StorageTek Enterprise Library Software

XAPI Client Interface to ACSLS Server Reference

E65776-01

September 2015

ORACLE®

StorageTek Enterprise Library Software XAPI Client Interface to ACSLS Server Reference

E65776-01

Contents

1 Introduction

2 XCMD Command Conventions

3 XCMD XAPI Configuration and Administration Commands

4 XCMD XAPI Server Operator Commands

5 XCMD Support for XAPI User Requests and HSC/VTCS Commands

6 ACSLS XAPI Server Control Variables

List of Tables

Preface

This publication provides configuration and administration information for an Oracle StorageTek ACSLS server with XAPI support installed.

Audience

This document is intended for storage administrators, system programmers and operators responsible for configuring and maintaining the ACSLS XAPI server.

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.

Related Documents

For more information, refer to the following documents in the Oracle StorageTek ELS Release 7.3 documentation set and ACSLS Release 8.4 documentation sets:

- *ACSLS 8.4 Administrator's Guide*
- *ACSLS 8.4 Installation Guide*
- *ACSLS 8.4 Quick Reference*
- *ELS 7.3 Command, Control Statement,. and Utility Reference*
- *Configuring and Managing SMC* 7.3
- *VM Client Installation, Configuration, and Administration Guide*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

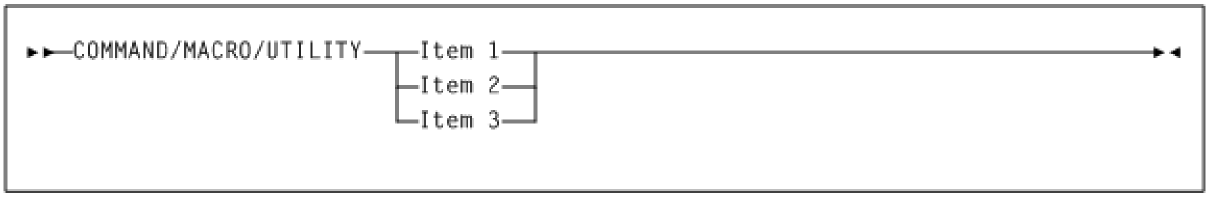
Syntax Conventions

Note: The following syntax conventions ***only*** apply to XAPI and XCMD syntax. Refer to the *ACSLS Administrator's Guide* for information about syntax conventions that apply to ACSLS commands and utilities.

Syntax flow diagramming conventions include the following:

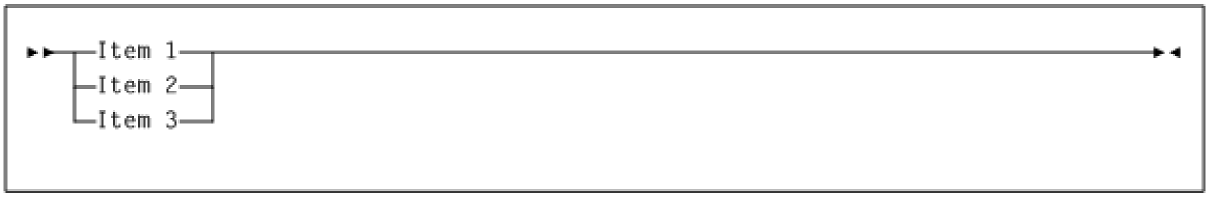
Flow Lines

Syntax diagrams consist of a horizontal base line, horizontal and vertical branch lines, and the text for a command, control statement, macro, or utility. Diagrams are read left to right, and top to bottom. Arrows indicate flow and direction. For example:



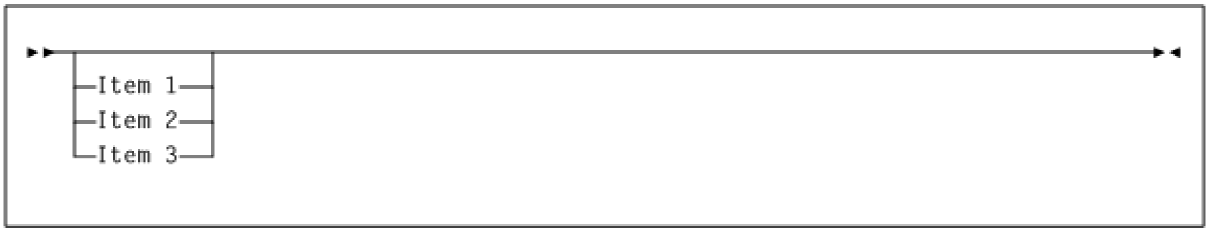
Single Required Choice

Branch lines indicate that a single choice must be made. If one of the items to choose from is positioned on the baseline of the diagram, one item must be selected. For example:



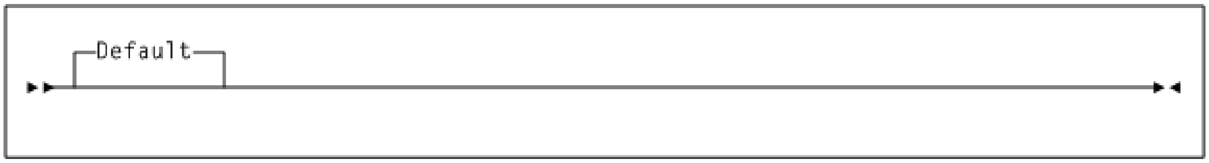
Single Optional Choice

If the first item is positioned on the line below the baseline, one item may be optionally selected. For example:

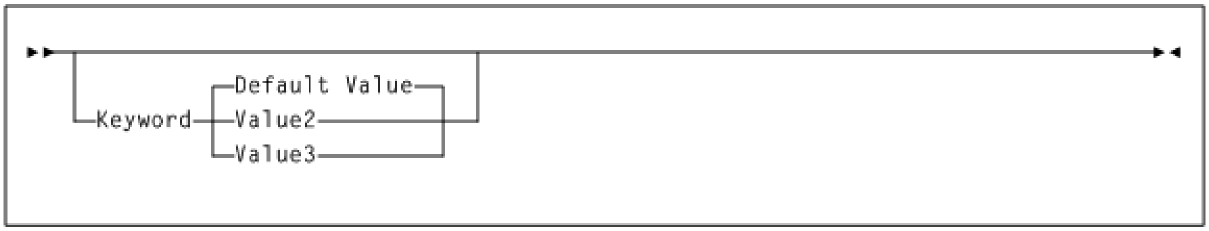


Defaults

Default values and parameters appear above the baseline. For example:

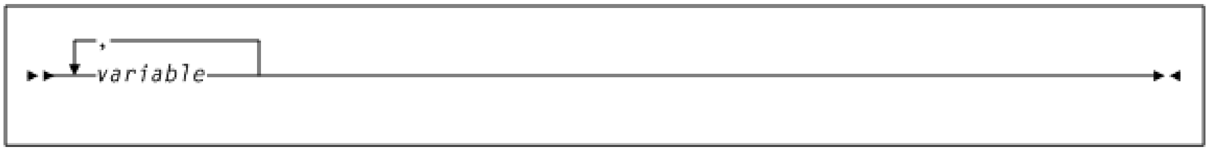


Some keyword parameters provide a choice of values in a stack. When the stack contains a default value, the keyword and the value choices are placed below the base line to indicate that they are optional, and the default value appears above the keyword line. For example:



Repeat

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



Keywords

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

Variables

Italic type indicates a variable.

Alternatives

A bar (|) separates alternative parameter values.

