Yamcs Command-Line Interface

Release 1.4.17

Yamcs
MISSION CONTROL

Space Applications Services, NV/SA
Leuvensesteenweg 325
1932 Sint-Stevens-Woluwe
Belgium
spaceapplications.com
yamcs.org

Aerospace Applications North America, Inc.
16850 Saturn Ln, Ste 100
Houston, TX 77058
United States of America
aerospaceapplications-na.com

Contents

About Yamcs Command-Line Interface

The Yamcs Command-Line Interface is written in Python and distributed separately from Yamcs Server.

Install with pip:

```
pip install --upgrade yamcs-cli
```

Run *yamcs login* (page 18) to initialize your environment:

```
yamcs login
```

1. yamcs

1.1 Synopsis

yamcs [--version] [-h, --help] <*COMMAND*> [<*ARGS*>]

1.2 Description

Command-line interface for Yamcs.

Run `yamcs login` to initialize your environment.

1.3 Options

`--version`
: Prints the CLI version and quit.

`-h, --help`
: Prints the synopsis and the list of commands.

1.4 Commands

alarms **(page 4)**
: Read alarms. See *yamcs-alarms(1)*.

algorithms **(page 6)**
: Read algorithms. See *yamcs-algorithms(1)*.

commands **(page 7)**
: Read commands. See *yamcs-commands(1)*.

config **(page 9)**
: Manage Yamcs CLI properties. See *yamcs-config(1)*.

containers **(page 11)**
: Read containers. See *yamcs-containers(1)*.

dbshell **(page 12)**
: Launch Yamcs DB Shell. See *yamcs-dbshell(1)*.

events **(page 14)**
: Read and create events. See *yamcs-events(1)*.

instances **(page 16)**
: Read Yamcs instances. See *yamcs-instances(1)*.

links **(page 17)**
: Read and manipulate data links. See *yamcs-links(1)*.

2. yamcs alarms

2.1 Synopsis

yamcs alarms list [--processor <*PROCESSOR*>]
[--format <*FORMAT*>]
yamcs alarms acknowledge [--processor <*PROCESSOR*>]
[-c <*COMMENT*>, --comment <*COMMENT*>] <*ALARM*> <*SEQNO*>
yamcs alarms shelve [--processor <*PROCESSOR*>]
[-c <*COMMENT*>, --comment <*COMMENT*>] <*ALARM*> <*SEQNO*>
yamcs alarms unshelve [--processor <*PROCESSOR*>]
[-c <*COMMENT*>, --comment <*COMMENT*>] <*ALARM*> <*SEQNO*>
yamcs alarms clear [--processor <*PROCESSOR*>]
[-c <*COMMENT*>, --comment <*COMMENT*>] <*ALARM*> <*SEQNO*>
yamcs alarms log [-n <*LINES*>, --lines <*LINES*>]
[-s <*DATE*>, --since <*DATE*>] [-u <*DATE*>, --until <*DATE*>]
[--format <*FORMAT*>]

2.2 Description

Manage alarms.

2.3 Commands

`list [--processor <PROCESSOR>] [--format <FORMAT>]`
Show active alarms

`acknowledge [--processor <PROCESSOR>] [-c <COMMENT>, --comment <COMMENT>] <ALARM> <SEQNO>`
Acknowledge an alarm

`shelve [--processor <PROCESSOR>] [-c <COMMENT>, --comment <COMMENT>] <ALARM> <SEQNO>`
Shelve an alarm

`unshelve [--processor <PROCESSOR>] [-c <COMMENT>, --comment <COMMENT>] <ALARM> <SEQNO>`
Unshelve an alarm

`clear [--processor <PROCESSOR>] [-c <COMMENT>, --comment <COMMENT>] <ALARM> <SEQNO>`
Clear an alarm

`log [-n <LINES>, --lines <LINES>] [-s <DATE>, --since <DATE>] [-u <DATE>, --until <DATE>] [--format <FORMAT>]`
Read alarm log

2.4 Options

--processor <PROCESSOR>

With `list`, `acknowledge`, `shelve`, `unshelve` or `clear`, specify the processor.

