [ -f field ... ] [ -s sort_field ... ] [ -n n ] ]
```

- Display drive information

```
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 ] ]
```

- Display lock information

```
display lock lock_id ...
[ -user user_id ... ]
[ [ -c ] | [ -f field ... ] [ -s sort_field ... ] [ -n n ] ]
```

- Display LSM information

```
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 ] ]
```

- Display panel information

```
display panel panel_id ...
[ -type panel_type ... ]
[ [ -c ] |[ -f field ... ] [ -s sort_field ... ] [ -n n ] ]
```

- Display pool information

```
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 ] ]
```

- Display port information

```
display port port_id ...
[ -online | -offline ]
[ -name port_name ... ]
[ [ -c ] | [ -f field ... ] [-s sort_field ... ] [ -n n ] ]
```

- Display volume information

```
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]
```

Maintaining the Database

- `db_export.sh -f [ db_file | tape_device ]`

  Exports database table data and ACSLS control database files to tape or a file. Use when reinstalling ACSLS or upgrading to a new ACSLS version using the same database.

- `db_import.sh -f [ db_file | tape_device ]`

  Imports database table data and ACSLS control database files from the export tape or file. Use when reinstalling ACSLS or upgrading to a new ACSLS version using the same database.

- `bdb.acsss -f [ backup_file | tape_device ]`

Backs up the database. The command, `bdb.acsss`, with no argument places the backup file in the default backup location (typically `/export/backup/`).

- `acsss db`

  Brings up the database

- `rdb.acsss`

  Recovers the database after a database failure

Managing CAPS

Helpful `cmd_proc` operations.

- `query cap cap_id ... | all` *or* `display cap cap_id ...|*`

  Displays CAP status

- `set cap mode manual | automatic cap_id`

  Sets CAP's entry mode (manual or automatic)

- `set cap priority cap_priority cap_id`

  Sets CAP's automatic selection priority

- `enter cap_id`

  Makes manual mode CAP ready to enter labelled carts

- `enter lsm_id`

  Makes multiple CAPs in an LSM ready

Managing Redundant Electronics

- `query lmu acs_id ... | all`

  Displays all connections to a library, or libraries, for either a single ACS or the ACSLS server.

- `switch lmu acs_id`

  Manually switches ACS management from the SL8500s active HBCR card to the standby HBCR card.

Managing Locks

Helpful `cmd_proc` operations:

- `set lock lock_id`

  Sets your lock ID.

- `show lock | user`

  Displays your current lock ID or user ID.

- `lock drive | volume identifier`

  Locks a volume or drive (to your current lock ID).

- `unlock drive | volume identifier ... | all`

  Removes active locks (to your current lock ID) on specified drives or volumes or all active locks.

- `clear lock drive | volume identifier`

  Removes all active and pending locks on specified drives or volumes.

Managing Logical Libraries

The `lib_cmd` command-line utility is an alternative to the ACSLS GUI for managing and monitoring resources in ACSLS logical libraries.

- `lib_cmd assign drive drive_id lib_id`

  Adds the specified drive to the logical library.

- `lib_cmd unassign drive lib_id logical_drive_id`

  Removes the specified logical drive from the logical library configuration.

- `lib_cmd assign volume  [ vol_id | vol-range ] lib_id`

  Assigns a volume for exclusive use within a logical library.

- `lib_cmd assign volume  [ vol_id | vol-range ] lib_id`

  Removes a volume from the logical library inventory.

- `lib_cmd create library lib_name backing_acs cell_capacity drive_ capacity`

  Creates a new logical library.

- `lib_cmd create mapping initiator_id target_id library_id`

  Establishes an initiator-target (client-server) relationship for the specified library.

- `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 ]`

  Alters the configuration of an existing logical library. Options are:

  – -n library alias name
  – -c logical storage cell capacity
  – -d number of logical drive slots
  – -f volume label format
  – -x number of logical import/export (CAP) cells