Optional

Brackets [ ] indicate that a command parameter is optional.

Delimiters

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

Ranges

An inclusive range is indicated by a pair of elements of the same length and data type, joined by a dash. The first element must be strictly less than the second element.

Hexadecimal Ranges

A hexadecimal range consists of a pair of hexadecimal numbers (for example, 0A2-0AD, or 000-0FC).

Decimal Ranges

A decimal range consists of a pair of decimal numbers (that is, 1-9, or 010-094). Leading zeros are not required. The decimal portion is an incremental range. The character positions of the incremental portion of both range elements must match, and the non incremental characters of the first element must be identical to those of the second element.

HSC Format Volser Ranges

An HSC format numeric VOLSER range (vol-range) consists of a pair of VOLSER elements containing a decimal numeric portion of 1 to 6 digits (for example, ABC012-ABC025, or X123CB-X277CB). The decimal portion is an incremental range. The following additional restrictions apply:

- The character positions of the incremental portion of both range elements must match.
- The non incremental characters of the first element must be identical to those of the second element.
- You cannot increment two portions of a range element. If 111AAA is the first element, you cannot specify 112AAB for the second element.
- If a VOLSER range contains more than one decimal portion, any portion is valid as the incremental range. For example:
  - A00B00 - The largest range that can be specified is A00B00 through A99B99.
  - A0B0CC - The largest range that can be specified is A0B0CC through A9B9CC.
  - 000XXX - The largest range that can be specified is 000XXX through 999XXX.

An HSC format alphabetic VOLSER range (vol-range) consists of a pair of VOLSER elements containing an incremental portion of 1 to 6 characters (for example, 000AAA-000ZZZ, or 9AAA55-9ZZZ55). This portion is an incremental range. The following additional restrictions apply:

- The character positions of the incremental portion of both range elements must match.
- The non incremental characters of the first element must be identical to those of the second element.
- You cannot increment two portions of a range element. If 111AAA is the first element, you cannot specify 112AAB for the second element.

- The alphabetic portion of the VOLSER range is defined as being from character A to Z. To increment multi-character sequences, each character increments to Z. For instance, ACZ is part of the AAA-AMM range. Examples are:
  - A00A0-A99A0

    increments VOLSERs A00A0 through A09A0, then A10A0 through A99A0.
  - 9AA9A-9ZZ9A

    increments VOLSERs 9AA9A through 9AZ9A, then 9BA9A through 9ZZ9A.
  - 111AAA-111ZZZ

    increments VOLSERs 111AAA through 111AAZ, then 111ABA through 111ZZZ
  - 999AM8-999CM8

    increments VOLSERs 999AM8 through 999AZ8, then 999BA8 through 999CM8
  - A3BZZ9-A3CDE9

    increments VOLSERs A3BZZ9 through A3CAA9, then A3CAB9 through A3CDE9
  - AAAAAA-AAACCC

    increments VOLSERs AAAAAA through AAAAAZ, then AAAABA through AAACCC
  - CCCNNN-DDDNNN

    increments VOLSERs CCCNNN through CCCNNZ, then CCCNOA through DDDNNN. This is a very large range.

  The number of volumes in an alphabetic VOLSER range depends on the number of elements in the incrementing portion of the VOLSER range. For an A to Z range in each character position, the number of volumes can be calculated by 26 to the power of the number of positions that are being incremented.
  - A-Z is equivalent to 26^1 or 26 volumes.
  - AA-ZZ is equivalent to 26^2 or 676 volumes.
  - AAA-ZZZ is equivalent to 26^3 or 17,576 volumes.
  - AAAA-ZZZZ is equivalent to 26^4 or 456,976 volumes.
  - AAAAA-ZZZZZ is equivalent to 26^5 or 11,881,376 volumes.
  - AAAAAA-ZZZZZZ is equivalent to 26^6 or 308,915,776 volumes.

ACSLS Format Alphanumeric Volser Ranges

An ACSLS format alphanumeric volser range is not required to conform to the HSC format volume pattern ranges specified above. The following additional restrictions apply:

- Both volsers specified in the range must contain the same number of characters.
- The first volser element in the range must be strictly less than the second volser element.
- If the specified volsers are less than 6 characters in length, the volsers are evaluated as if they were blank filled and left justified 6 character fields.

- Only volsers that are strictly greater than or equal to the first volser element, and strictly less than or equal to the second volser element, when evaluated using the ASCII collating sequence, are in the range. For example, volser range 123456-234567 contains the volser 12345A.

Note:

- A valid HSC format volser range is always valid under the ACSLS format volser rules. However, a valid ACSLS format volser range may not be valid when evaluated using the HSC format volser rules.
- Most XCMD commands that accept volser ranges require that ranges conform to the HSC volser rules. The only command that accepts volser ranges in either format is the XCMD SET POOL_Name command.

Lists

A list consists of one or more elements. If more than one element is specified, the elements must be separated by a comma or a blank space, and the entire list must be enclosed in parentheses.

Blanks

Keyword parameters and values may be separated by any number of blanks.

1

Introduction

This chapter introduces the XAPI client interface to ACSLS and describes the XCMD interface that enables you to enter commands targeting an ACSLS server with XAPI.

What is XAPI Client Interface to ACSLS Server?

The XML API (XAPI) is Oracle's StorageTek API that enables StorageTek clients and servers to communicate using a common protocol over TCP/IP.

With the introduction of this XAPI, clients who were previously required to use an MVS based server (Oracle's StorageTek Host software Component) for real tape processing can now use ACSLS (with XAPI support enabled) as follows:

- An SMC client on MVS can now request real tape requests from an ACSLS server with XAPI support enabled (without requiring MVS/CSC).

  For more information, refer to the ELS publication *Configuring and Managing SMC*.

- A VM Client can now request real tape services from an ACSLS server with XAPI support enabled.

  For more information, refer to the ELS publication *VM Client Installation, Configuration, and Administration Guide*.

The XAPI service is supported for ACSLS 8.4 and later releases. Refer to the *StorageTek ACSLS 8.4 Installation Guide* for more information about installing and configuring the ACSLS XAPI service.

What is the XCMD Interface Component?

The ACSLS cmd_proc component is a command-line interface for ACSLS that enables customers to manage an ACSLS system and the libraries that ACSLS controls.

Refer to the Overview chapter in the *StorageTek Automated Cartridge System Library Software Administrator's Guide* for more detailed information on using cmd_proc.

The `XCMD` command is an extension of the ACSLS cmd_proc that enables users of the ACSLS XAPI server component to enter a limited number of XAPI server user and administrator requests from the existing ACSLS cmd_proc component when the ACSLS XAPI server component is installed and active.

See Chapter 3, "XCMD XAPI Configuration and Administration Commands" for information about XAPI server commands you may enter from the ACSLS cmd_proc using the `XCMD` command.

See Chapter 4, "XCMD XAPI Server Operator Commands" for information about XAPI operator commands you may enter from the ACSLS cmd_proc using the XCMD command.

Additionally, you can use the SMC/MVS `Route` command or the VM/Client `Route` command to issue XAPI server operator commands, XCMD configuration and administration commands, and HSC/VTCS XAPI commands from an MVS or VM host. However, certain parameter restrictions apply.

See Chapter 5, "XCMD Support for XAPI User Requests and HSC/VTCS Commands" for information about supported HSC/VTCS commands and other XAPI requests, and parameter restrictions.