Default: realtime

-c <COMMENT>, **--comment** <COMMENT>

With `list`, `acknowledge`, `shelve`, `unshelve` or `clear`, attach a comment to the alarm change.

<ALARM>

With `acknowledge`, `shelve`, `unshelve` or `clear`, specify the alarm name.

<SEQNO>

With `acknowledge`, `shelve`, `unshelve` or `clear`, specify the alarm instance.

-n <LINES>, **--lines** <LINES>

With `log`, specify the number of alarms to show, or `all` to show all.

Default: 10, but only when `--since` and `--until` are unset.

-s <DATE>, **--since** <DATE>

With `log`, include alarms not older than the specified date.

The date should be specified in ISO format or as detailed under *Timestamps* (page 5).

-u <DATE>, **--until** <DATE>

With `log`, include alarms not newer than the specified date.

The date should be specified in ISO format or as detailed under *Timestamps* (page 5).

--format <FORMAT>

For subcommands that support it, set the output format to:

table
: Print a human-friendly table

json
: Print in JSON format

2.5 Timestamps

When parsing timestamps, yamcs-cli accepts a specification in ISO format. The following special patterns are also recognised:

- `now`: current time
- `now UTC`: current UTC time
- `today`: 00:00:00 of the current day
- `today UTC`: 00:00:00 UTC of the current day
- `yesterday`: 00:00:00 of the day before
- `yesterday UTC`: 00:00:00 UTC of the day before
- `tomorrow`: 00:00:00 of the next day
- `tomorrow UTC`: 00:00:00 UTC of the next day

3. yamcs algorithms

3.1 Synopsis

yamcs algorithms list [--format <*FORMAT*>]
yamcs algorithms describe <*ALGORITHM*>

3.2 Description

Read algorithms.

3.3 Commands

list
: List algorithms [--format <FORMAT>]

describe <ALGORITHM>
: Describe an algorithm

3.4 Options

--format <FORMAT>
: For subcommands that support it, set the output format to:

  table
  : Print a human-friendly table

  json
  : Print in JSON format

4. yamcs commands

4.1 Synopsis

yamcs commands list [--format <*FORMAT*>]
yamcs commands describe <*COMMAND*>
yamcs commands run [--stream <*STREAM*>] [--processor <*PROCESSOR*>]
[--dry-run] [--sequence-number <*SEQNO*>] [--arg-file <*FILE*>]
[--arg <*KEY=VALUE*> [<*KEY=VALUE*> ...]]
<*COMMAND*>
yamcs commands log [-n <*LINES*>, --lines <*LINES*>]
[-s <*DATE*>, --since <*DATE*>] [-u <*DATE*>, --until <*DATE*>]
[--format <*FORMAT*>]

4.2 Description

Manage commands.

4.3 Commands

`list [--format <FORMAT>]`
List commands

`describe <COMMAND>`
Describe a command

`run <COMMAND>`
Run a command

`log [-n <LINES>, --lines <LINES>] [-s <DATE>, --since <DATE>] [-u <DATE>, --until <DATE>] [--format <FORMAT>]`
Read command log

4.4 Options

`--stream` <STREAM>
With `run`, specifies the name of the target stream.

Default behaviour is for Yamcs to automatically select a stream.

`--processor` <PROCESSOR>
With `run`, specifies the name of the target processor.

Default is `realtime`.

--dry-run

With `run`, validate the command, but do not queue it.

--arg-file `<FILE>`

With `run`, read command arguments from a file

--arg `<KEY=VALUE> [KEY=VALUE ...]`

With `run`, set command arguments.

--sequence-number `<SEQNO>`

With `run`, set the sequence number of this command. This is used to determine unicity of commands at the same time and coming from the same origin. If not set Yamcs will automatically assign a sequential number as if every submitted command is unique.

-n `<LINES>`, **--lines** `<LINES>`

With `log`, specify the number of commands to show, or `all` to show all.

Default: 10, but only when `--since` and `--until` are unset.