  Volume label format may be expressed with any of the following:

  – 6 - six character legacy volume labels
  – 8p - eight character prefixed with media type code
  – 8s - eight character suffixed with media type code
  – all - accept all label formats

- `lib_cmd refresh initiator`

  Retrieves all initiators currently known to ACSLS.

- `lib_cmd refresh target`

  Retrieves all targets currently known to ACSLS.

- `lib_cmd delete initiator initiator_id (y/n)`

Removes the specified initiator from the configuration. Confirmation (y) is required in order to commit the specified deletion.

- `lib_cmd delete library lib_id (y/n)`

  Removes the specified logical library from the ACSLS configuration. This creates a cascading delete, disassociating any assigned volumes and drives from the library.

- `lib_cmd delete mapping lib_id (y/n)`

  Removes all initiator-target mappings from the specified logical library. The operation disables any current client connections.

- `lib_cmd delete target target_id (y/n)`

  Removes the specified target from the configuration. Confirmation (y) is required in order to commit the specified deletion.

- `lib_cmd display drive [drive_id ][-p acs_id | all ][-l lib_id | all ][-t drive_type ][-u acs_id ]`

  Generates a summary of the specified drive or all drives in the specified library.

  - If 'display drive all' is submitted without options, the utility generates a listing of all volumes contained in each physical ACS.
  - If the '-t' option is passed, only drives of the specified drive type are displayed.
  - The '-p' option limits the display to drives in the specified physical ACS.
  - If the '-l' option is included, then only the drives assigned to the specified logical library are displayed. If "all" is specified following the '-l' option, then a display is generated showing all of the drives associated with each of the configured logical libraries.
  - The '-u' option displays only unassigned drives in the associated ACS. This option must be followed with an argument specifying the desired ACS

- `lib_cmd display initiator`

  Generates a list of all initiators identified by ACSLS. Each initiator is listed by its world-wide name and the alias name.

- `lib_cmd display library [ -p acs_id | all ] [ -l lib_id | all ]`

  Displays the attributes of the requested library ID.

  - l - logical libraries
  - p - physical libraries

- `lib_cmd display mapping [ lib_id | all ]`

  Generates a list of initiator-target mappings for each library ID (or all libraries).

- `lib_cmd display target`

  Generates a list of all targets identified by ACSLS.

- `lib_cmd display volume [ vol_id  | vol-range] [ -p acs_id | all ][ -l lib_id | all ][ -u acs_id ]`

  Generates summary information for the specified volume or volume set.

- `lib_cmd vary library lib_id [online|offline|diagnostic]`

  Changes the desired state of the logical drive to the specified (online, offline, diagnostic) state.

- `lib_cmd vary library lib_id drive_id [online|offline|diagnostic]`

  Changes the desired state of the logical library to the specified (online, offline, diagnostic) state.

- `lib_cmd [ exit | quit | log ]`

Managing Pools

Helpful `cmd_proc` operations:

- `define pool low_water_mark high_water_mark pool_id ...[overflow]`

  Creates or modifies scratch pools

- `query pool pool_id ... | all`

  Displays scratch pool attributes

- `query scratch pool_id ... | all`

  Displays the status of scratch volumes in a pool

- `set scratch pool_id vol_id | volrange`

  Sets volume's scratch attribute and assign the volume to a scratch pool

- `set scratch off pool_id vol_id | volrange`

  Changes volume from scratch to data

- `delete pool pool_id ... | all`

  Deletes an empty scratch pool

- `mount * drive_id pool_id`

  Mounts a scratch volume from a specified pool (single media libraries)

- `mount * drive_id`

  Mount a scratch volume from the common pool (single media libraries)

- `mount * drive_id pool_id media media_type`

  Mounts a scratch volume from a specified pool with specific media type