2

XCMD Command Conventions

The majority of client/server interaction between SMC/MVS and VM/Clients, and an ACSLS server with XAPI are transparent to the end user. Requests for volume information, mounts, and dismounts are generated automatically by the SMC/MVS and VM/Clients and are processed without operator intervention.

In addition to these automatic interactions, the ACSLS server with XAPI provides additional administrator, configuration, and operator commands that enable you to manage the XAPI component.

- See Chapter 3, "XCMD XAPI Configuration and Administration Commands"for information about XCMD XAPI server configuration and administration commands.
- See Chapter 4, "XCMD XAPI Server Operator Commands"for information about XCMD XAPI server operator commands

These commands are not part of the ACSLS base installation, but are available when the ACSLS XAPI component is optionally installed. Refer to the *StorageTek Automated Cartridge System Library (ACSLS) Installation Guide* for information about installing ACSLS and the ACSLS XAPI component.

To issue these commands, use the XCMD command facility, issued from the ACSLS cmd_proc. The general syntax of this command is:

```
XCMD command
```

Where `command` is one of the ACSLS XAPI component commands described in this publication.

> **Note:** You can also use the XCMD command facility to issue any XAPI user request supported by the ACSLS XAPI server. See Chapter 5, "XCMD Support for XAPI User Requests and HSC/VTCS Commands" for more information.

The following conventions apply whenever you issue an XCMD command from the ACSLS cmd-proc:

- XCMD does not use ACSLS command prompting.
- When you enter keyword parameters with an associated value, the value must be preceded by a single blank character.
- You can enter XCMD commands in uppercase or lowercase. The ACSLS XAPI component automatically translates all command input to upper case. For

example, the subpool names "Testpool1", "testpool1", "TestPool1", and "TESTPOOL1" are all translated to the same subpool name "TESTPOOL1".

Note: Normal ACSLS cmd_proc commands and parameters are usually specified in lowercase. Lowercase characters are not automatically translated to uppercase, except for vol_ids (volsers).

- You cannot precede an XCMD parameter value with an equal sign (=).
- You cannot enclose XCMD values in parentheses. When multiple values are supported, you can use a comma to separate each value. For example, to query multiple volumes, enter XCMD QUERY VOLUME volser1,volser2 and not XCMD QUERY VOLUME(volser1,volser2).

3

XCMD XAPI Configuration and Administration Commands

This chapter describes XAPI configuration and administration commands supported by XCMD. These commands enable ELS-compatible volume pool features in ACSLS Release 8.4. Supported commands include:

- DEFINE POOL_Name
- DEFINE POOL_Access
- DELETE POOL_Name
- DELETE POOL_Access
- SET POOL_Name
- QUERY POOL_Name
- QUERY POOL_Access

Issue these commands from the ACSLS user_proc using the XCMD command. See Chapter 2, "XCMD Command Conventions" for more information.

See "Syntax Conventions" for conventions used in the syntax flow diagrams included in the following command sections.

DEFINE POOL_Name

The following section describes the `DEFINE POOL_Name` command.

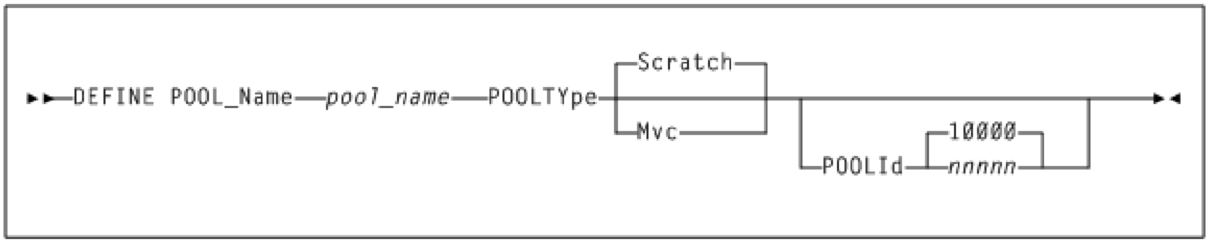
Description

The `DEFINE POOL_NAME` command associates a pool type and pool name with an ACSLS pool ID.

Syntax

The following figure displays syntax for the `DEFINE POOL_Name` command:

Figure 3–1 DEFINE POOL_Name command syntax



Parameters

As shown in Figure 3–1, the `DEFINE POOL_Name` command includes the following parameters:

pool_name
A pool name of up to 13 characters. The pool name can include any characters except for the percent sign ("%"), underscore ("_"), or asterisk (“*”).

POOLTYpe
The type of pool, either `Scratch` (the default) or `Mvc`.

POOLId (nnnnn)
Optionally, specifies the pool ID (*`nnnnn`*).

- If you choose to specify this parameter, you must specify a pool ID that does not currently exist.
- If you do not specify this parameter, the system automatically assigns a pool ID as the first unused value starting at 10000.

Example

In the following example, a user issues the `DEFINE POOL_Name` command to define scratch pool `SCRPOOL1`:

```
XCMD DEFINE POOL_NAME SCRPOOL1 POOLTYPE SCRATCH
```

DEFINE POOL_Access

The following section describes the `DEFINE POOL_Access` command.

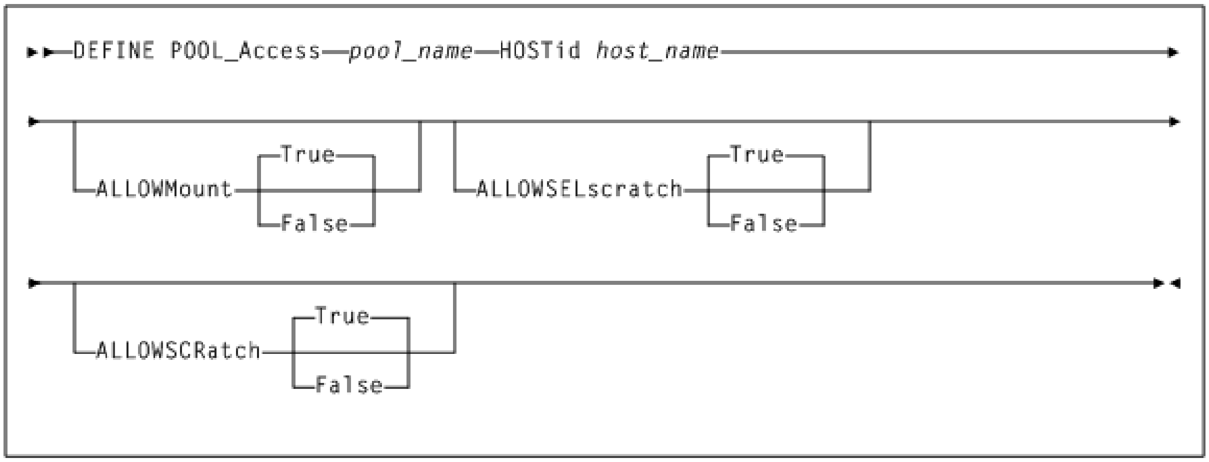
Description

The `DEFINE POOL_Access` command defines the access allowed to a *`pool_name`* by a specified host. If no pool access is defined for a specific *`pool_name`*, then all access is allowed.