-s `<DATE>`, **--since** `<DATE>`

With `log`, include commands not older than the specified date.

The date should be specified in ISO format or as detailed under *Timestamps* (page 8).

-u `<DATE>`, **--until** `<DATE>`

With `log`, include commands not newer than the specified date.

The date should be specified in ISO format or as detailed under *Timestamps* (page 8).

--format `<FORMAT>`

For subcommands that support it, set the output format to:

table
: Print a human-friendly table

json
: Print in JSON format

4.5 Timestamps

When parsing timestamps, yamcs-cli accepts a specification in ISO format. The following special patterns are also recognised:

- `now`: current time
- `now UTC`: current UTC time
- `today`: 00:00:00 of the current day
- `today UTC`: 00:00:00 UTC of the current day
- `yesterday`: 00:00:00 of the day before
- `yesterday UTC`: 00:00:00 UTC of the day before
- `tomorrow`: 00:00:00 of the next day
- `tomorrow UTC`: 00:00:00 UTC of the next day

5. yamcs config

5.1 Synopsis

yamcs config get <*PROPERTY*>
yamcs config list
yamcs config set <*PROPERTY*> <*VALUE*>
yamcs config unset <*PROPERTY*>

5.2 Description

Manage Yamcs CLI properties.

5.3 Commands

`get <PROPERTY>`
: Get value of CLI property

`list`
: List CLI properties

`set <PROPERTY> <VALUE>`
: Set CLI property

`unset <PROPERTY>`
: Unset CLI property

5.4 Configuration File

Configuration properties are stored to the file `$HOME/.config/yamcs-cli/config` and divided in sections. Properties affect the commands' behavior.

Currently all supported properties belong to the `core` section only.

5.4.1 Example

```
[core]
url = http://localhost:8090
instance = simulator
```

5.4.2 Properties

`url`
: Yamcs Server URL

`instance`
: Yamcs instance name

`enable_utc`
: If true, dates are printed in UTC format rather than using the system local timezone.

6. yamcs containers

6.1 Synopsis

yamcs containers list [--format <*FORMAT*>]
yamcs containers describe <*CONTAINER*>

6.2 Description

Read containers.

6.3 Commands

list
: List containers

describe <CONTAINER>
: Describe a container

6.4 Options

--format <FORMAT>
: For subcommands that support it, set the output format to:

  table
  : Print a human-friendly table

  json
  : Print in JSON format

7. yamcs dbshell

7.1 Synopsis

yamcs dbshell [<*OPTIONS*>]

7.2 Description

Launch Yamcs DB Shell.

7.3 Options

-c <COMMAND>, **--command** <COMMAND>
: Run a single SQL command string.

-N, **--skip-column-names**
: Don't print column names.

-B, **--batch**
: Print results using tab as the column seperator.

  Use of this option disables interactive behavior such as prompt display or the history file.

  Any binary fields are printed raw instead of in hexadecimal notation.

--binary-as-hex
: Display binary data using hexadecimal notation. This the default when running in interactive mode, but can be used in combination with *--batch* (page 12) to show hexadecimal values also in batch mode.

7.4 DB Shell Commands

The shell sends each SQL statement that you issue to Yamcs. There is also a set of commands that are interpreted by dbshell itself. For a list of these, type help or h at the shell prompt:

```
simulator> help

List of dbshell commands:
?          (\?) Show help.
delimiter  (\d) Set statement delimiter.
edit       (\e) Edit a command with $EDITOR.
exit       (\q) Synonym for quit.
help       (\h) Display this help.
nopager    (\n) Disable pager. Results are printed to stdout.
pager      (\P) Print results to a pager.
quit       (\q) Quits the DB Shell.
rehash     (\#) Rebuild completion hash.
source     (\.) Execute an SQL script file, provided as argument.
status     (\s) Print status information.
system     (\!) Execute a system command.
use        (\u) Use another instance, provided as argument.
```

help [<COMMAND>], \h [<COMMAND>], ? [<COMMAND>], \? [<COMMAND>]

Display a help message listing all available commands.