- `mount * drive_id pool_id media *`

  Mounts a scratch volume from a specific pool, media type based on scratch preferences defined

- `mount * drive_id media *`

  Mounts a scratch volume from common pool, media type based on defined scratch preferences

- `mount * drive_id media media_type`

  Mounts a scratch volume from common pool with specified media type

- `display pool pool_id ...| *`

  Displays scratch pool information for a specific pool or for all pools

- `query mount * pool_id ... [media media_type | media *]`

  Displays status of media-compatible drives for a specified scratch pool (or volume media type within the pool)

Helpful Unix operations include:

- `watch_vols [start|stop]`

  Reviews pre-defined policies for volumes that are: newly entered, discovered by audit or cartridge recovery, or re-activated by audit/cartridge recovery/an enter.

Managing Volumes

Helpful `cmd_proc` operations include:

- `mount vol_id drive_id [bypass] [readonly]`

  Mounts a data volume or cleaning cartridge.

- `dismount vol_id drive_id [force]`

  Dismounts a data volume or cleaning cartridge.

- `display volume vol_id | vol_range | *-clean`

  Displays volume information for cleaning cartridges.

- `display volume * [-media media type] -f media end_of_life warranty_life -s end_of_life`

  Displays volume end of warranty and end of life percentages, sorted by end of life.

- `eject cap_id vol_id | volrange ...`

  Ejects volumes from the library.

- `move vol_id lsm_id`

  Moves volumes to a specified LSM.

- `set clean max_usage | vol_id | volrange`

  Sets cleaning cartridge attributes.

- `set clean off vol_id | volrange`

  Sets cleaning attributes back to data cartridges.

- `volrpt [-s vol | loc | use] [-d] [-f filename][-z] [-a | -l | -v identifier_list]`

  Creates a volume report.

Helpful Unix operations include:

- `del_vol [-n] [-d] [-f] [-q] vol_id`

  Deletes a volume in an offline LSM.

- `ejecting.sh [-dmox] -c <CAPlist> -v <volumelistfile>`

  Facilitates mass eject vaulting operations.

- `lib_cmd eject cap <cap_id> [-verbose] volume <vol_id...vol_id> | file <path_to_volume_list>`

  Supports eject operations, including the ability to specify long (greater than 42) volume lists.

- `moving.sh -f vol_list_file -t lsm_id...`

  Moves multiple cartridges to one or more LSMs.

Query Status

- `query server`

  Queries ACSLS and library status.-

- `query acs acs_id ... | all`

  Queries ACS status.

- `query lsm lsm_id ... | all`

  Queries LSM status.

- `query cap cap_id ... | all`

  Queries CAP status.

- `query drive drive_id ... | all`

  Queries drive status.

- `query lmu acs_id ... | all`

  Queries LMU and port status for both single-LMU and dual-LMU ACS configurations.

- `query mount vol_id`

  Queries media-compatible drives for a specified data volume.

- `query mount * pool_id ... [ media media_type | media *]`

  Queries media-compatible drives for a specified scratch pool (or volume media type within the pool).

- `query port port_id .... | all`

  Queries port status.

- `query volume vol_id ... | all`

  Queries location of a volume.

- `query clean vol_id ... | all`

  Queries cleaning cartridge status.

- `query scratch pool_id ... | all`

  Queries scratch volumes in a pool.

- `query pool pool_id ... | all`

  Queries scratch pool attributes.

- `query request request_id ... | all`

  Queries request status.

- `query lock  drive | volume identifier ... | all`

  Queries the lock status of a drive or volume.

- `query clean vol_id... | all`

  Queries cleaning cartridge attributes.

Varying Library Components

- `vary acs acs_id ... online | offline | diagnostic [force]`

Changes the desired state and the state of an ACS.

- `vary lsm lsm_id ... online | offline | diagnostic [force]`

  Changes the state of an LSM.

- `vary cap cap_id ... online | offline | diagnostic [force]`

  Changes the state of a CAP.

- `vary drive drive_id ... online | offline | diagnostic [force]`

  Changes the state of a drive.

- `vary port _port_id ... online | offline`

  Changes the desired state and the state of a port.