Syntax

The following figure displays syntax for the `DEFINE POOL_Access` command:

Figure 3–2 DEFINE POOL_Access command syntax



Parameters

As shown in Figure 3–2, the `DEFINE POOL_Access` command includes the following parameters:

pool_name
A pool name of up to 13 characters. The pool name can include any characters except for the percent sign ("%"), underscore ("_"), or asterisk ("*") (asterisk). The specified *`pool_name`* must have been previously defined using the `DEFINE POOL_Name` command with a pool-type of `SCRATCH` (access is not defined for MVC pools). See "DEFINE POOL_Name" for more information.

HOSTid host_name
Specifies a host-name of up to 31 characters. This parameter is required.

The specified `HOSTid` is the host name portion of the internet fully qualified domain name. For example, if your fully qualified domain name is `myhost.plus.domain.name`, the specified `HOSTid` is `myhost`.

ALLOWMount
optionally, specifies mount access. You can specify `True` or `False`. This parameter is only valid for a pool defined as `POOLTYPE MVC`.

ALLOWSELscratch
optionally, specifies access for the XAPI commands `query_scr_mnt_info`, `sel_scratch`, and `mount` for a scratch volume. You can specify `True` or `False`. If you do not specify this parameter, it is set to `True` by default.

This parameter is only valid for a pool defined as `POOLTYPE SCRATCH`. If you specify this parameter for a pool defined as `MVC`, the parameter automatically defaults to `False`.

ALLOWSCRatch
optionally, specifies scratch access. You can specify `True` or `False`. If you do not specify this parameter, it is set to `True` by default.

This parameter is only valid for a pool defined as `POOLTYPE SCRATCH`. If you specify this parameter for a pool defined as `MVC`, the parameter automatically defaults to `False`.

Example

In the following example, a user issues the `DEFINE POOL_Access` command to define set access for scratch pool name `SCRPOOL1`, where the host is `PRODMVS` and `ALLOWMOUNT` and `ALLOWSCRATCH` are `True`:

```
XCMD DEFINE POOL_ACCESS SCRPOOL1 HOSTID PRODMVS ALLOWMOUNT TRUE
ALLOWSCRATCH TRUE
```

DELETE POOL_Name

The following section describes the `DELETE POOL_Name` command.

Description

The `DELETE POOL_Name` command deletes a pool. This command is similar to the ACSLS `delete pool` command, with the following exceptions:

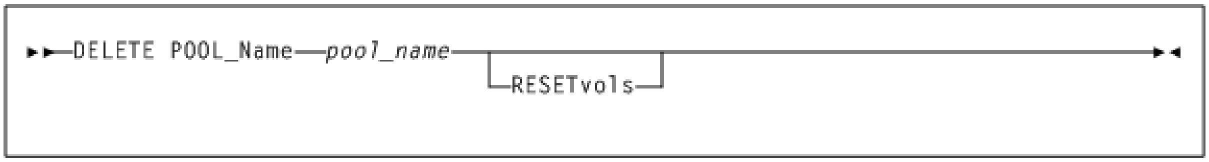
- The input specifies a *`pool name`* instead of a *`pool ID`*.
- The user has the option to reassign volumes assigned to the pool to pool 0.

Refer to the *StorageTek Automated Cartridge System Library Software (ACSLS) Administrator's Guide* for more information about the ACSLS `delete pool` command.

Syntax

The following figure displays syntax for the `DELETE POOL_Name` command:

Figure 3–3 DELETE POOL_Name command syntax



Parameters

As shown in Figure 3–3, the `DELETE POOL_Name` command includes the following parameters:

pool_name
A pool name of up to 13 characters. The pool name can include any characters except for the percent sign ("%"), underscore ("_"), or asterisk ("*"). The specified *`pool_name`* must have been previously defined using the `DEFINE POOL_Name` command. See "DEFINE POOL_Name" for more information

RESETvols
Indicates that volumes currently assigned to the pool being deleted should be reassigned to pool 0. If you do not specify this parameter, and volumes currently exist within the pool, then the `delete pool-name` command fails.

Example

In the following example, a user issues the `DELETE POOL_Name` command to delete pool `SCRPOOL1` and reassign its volumes to pool `0` (the default pool):

```
XCMD DELETE POOL_NAME SCRPOOL_TEST1 RESETVOLS
```

DELETE POOL_Access

The following section describes the `DELETE POOL_Access` command.

Description

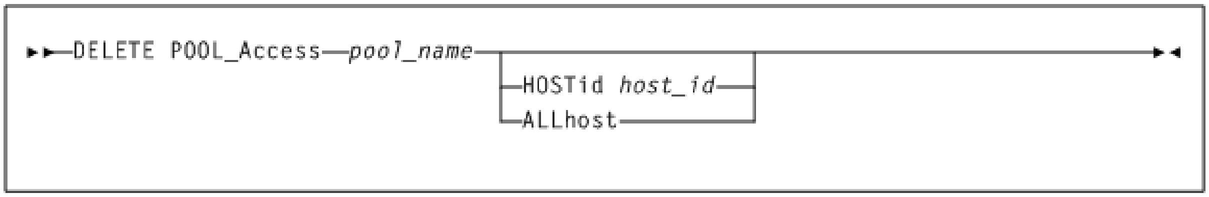
The `DELETE POOL_Access` command deletes pool access rules associated with a single host name (*`host_id`*), or deletes all pool access rules for all hosts.

- If a host name (*`host_id`*) is specified, then only the access rules for the specified host are deleted from the specified pool.
- If `ALLhost` is specified, all pool access rules for the pool are deleted.

Syntax

The following figure displays syntax for the `DELETE POOL_Access` command:

Figure 3–4   DELETE POOL_Access command syntax



Parameters

As shown in "DELETE POOL_Access command syntax", the `DELETE POOL_Access` command includes the following parameters:

pool_name
A pool name of up to 13 characters. The pool name can include any characters except for the percent sign ("%"), underscore ("_"), or asterisk (“*”). The specified *`pool_name`* must have been previously defined using the `DEFINE POOL_Name` command. See "DEFINE POOL_Name" for more information.

HOSTid host_name
optionally, specifies a host-name of up to 31 characters, and indicates that access restrictions for this host ID should be removed (with any other host ID access rules remaining unchanged). This parameter is mutually exclusive with `ALLhost`

ALLhost
optionally, specifies that all access restrictions for the specific *`pool_name`* for all host IDs are removed. This parameter is mutually exclusive with `HOSTid`.

Example

In the following example, a user issues the `DELETE POOL_Access` command to delete pool access from host `PRODMVS` to `SCRPOOL1`:

```
XCMD DELETE POOL_ACCESS SCRPOOL1 HOSTID PRODMVS
```

SET POOL_Name

The following section describes the SET POOL_Name command.

Description

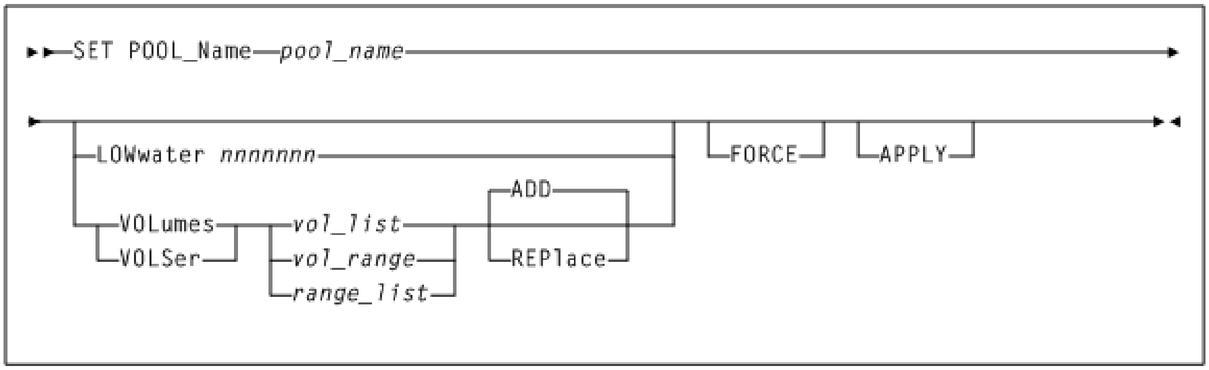
The SET POOL_Name command assigns volumes to a pool name, or sets the low water mark for a pool name.