If you provide an argument, the help message for that specific command is shown.

delimiter <STRING>, \d <STRING>

Change the string that separates SQL statements. Default is the semicolon character: `;`.

edit, \e

Open an editor for entering the next SQL statement. This uses the editor indicated by the `$EDITOR` environment variable.

nopager, \n

Disable result paging. It is disabled by default.

pager, \P

Enable result paging. It is disabled by default.

quit, \q

Quits the DB Shell.

rehash, \#

Reload database objects, used for completion.

source <FILENAME>, \. <FILENAME>

Run statements from the provided file.

status, \s

Print information on the current state.

system <COMMAND>, \! <COMMAND>

Execute a local command in a subshell.

use <INSTANCE>, \u <INSTANCE>

Switch the prompt to another instance.

8. yamcs events

8.1 Synopsis

yamcs events create [-m <*MESSAGE*>, --message <*MESSAGE*>] [--date <*DATE*>]
[--sequence-number <*SEQNO*>] [--severity <*LEVEL*>] [--source <*SOURCE*>]
[--type <*TYPE*>] [--extra <*KEY=VALUE*> [<*KEY=VALUE*> ...]]
yamcs events log [-n <*LINES*>, --lines <*LINES*>]
[-s <*DATE*>, --since <*DATE*>] [-u <*DATE*>, --until <*DATE*>]
[--filter <*EXPRESSION*>] [--format <*FORMAT*>]

8.2 Description

Add events to the Yamcs event log.

8.3 Commands

`create`
: Create an event. This command shows an editor where you can enter the event message. Alternatively you can specify the message using the `--message` option.

`log [-n <LINES>, --lines <LINES>] [-s <DATE>, --since <DATE>] [-u <DATE>, --until <DATE>] [--filter <EXPRESSION>] [--format <FORMAT>]`
: Read event log

8.4 Options

`-m` `<MESSAGE>`**`, --message`** `<MESSAGE>`
: Event message.

`--date` `<DATE>`
: Event time. If unspecified, defaults to mission time.

  The date should be specified in ISO format or as detailed under *Timestamps* (page 15).

`--sequence-number` `<SEQNO>`
: Sequence number of this event. This is used to determine unicity of events at the same time and coming from the same source. If not set Yamcs will automatically assign a sequential number as if every submitted event is unique.

`--severity` `<LEVEL>`
: The severity level of the event. One of `info`, `watch`, `warning`, `distress`, `critical` or `severe`. Default is `info`.

--source <SOURCE>
: Source of the event. Defaults to `User` if unset.

--type <TYPE>
: Type of the event.

--extra <KEY=VALUE> [KEY=VALUE ...]
: Set additional event properties.

-n <LINES>, **--lines** <LINES>
: With `log`, specify the number of events to show, or `all` to show all.

  Default: 10, but only when `--since` and `--until` are unset.

-s <DATE>, **--since** <DATE>
: With `log`, include events not older than the specified date.

  The date should be specified in ISO format or as detailed under *Timestamps* (page 15).

-u <DATE>, **--until** <DATE>
: With `log`, include events not newer than the specified date.

  The date should be specified in ISO format or as detailed under *Timestamps* (page 15).

--filter <EXPRESSION>
: With `log`, include events that match the filter expression.

--format <FORMAT>
: For subcommands that support it, set the output format to:

  table
  : Print a human-friendly table

  json
  : Print in JSON format

8.5 Timestamps

When parsing timestamps, yamcs-cli accepts a specification in ISO format. The following special patterns are also recognised:

- `now`: current time
- `now UTC`: current UTC time
- `today`: 00:00:00 of the current day
- `today UTC`: 00:00:00 UTC of the current day
- `yesterday`: 00:00:00 of the day before
- `yesterday UTC`: 00:00:00 UTC of the day before
- `tomorrow`: 00:00:00 of the next day
- `tomorrow UTC`: 00:00:00 UTC of the next day

9. yamcs instances

9.1 Synopsis

yamcs instances list [--format <*FORMAT*>]
yamcs instances start <*INSTANCE*>...
yamcs instances stop <*INSTANCE*>...