> **Note:** If the command syntax is valid, specified volume ranges are added to the database even if some volumes within the range cannot be added to the pool. Volumes are not eligible to be added to a pool if one of the following is true:
>
> - The volume is LOCKed.
> - The volume is already in a different pool, unless FORCE is specified.
> - The volume is in a logical library.
> - The volume has an ACSLS owner, and access control is in effect.

Syntax

The following figure displays syntax for the SET POOL_Name command:

Figure 3–5 SET POOL_Name command syntax



Parameters

As shown in Figure 3–5 the SET POOL_Name command includes the following parameters:

pool_name
A pool name of up to 13 characters. The pool name can include any characters except for the percent sign ("%"), underscore ("_"), or asterisk (“*”). The specified *pool_name* must have been previously defined using the DEFINE POOL_Name command. See "DELETE POOL_Name" for more information.

LOWwater nnnnnnn
Optionally, defines the minimum number of available volumes that the pool must contain. If the number of available volumes drops below the low water mark, the ACSLS system generates a warning message.

VOLumes or VOLSer (volser-list, volser-range, range-list)

Optionally, specifies a list of volsers, range of volsers, or list of volser ranges. You can specify a maximum of 16 single volumes or ranges.

Volume ranges are evaluated according to ACSLS format volser ranges: ACSII collating sequence, and do not need to conform to HSC volume pattern ranges; however, both volume serial numbers in a single range must contain the same number of characters, and only volume serial numbers with a matching number of characters are considered within the range. For example, the range 1234-12356 is invalid. In the range 1234-2345, a volume serial of 123567 is not considered within the range. If the specified ranges are valid, they are added to the database even if some volumes within the range cannot be assigned to the pool. The specified volume ranges must not overlap any existing volume ranges, and must not overlap each other.

Volume ranges are evaluated according to ACSLS format volser ranges:

- Both volsers specified in the range must contain the same number of characters.
- The first volser element in the range must be strictly less than the second volser element.
- If the specified volsers are less than 6 characters in length, the volsers are evaluated as if they were blank filled and left justified 6 character fields.
- Only volsers that are strictly greater than or equal to the first volser element, and strictly less than or equal to the second volser element, when evaluated using the ASCII collating sequence, are in the range. For example, volser range 123456-234567 contains the volser 12345A.

If the specified ranges are valid, they are added to the *`pool_name`* database even if some volumes within the range cannot be assigned to the pool. The specified volume ranges must not overlap any existing volume ranges and must not overlap each other.

ADD

Indicates that the specified volume ranges are to be added to existing ranges for the pool. This is the default. If specified, VOLumes must also be specified. `ADD` is mutually exclusive with `REPLace`.

REPlace

Indicates that the specified volume ranges are to replace any volume ranges currently specified, and that volumes in the previous ranges are to be reassigned to pool 0. If specified, `VOLumes` (or `VOLSer`) must also be specified. `REPlace` is mutually exclusive with `ADD`.

FORCE

Optionally, indicates that volumes already assigned to other pools should be reassigned to the new pool. If this parameter is not specified, any volume currently assigned to a different pool is not updated. If specified, `VOLumes` (or `VOLSer`) must also be specified.

APPLY

Optionally, indicates that previously defined volume ranges should be applied to volumes in the database. This parameter can be used when some volumes were not updated due to restrictions listed above. If specified, `VOLumes` (or `VOLSer`) must also be specified.

Example

In the following example, the command adds volumes `DRL001-DRL999` to `SCRPOOL1`:

```
XCMD SET POOL_NAME SCRPOOL_TEST1 VOLUMES DRL001-DRL999
```

QUERY POOL_Name

The following section describes the `QUERY POOL_Name` command.

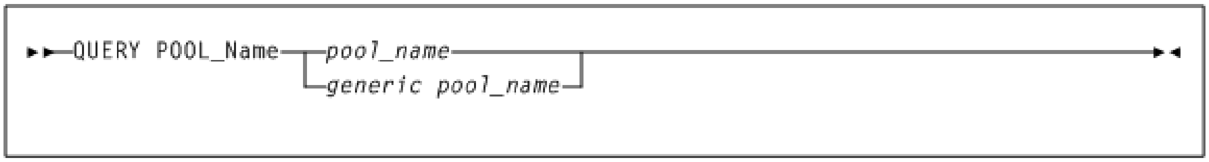
Description

The `QUERY POOL_Name` command displays all pools, pools starting with specified characters, or a single pool.

Syntax

The following figure displays syntax for the `QUERY POOL_Name` command:

Figure 3–6 QUERY POOL_Name command syntax



Parameters

As shown in Figure 3–6, the `QUERY POOL_Name` command includes the following parameters:

pool_name
Either an existing pool name of up to 13 characters, or a generic name, with the final asterisk ('*') character. This parameter is required.

The *`pool_name`* format determines the output display, as follows:

- When *`pool_name`* is non-generic, the displayed pool information includes the defined volsers and volser ranges for the single matched pool.
- When *`pool_name`* is generic (includes a trailing '*'), the displayed pool information does not include the defined volsers and volser ranges for the matched pools.

Example

In the following example, a user issues the `QUERY POOL_Name` command to display information about pool1, including associated volume ranges.

```
XCMD QUERY POOL_NAME POOL1
```

In the next example, a user issues the `QUERY POOL_Name` to display information about all named pools (not including volume ranges):

```
XCMD QUERY POOL_NAME *
```

In the next example, a user issues the `QUERY POOL_Name` command to display information about all pools starting with "`A`" (not including volume ranges):

```
XCMD QUERY POOL_NAME A*
```

QUERY POOL_Access

The following section describes the `QUERY POOL_Access` command.

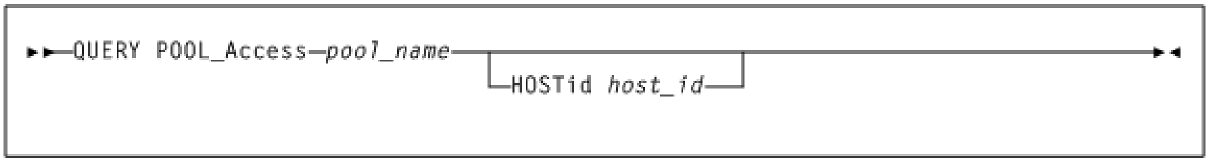
Description

The `QUERY POOL_Access` command displays pool access rules for a single host or all hosts associated with a pool name.

Syntax

The following figure displays syntax for the `QUERY POOL_Access` command:

Figure 3–7 QUERY POOL_Access command syntax



Parameters

As shown in Figure 3–7, the `QUERY POOL_Access` command includes the following parameters:

pool_name
A pool name of up to 13 characters. The pool name can include any characters except for the percent sign ("%"), underscore ("_"), or asterisk (“*”). The specified *`pool_name`* must have been previously defined using the `DEFINE POOL_Name` command. See "DEFINE POOL_Name" for more information.