9.2 Description

Read Yamcs instances.

9.3 Commands

list [--format <FORMAT>]
: List instances

start <INSTANCE>...
: Start an instance

stop <INSTANCE>...
: Stop an instance

9.4 Options

--format <FORMAT>
: For subcommands that support it, set the output format to:

  table
  : Print a human-friendly table

  json
  : Print in JSON format

10. yamcs links

10.1 Synopsis

yamcs links list [--format <*FORMAT*>]
yamcs links enable <*LINK*>...
yamcs links disable <*LINK*>...
yamcs links describe <*LINK*>
yamcs links run-action <*LINK*> <*ACTION*>

10.2 Description

Read and manipulate data links.

10.3 Commands

list [--format <FORMAT>]
: List links

enable
: Enable a link

disable
: Disable a link

describe
: Describe a link

run-action
: Run a custom action

10.4 Options

--format <FORMAT>
: For subcommands that support it, set the output format to:

  table
  : Print a human-friendly table

  json
  : Print in JSON format

11. yamcs login

11.1 Synopsis

yamcs login [-u <*USERNAME*>, --username <*USERNAME*>]
[--instance <*INSTANCE*>] [<*URL*>]

11.2 Description

Login to a Yamcs server.

11.3 Options

<URL>
The server URL. Example: `http://localhost:8090`

-u <USERNAME>, **--username** <USERNAME>
Username

--instance <INSTANCE>
Initial instance.

This defaults to the first available instance on the target Yamcs server.

11.4 Environment

YAMCS_CLI_PASSWORD
Provide a password instead of prompting for input.

12. yamcs logout

12.1 Synopsis

yamcs logout

12.2 Description

Logout of a Yamcs server.

13. yamcs packets

13.1 Synopsis

yamcs packets log [-p <*PACKET*>, --packet <*PACKET*>]
[-n <*LINES*>, --lines <*LINES*>] [-s <*DATE*>, --since <*DATE*>]
[-u <*DATE*>, --until <*DATE*>] [--filter <*EXPRESSION*>]
[--format <*FORMAT*>]
yamcs packets rebuild-histogram [-s <*DATE*>, --since <*DATE*>]
[-u <*DATE*>, --until <*DATE*>]
yamcs packets rebuild-ccsds-index [-s <*DATE*>, --since <*DATE*>]
[-u <*DATE*>, --until <*DATE*>]

13.2 Description

Read packets.

13.3 Commands

`log [-p <PACKET> --packet <PACKET>] [-n <LINES>, --lines <LINES>] [-s <DATE>, --since <DATE>] [-u <DATE>, --until <DATE>] [--filter <EXPRESSION>] [--format <FORMAT>]`

Read packet log

`rebuild-histogram [-s <DATE>, --since <DATE>] [-u <DATE>, --until <DATE>]`

Rebuilds the packet histogram. This may be necessary for example after bulk loading data.

`rebuild-ccsds-index [-s <DATE>, --since <DATE>] [-u <DATE>, --until <DATE>]`

Rebuilds the CCSDS index. This may be necessary for example after bulk loading data.

This method is only applicable when a `CcsdsTmIndex` service is used to calculate completeness.

13.4 Options

`-p <PACKET>, --packet <PACKETS>`

With `log`, filter by packet name.

`-n <LINES>, --lines <LINES>`

With `log`, specify the number of packets to show, or `all` to show all.

Default: 10, but only when `--since` and `--until` are unset.

`-s <DATE>, --since <DATE>`

Include packets not older than the specified date.

The date should be specified in ISO format or as detailed under *Timestamps* (page 21).

`-u` <DATE>, `--until` <DATE>

Include packets not newer than the specified date.

The date should be specified in ISO format or as detailed under *Timestamps* (page 21).

`--filter` <EXPRESSION>

With `log`, include packets that match the filter expression.

`--format` <FORMAT>

For subcommands that support it, set the output format to:

table
Print a human-friendly table

json
Print in JSON format

13.5 Timestamps

When parsing timestamps, yamcs-cli accepts a specification in ISO format. The following special patterns are also recognised:

- `now`: current time
- `now UTC`: current UTC time
- `today`: 00:00:00 of the current day
- `today UTC`: 00:00:00 UTC of the current day
- `yesterday`: 00:00:00 of the day before
- `yesterday UTC`: 00:00:00 UTC of the day before
- `tomorrow`: 00:00:00 of the next day
- `tomorrow UTC`: 00:00:00 UTC of the next day

14. yamcs parameter-archive

14.1 Synopsis

yamcs parameter-archive rebuild [-s <*DATE*>, --since <*DATE*>]
[-u <*DATE*>, --until <*DATE*>]
yamcs parameter-archive purge
yamcs parameter-archive backfilling enable
yamcs parameter-archive backfilling disable

14.2 Description

Manage the Parameter Archive.

14.3 Commands

`rebuild`
: Rebuild the Parameter Archive.

  This operation may be constrained by using the `--since` and `--until` options. These values are only hints to the Parameter Archive, which will extend the requested range based on archive segmentation.

  Rebuild run as an asynchronous operation: this command will not await the outcome.

`purge`
: Remove all data from the Parameter Archive

`backfilling`
: Enable or disable backfilling in the Parameter Archive

14.4 Options

`-s` `<DATE>`, **`--since`** `<DATE>`
: With `rebuild`, date specification in ISO format or as detailed under *Timestamps* (page 22).

`-u` `<DATE>`, **`--until`** `<DATE>`
: With `rebuild`, date specification in ISO format or as detailed under *Timestamps* (page 22).

14.5 Timestamps

When parsing timestamps, yamcs-cli accepts a specification in ISO format. The following special patterns are also recognised:

- `now`: current time
- `now UTC`: current UTC time

- `today`: 00:00:00 of the current day
- `today UTC`: 00:00:00 UTC of the current day
- `yesterday`: 00:00:00 of the day before
- `yesterday UTC`: 00:00:00 UTC of the day before
- `tomorrow`: 00:00:00 of the next day
- `tomorrow UTC`: 00:00:00 UTC of the next day

15. yamcs parameters

15.1 Description

yamcs parameters list [--format <*FORMAT*>]
yamcs parameters describe <*PARAMETER*>
yamcs parameters get [--next [--timeout <*TIMEOUT*>]] <*PARAMETER*>
yamcs parameters set [--date <*DATE*>] <*PARAMETER*> <*VALUE*>
yamcs parameters export-csv [--interval <*INTERVAL*>]
[-s <*DATE*>, --since <*DATE*>] [-u <*DATE*>, --until <*DATE*>] <*PARAMETER*>

15.2 Desription

Manage parameters.

15.3 Commands

`list [--format <FORMAT>]`
List parameters

`describe <PARAMETER>`
Describe a parameter

`get <PARAMETER>`
Get a parameter's value

`set <PARAMETER> <VALUE>`
Set a parameter's value (unless this is a readonly parameter).

`export-csv <PARAMETER>`
Export parameter values in CSV format

15.4 Options

`--processor` `<PROCESSOR>`
With `get` and `set`, specifies the name of the target processor.

Default is `realtime`.

`--next`
With `get`, wait for the next parameter value to be processed by Yamcs.

If not set, the latest received value is returned.

`--timeout` `<TIMEOUT>`
With `get`, this indicates the maximum time to wait for a new value if the `next` option is used.

`--date` <DATE>

Value time. If unspecified, defaults to mission time.

When parsing timestamps, yamcs-cli accepts a specification in ISO format. The following special patterns are also recognised:

- `now`: current time
- `now UTC`: current UTC time
- `today`: 00:00:00 of the current day
- `today UTC`: 00:00:00 UTC of the current day
- `yesterday`: 00:00:00 of the day before
- `yesterday UTC`: 00:00:00 UTC of the day before
- `tomorrow`: 00:00:00 of the next day
- `tomorrow UTC`: 00:00:00 UTC of the next day

`--interval` <INTERVAL>

With `export-csv`, limit values to max one per interval (in seconds).