HOSTid host_id
optionally, specifies a host_name of 1-31 characters. The specified *`host_id`* must have been previously defined for the *`pool_name`* using the `DEFINE_POOL_Access` command. See "DEFINE POOL_Access" for more information.

If you do not specify this parameter, access rules for all hosts associated with the pool are displayed.

Example

In the following example, a user issues the `QUERY POOL_Access` command to query access rules for all hosts associated with pool_name `SCRPOOL1`:

```
XCMD QUERY POOL_ACCESS SCRPOOL1
```

4

XCMD XAPI Server Operator Commands

This chapter describes XAPI server operator commands supported by XCMD. These commands enable an operator or administrator to monitor the ACSLS XAPI component. Supported commands include:

- LOG
- LIST
- MSGLVL
- TRACE

Issue these commands from the ACSLS user_proc using the `XCMD` command. See "XCMD Command Conventions" for more information.

See "Syntax Conventions" for conventions used in the syntax flow diagrams included in the following command sections.

LOG

The following section describes the `LOG` command.

Description

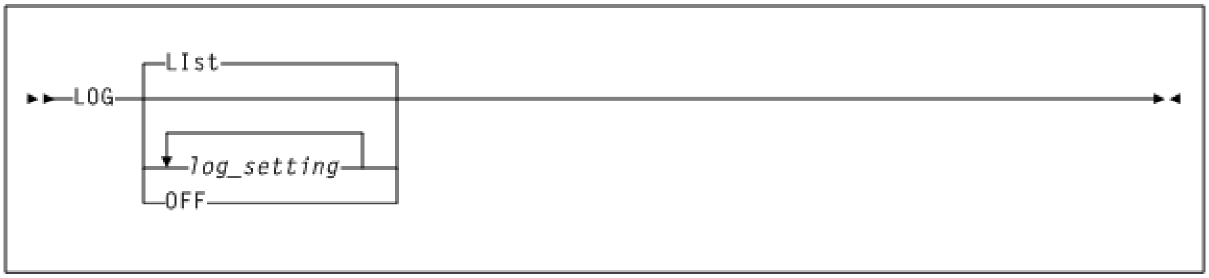
The `LOG` command changes or lists the current XAPI server log settings.

> **Note:** The log file name and location are set by ACSLS environmental variables and default to file name `vlog.file` in the `XAPI_WORK_PATH` directory.

Syntax

The following figure displays syntax for the `LOG` command:

Figure 4–1 LOG command syntax



Parameters

As shown in Figure 4–1, the `LOG` command includes the following parameters:

LIst
optionally, specifies that the current log setting is displayed. The default if no command parameters are specified is `LIst`.

log_setting
optionally, indicates the desired log setting specified as a series of '0' and '1' characters.

- `0` signifies that the desired positional log setting should be `OFF` or disabled.
- `1` signifies that the desired positional log setting should be `ON` or enabled.

The value specified completely replaces, and is not merged with the current log setting. The positional log settings are as follows:

- `1` - Log error messages to the ACSLS XAPI component log.
- `01` - Log messages to log file.
- `001` - Log input transaction errors to the log file.
- `0001` - Log all XML recv() packets to the log file.
- `00001` - Log all XML send() packets to the log file.
- `000001` - Log local commands and responses to the log file.

XAPI `LOG` messages will be saved in the `$ACS_HOME/log/xapi/vlog.file`.

> **Note:** The `LOG` command enables you to set and display more positions of 0's and 1's (16) than are currently defined as `LOG` settings above; this allows for future expansion and any extraneous 0 or 1 `LOG` setting are simply ignored.

OFF
optionally, specifies that logging should be disabled. This is equivalent to `LOG 0`.

Example

In the following example, a user issues the `LOG` command to set the XAPI server to log all errors and all XML recv() packets to the log file:

```
XCMD LOG 1011
```

LIST

The following section describes the `LIST` command.

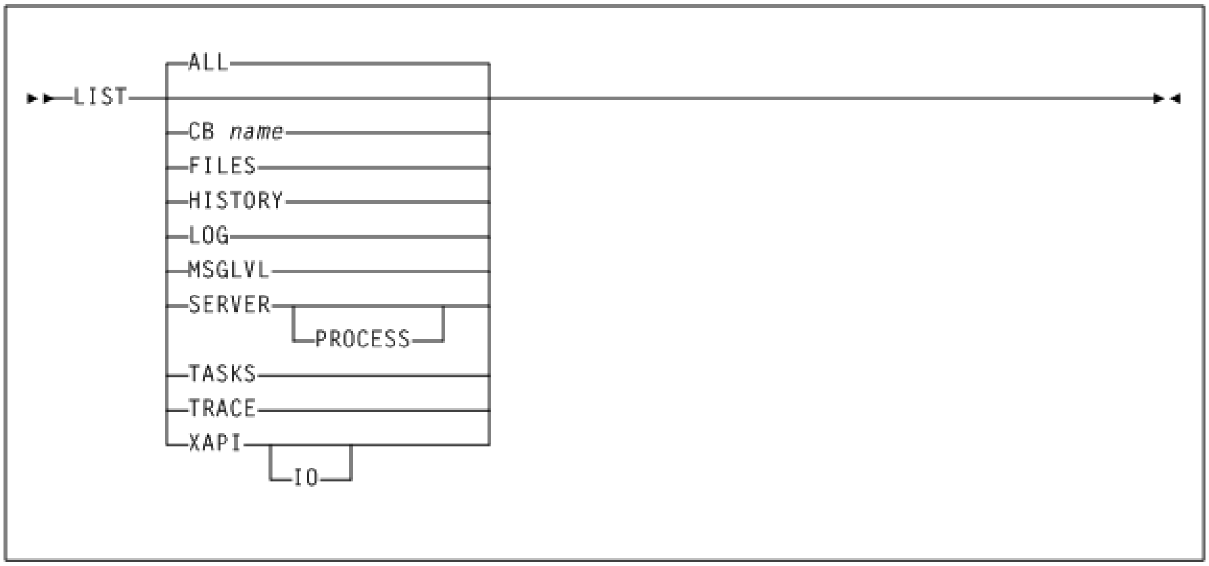
Description

The `LIST` command lists current XAPI server status and settings.

Syntax

The following figure displays syntax for the `LIST` command:

Figure 4–2 LIST command syntax



Parameters

As shown in Figure 4–2, the `LIST` command includes the following parameters:

ALL
optionally, displays all XAPI server status and settings. This is the default if you issue the `LIST` command with no parameters.

CB name
Specifies that the named control block is displayed in character, and character hexadecimal format. Use only at the direction of Oracle support.

Specify one of the following control blocks for *`name`*:

- `HTTPCVT`
- `HTTPGBL`
- `HTTPREQ-`*`nnn`*
- `HTTPAPI-`*`nnn`*

For `LIST CB HTTPREQ` and `LIST CB HTTPAPI`, an index between 0 and 999 must be specified. Additionally, the index must be delimited from the control block name by a single dash ("-") with no intervening spaces. For example:

```
LIST HTTPREQ-0
```

or

```
LIST HTTPAPI-900
```

FILES
optionally, specifies that the full path name of XAPI server control, log, and trace files is displayed.