`-s` <DATE>, `--since` <DATE>

With `export-csv`, include values not older than the specified date.

The date should be specified in ISO format or as detailed under *Timestamps* (page 25).

`-u` <DATE>, `--until` <DATE>

With `export-csv`, include values not newer than the specified date.

The date should be specified in ISO format or as detailed under *Timestamps* (page 25).

`--format` <FORMAT>

For subcommands that support it, set the output format to:

`table`
: Print a human-friendly table

`json`
: Print in JSON format

15.5 Timestamps

When parsing timestamps, yamcs-cli accepts a specification in ISO format. The following special patterns are also recognised:

- `now`: current time
- `now UTC`: current UTC time
- `today`: 00:00:00 of the current day
- `today UTC`: 00:00:00 UTC of the current day
- `yesterday`: 00:00:00 of the day before
- `yesterday UTC`: 00:00:00 UTC of the day before
- `tomorrow`: 00:00:00 of the next day
- `tomorrow UTC`: 00:00:00 UTC of the next day

16. yamcs processors

16.1 Synopsis

yamcs processors list [--format <*FORMAT*>]
yamcs processors delete <*PROCESSOR*>

16.2 Description

Manage processors.

16.3 Commands

list [--format <FORMAT>]
: List processors

delete <PROCESSOR>
: Delete a processor

16.4 Options

--format <FORMAT>
: For subcommands that support it, set the output format to:

  table
  : Print a human-friendly table

  json
  : Print in JSON format

17. yamcs rocksdb

17.1 Synopsis

yamcs rocksdb tablespaces
yamcs rocksdb compact [--dbpath <*DBPATH*>] <*TABLESPACE*> <*CF*>

17.2 Description

Manage RocksDB storage engine.

17.3 Commands

`tablespaces`
: List processors

`compact [--dbpath <DBPATH>] <TABLESPACE> <CF>`
: Delete a processor

17.4 Options

`--dbpath` <DBPATH>
: With `compact`, specify a path within the tablespace. Leave unspecified for the root database.

18. yamcs services

18.1 Synopsis

yamcs services list [--format <*FORMAT*>]
yamcs services start <*SERVICE*>...
yamcs services stop <*SERVICE*>...

18.2 Description

Read and manipulate services.

18.3 Commands

`list [--format <FORMAT>]`
: List services

`start <SERVICE>...`
: Start a service

`stop <SERVICE>...`
: Stop a service

18.4 Options

`--format` <FORMAT>
: For subcommands that support it, set the output format to:

  `table`
  : Print a human-friendly table

  `json`
  : Print in JSON format

19. yamcs space-systems

19.1 Synopsis

yamcs space-systems list [--format <*FORMAT*>]
yamcs space-systems describe <*SPACESYSTEM*>
yamcs space-systems export [--xtce-version <*VERSION*>] <*SPACESYSTEM*>

19.2 Description

Read space systems.

19.3 Commands

list [--format <FORMAT>]
: List space systems

describe <SPACESYSTEM>
: Describe a space system

export <SPACESYSTEM>
: Export an XTCE representation of a space system

19.4 Options

--xtce-version <VERSION>
: XTCE version. One of `1.2` or `1.3`.

  Default is `1.2`

--format <FORMAT>
: For subcommands that support it, set the output format to:

  table
  : Print a human-friendly table

  json
  : Print in JSON format

20. yamcs storage

20.1 Synopsis

yamcs storage ls [-l] [-r, -R] [<*BUCKET*>]
yamcs storage mb <*BUCKET*>...
yamcs storage rb <*BUCKET*>...
yamcs storage cat <*OBJECT*>...
yamcs storage cp <*SRC*> <*DST*>
yamcs storage mv <*SRC*> <*DST*>
yamcs storage rm <*OBJECT*>...

20.2 Description

Manage object storage.