HISTORY
optionally, specifies that the XAPI transaction count history is displayed for the past 24 hours.

LOG
optionally, specifies that the current XAPI server log setting is displayed. This is equivalent to issuing a `LOG` command with no parameters.

MSGLVL
optionally, specifies that the current XAPI server message level setting is displayed. This is equivalent to issuing `MSGLVL` command with no parameters.

SERVER
optionally, specifies that the current XAPI server and UNIX versions, relevant parameters and environment variables, and shared segment and message queues are displayed.

You can optionally include the `PROCESS` keyword to request that all active XAPI server process ids, thread and open file counts, CPU and memory usage are also displayed.

PROCESS
When specified with `SERVER`, the `PROCESS` keyword requests that all active XAPI server process ids, thread and open file counts, CPU and memory usage are displayed as well.

TASKS
optionally, specifies that the current XAPI server system and work tasks are displayed

TRACE
optionally, specifies that the current XAPI server trace setting is displayed. This is equivalent to issuing a `TRACE` command with no parameters

XAPI
optionally, specifies that the current XAPI server listener port, and IP address are displayed.

You can optionally include the `IO` keyword to request that all XAPI server listener statistics are also displayed.

Example

In the following example, a user issues the `LIst` command to display the current XAPI server listener port, IP address, and listener status:

```
XCMD LIST XAPI IO
```

MSGLVL

The following section describes the `MSGLVL` command.

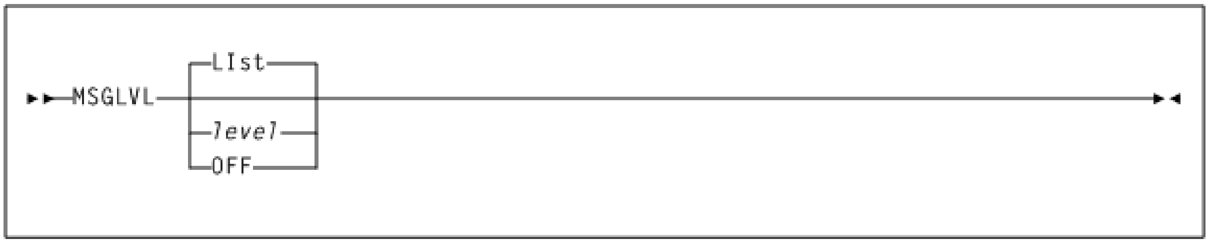
Description

The MSGLVL command changes or lists the message verbosity of the XAPI server.

Syntax

The following figure displays syntax for the MSGLVL command:

Figure 4–3 MSGLVL command syntax



Parameters

As shown in Figure 4–3, the MSGLVL command includes the following parameters:

LIst
Specifies that the current MSGLVL setting is displayed. The default if no command parameters are specified is LIst.

level
specifies the desired MSGLVL setting. The higher the level, the more verbose the XAPI server messaging. Message levels are cumulative; MSGLVL 8 produces all messages up to and including MSGLVL 8 (that is, MSGLVL 0 through 8 are produced). Message levels are generalized as follows:

- 0 - Normal startup, shutdown, and error messages; these messages cannot be suppressed.
- 4 - Serious warning messages.
- 8 - Minor warning messages.
- 12 - Startup parameter and option messages.
- 16 - Additional startup and shutdown messages.
- 20 - Additional diagnostic level 20 messages.
- 24 - Additional diagnostic level 24 messages.
- 28 - Additional task startup and shutdown messages.

OFF
Specifies that all verbose messages should be disabled. This is equivalent to MSGLVL 0.

Example

In the following example, a user issues the MSGLVL command to change the XAPI server messaging to include all warning messages:

```
XCMD MSGLVL 8
```

TRACE

The following section describes the TRACE command.

Description

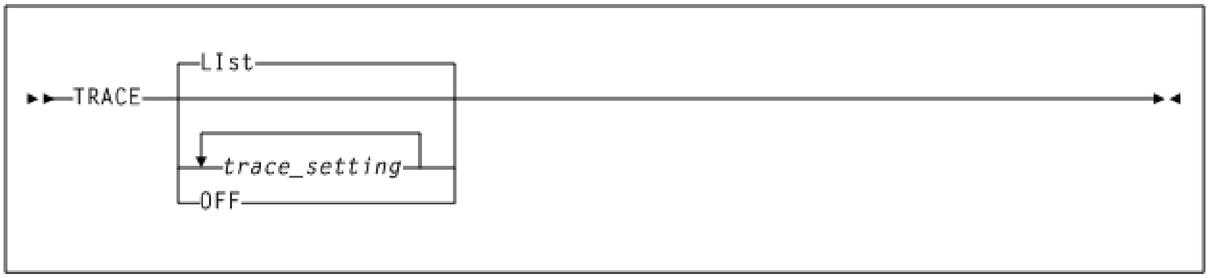
The TRACE command changes or lists the XAPI server trace settings.

> **Note:**
>
> - The trace file name and location are set by ACSLS environmental variables and default to file name vtrace.file in the DV_TAG_XAPI_WORK_PATH variable.
> - Tracing can have significant impact on system performance. Only set tracing ON at the request of Oracle StorageTek support.
> - The TRACE command enables you to set and display more positions of 0's and 1's (16) than are currently defined as TRACE settings above; this allows for future expansion and any extraneous 0 or 1 TRACE setting is simply ignored.

Syntax

The following figure displays syntax for the TRACE command:

Figure 4–4 TRACE command syntax



Parameters

As shown in Figure 4–4, the TRACE command includes the following parameters:

LIst
optionally, specifies that the current trace setting is displayed. The default if no command parameters are specified is LIst.

trace_setting
optionally, indicates the desired trace setting specified as a series of '0' and '1' characters. 0 signifies that the desired positional trace setting should be OFF or disabled, while 1 signifies that the desired positional trace setting should be ON or enabled. The value specified completely replaces, and is not merged with the current trace setting. The positional trace settings are as follows:

- 1 - Trace errors to the trace file.
- 01 - Trace TCP/IP functions and events to the trace file.
- 001 - Trace PGMI or ACSAPI functions and events to the trace file.

- 0001 - Trace otherwise unclassified XAPI server events to the trace file.
- 00001 - Trace malloc() and free() events to the trace file.
- 000001 - Trace XML parser events to the trace file.
- 0000001 - Trace command server process events to the trace file.
- 00000001 - Trace monitor process events to the trace file.
- 000000001 - Trace CSV functions and events to the trace file.

OFF
optionally, Specifies that tracing is disabled. This is equivalent to `TRACE 0`.

Example

In the following example, a user issues the `TRACE` command to set the XAPI server to trace all errors, and all malloc() and free() events to the trace file:

```
XCMD TRACE 10001
```

5

XCMD Support for XAPI User Requests and HSC/VTCS Commands

In addition to the XAPI server configuration and administration commands described in Chapter 3, and the XAPI server operator commands described in Chapter 4, the XCMD interface also supports many XAPI user requests and compatible HSC/MVS text commands.