20.3 Commands

`ls [-l] [-r, -R] [<BUCKET>]`

List buckets or objects
Synonym: `list`

`mb <BUCKET>...`

Make buckets

`rb <BUCKET>...`

Remove buckets

`cat <OBJECT>...`

Concatenate object content to stdout

`cp <SRC> <DST>`

Copy a file or object

`mv <SRC> <DST>`

Move a file or object

`rm <OBJECT>...`

Remove objects

20.4 Options

`-l`

With `ls`, list in long format.

`-r, -R`
: With `ls`, list recursively.

`<SRC>`
: With `cp`, local file or an object in the format ys://BUCKET/OBJECT.

`<DST>`
: With `cp`, local file, or an object in the format ys://BUCKET/OBJECT.

  With `mv`, local file, local directory, or an object in the format ys://BUCKET/OBJECT.

21. yamcs streams

21.1 Synopsis

yamcs streams list [--format <*FORMAT*>]
yamcs streams describe
yamcs streams subscribe

21.2 Description

Read and manipulate streams.

21.3 Commands

list [--format <FORMAT>]
: List streams

describe
: Describe a stream

subscribe
: Subscribe to a stream

21.4 Options

--format <FORMAT>
: For subcommands that support it, set the output format to:

  table
  : Print a human-friendly table

  json
  : Print in JSON format

22. yamcs tables

22.1 Synopsis

yamcs tables list [--format <*FORMAT*>]
yamcs tables describe <*TABLE*>
yamcs tables dump [-d <*DIR*>, --dir <*DIR*>] [--gzip]
[-q <*QUERY*>, --query <*QUERY*>] [--query-file <*FILE*>]
<*TABLE*>...
yamcs tables load [-d <*DIR*>, --dir <*DIR*>] [--gzip] <*TABLE*>...
yamcs tables rebuild-histogram [-s <*DATE*>, --since <*DATE*>]
[-u <*DATE*>, --until <*DATE*>] <*TABLE*>...

22.2 Description

Read and manipulate tables.

22.3 Commands

`list [--format <FORMAT>]`
List tables

`describe <TABLE>`
Describe a table

`dump [-d <DIR>, --dir <DIR>] [-q <QUERY>, --query <QUERY>] [--query-file FILE] [--gzip] <TABLE>...`
Dump table data

`load [-d <DIR>, --dir <DIR>] [--gzip] <TABLE>...`
Load data into a table

`rebuild-histogram [-s <DATE>, --since <DATE>] [-u <DATE>, --until <DATE>] <TABLE>...`
Rebuilds the histogram for a table. This may be necessary for example after bulk loading data.

22.4 Options

`-d` `<DIR>`, **`--dir`** `<DIR>`
Specifies the directory where to locate dump files. Defaults to current directory.

`--gzip`
With `dump`, compress the output.

With `load`, decompress the dump.

-q <QUERY>, **--query** <QUERY>

With dump, provide a SQL WHERE search condition to limit the rows included in the output.

--query-file <FILE>

With dump, specify the path to a file containing a SQL WHERE search condition to limit the rows included in the output.

-s <DATE>, **--since** <DATE>

With rebuild-histogram, include records not older than the specified date.

The date should be specified in ISO format or as detailed under *Timestamps* (page 34).

-u <DATE>, **--until** <DATE>

With rebuild-histogram, include records not newer than the specified date.

The date should be specified in ISO format or as detailed under *Timestamps* (page 34).

--format <FORMAT>

For subcommands that support it, set the output format to:

table
: Print a human-friendly table

json
: Print in JSON format

22.5 Timestamps

When parsing timestamps, yamcs-cli accepts a specification in ISO format. The following special patterns are also recognised:

- now: current time
- now UTC: current UTC time
- today: 00:00:00 of the current day
- today UTC: 00:00:00 UTC of the current day
- yesterday: 00:00:00 of the day before
- yesterday UTC: 00:00:00 UTC of the day before
- tomorrow: 00:00:00 of the next day
- tomorrow UTC: 00:00:00 UTC of the next day

Index

Symbols

Y