For example, to enter a QUERY_VOLUME request from the ACSLS user_proc using the XCMD command:

```
XCMD QUERY_VOLSER S00001
```

Supported XAPI Requests, Commands, and Parameters

The following table lists XAPI requests and corresponding HSC/MVS text commands that are supported by XCMD:

Table 5–1 XCMD Supported XAPI Requests and Corresponding HSC/MVS Commands

XAPI Request	Corresponding HSC/MVS Text Command
DISMOUNT	DISMount
EJECT	EJect
MOUNT	Mount
MOVE	MOVe
QUERY_CAP	Display Cap
QUERY_DRIVES	Display DRives
QUERY_DRIVE_INFO	Display DRIVE_INFO
QUERY_LSM	Display Lsm
QUERY_SERVER	No corresponding HSC/MVS text command
QUERY_SCRATCH	Display SCRatch
QUERY_THRESHOLD	Display THReshold
QUERY_VOLSER or QUERY_VOLUME	Display Volser
QUERY_VOLUME_INFO	Display VOLume_info
SCRATCH	SCRAtch
UNSCRATCH	UNSCratch

Table 5–1 (Cont.) XCMD Supported XAPI Requests and Corresponding HSC/MVS

XAPI Request	Corresponding HSC/MVS Text Command
VOLRPT[1]	VOLRpt

[1] This VOLRPT is the HSC format VOLRPT command and report, NOT the ACSLS format volrpt.

Refer to the *ELS Command, Control Statement, and Utility Reference* for information about the HSC/MVS text commands listed above.

Though these requests and commands are supported, not all parameters are valid in ACSLS. Invalid parameters are as follows:

- DISMOUNT (or DISMount command)

  Specification of the following results in a syntax error:

  – MVS device address format ccuu

  – hostid positional parameter

- EJECT (or EJect command)

  Specification of the following parameters results in a syntax error:

  – WAITCAP

  – SEQ

  – RECTECH

- MOUNT (or Mount command)

  Specification of the following results in a syntax error:

  – MVS device address format ccuu

  – hostid positional parameter

  – MGMTCLAS parameter is invalid

- MOVE (or MOVe command)

  All parameters are supported.

- QUERY_CAP (or Display Cap command)

  – All parameters are supported.

- QUERY_DRIVES (or Display DRives command)

  Specification of the following parameters results in a syntax error:

  – ALL

  – IDLE

  – LIBRARY

  – DETAIL

  – IDENTITY

  – MEDIA

  – RECTECH

  – UNIT

  – BYDRIVE

  - BYLOC
  - SHOWLSLOT

- QUERY_DRIVE_INFO (or Display DRIVE_INFO command)

  Specification of the following parameters results in a syntax error:

  - ALL
  - IDLE
  - LIBRARY
  - DETAIL
  - IDENTITY
  - MEDIA
  - RECTECH
  - VIRTUAL
  - DEVADDR
  - LSMLOC

- QUERY_LSM (or Display Lsm command)

  All parameters are supported.

- QUERY_SERVER (or Display SERVER command)

  All parameters are supported.

- QUERY_SCRATCH (or Display SCRatch command)

  All parameters are supported.

- QUERY_THRESHOLD (or Display THReshold command)

  All parameters are supported.

- QUERY_VOLSER (or Display Volser command)

  All parameters are supported.

- QUERY_VOLUME_INFO (or Display VOLume_info command)

  All parameters are supported.

- SCRATCH (or SCRAtch command)

  All parameters are supported.

- UNSCRATCH (or UNSCRAtch command)

  All parameters are supported.

- VOLRPT (or VOLRpt command)

  Specification of the following results in a syntax error:

  - MEDEQUAL
  - NONMEDEQ
  - VAULT

Issuing Commands to the ACSLS XAPI Server using the SMC or VM Client Route Command

With the exception of the EJECT or VOLRpt commands, you may issue any of the HSC/MVS commands listed in Table 5–1 from an SMC/MVS client or VM/client to the ACSLS XAPI server using the Route command.

Unlike XCMD commands entered from the ACSLS cmd_proc, the SMC/MVS or VM/Client Route command honors established SMC/HSC syntax conventions. Also, the XCMD token is omitted.

For example:

To issue an XAPI QUERY VOLUME request as an ACSLS cmd_proc XCMD command:

```
XCMD QUERY VOLUME VOLSER
```

Where VOLSER is the desired volume.

To issue the XAPI QUERY VOLUME request from an SMC/MVS client to the ACSLS XAPI server using the ROUTE command:

```
ROUTE ACSLS,QUERY VOLUME(volser)
```

Where *volser* is the desired volume.

Refer to the *ELS Command, Control Statement, and Utility Reference* for more information about the SMC ROUTE command.

6

ACSLS XAPI Server Control Variables

Oracle's ACSLS software provides a set of system variables that enable you to control certain behaviors of your ACSLS system. Included are several control variables that are specific to the ACSLS XAPI server component. These variables include the following:

- XAPI_PORT
- XAPI_WORK_PATH
- XAPI_LOG_SIZE
- XAPI_LOG_FILE_NUM
- XAPI_TRACE_SIZE
- XAPI_TRACE_FILE_NUM
- XAPI_STARTUP_FILE
- XAPI_TAPEPLEX_NAME

Setting ACSLS XAPI Server Control Variables

Use the ACSLS `acsss_config` or `dv_config` utility to display, and set the ACSLS XAPI static variables. You must restart ACSLS for the changes to take effect.

- To use the `dv_config` utility, enter one of the following commands:
  - `dv_config -d` to display all ACSLS variables.
  - `dv_config -p <`*`variable_name`*`>` to update the XAPI variable.
- To use the `acsss_config` utility, use the ACSLS `acsss_config` script to access the ACSLS Feature Configuration screen:

```
             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 Configuration
9: Set XAPI server variables
E: Exit
```

Select option 9 (`Set XAPI server variables`).

> **Note:** This option is only visible if you have enabled the ACSLS XAPI server feature.

Refer to the *StorageTek Automated Cartridge System Library Software Administrator's Guide* for more information about the `acsss_config` and `dv_config` utilities and how you can use these utilities to display and update ACSLS variables.

ACSLS XAPI Variable Descriptions

This section describes valid variables used to control the ACSLS XAPI server component.

XAPI_PORT

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

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

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

XAPI_WORK_PATH

Prompt: *Changes to the XAPI work directory will not take effect until the xapi server is restarted. Place the XAPI log and trace information in which directory. [$ACS_HOME/log/xapi]*

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

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

XAPI_LOG_SIZE

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

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

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

XAPI_LOG_FILE_NUM

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

This option specifies the number of archived XAPI Log files to retain.

When the current vlog.file size exceeds the threshold size, the log file is renamed with a 0 to *n* suffix. 0 is the newest file and *n* is the oldest. The archived files are saved in the `API_WORK_PATH` directory.

Once the specified number of archived logs is reached, the oldest file will be removed from the archive directory each time a new one is added to that directory. You can retain no fewer than one archived file and no more than 99.

Enter a number between 1 and 99 to specify the number of archived log files to retain.

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

XAPI_TRACE_SIZE

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

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

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

XAPI_TRACE_FILE_NUM

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

This option specifies the number of archived XAPI TRACE files to retain.

When the current vtrace.file size exceeds the threshold size, the trace file is renamed with a 0 to *n* suffix. 0 is the newest file and *n* is the oldest. The archived files are saved in the `XAPI_WORK_PATH` directory.

Once the specified number of archived logs is reached, the oldest file will be removed from the archive directory each time a new one is added to that directory. You can retain no fewer than one archived file and no more than 99.

Enter a number between 1 and 99 to specify the number of archived log files to retain.

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

XAPI_STARTUP_FILE

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

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

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

XAPI_TAPEPLEX_NAME

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

This option specifies the name of the XAPI Tapeplex. Please enter text whose length does not exceed 8 characters.

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

Index