

mysqlsla v2 Documentation

This document outlines and explains every [mysqlsla](#) command line option. Where necessary, more detailed explanations are referred to other documents because some options entail rather lengthy explanations. Also covered in this document are known limitations, future features, and bugs (and what to do about them).

This documentation assumes that you have already a working knowledge of mysqlsla. Therefore, it serves more as a reference than a guide. For a general introduction to mysqlsla, read [mysqlsla v2 Guide](#).

mysqlsla v2 Documentation Table of Contents

- » [mysqlsla v2 Documentation - Synopsis](#)
- » [.mysqlsla Config File](#)
- » [Command Line Options](#)
 - ... [--log-type \(-lt\) TYPE LOGS](#)
- » [Limitations](#)
- » [Future Features](#)
- » [Bugs](#)

.mysqlsla Config File

[« Top](#)

Reading ~/.mysqlsla is the very first thing mysqlsla does when it starts. Command line options can be set in this file. Example:

```
atomic-statements
statement-filter=+UPDATE,INSERT
```

Notice: no leading dashes (- or --), no quotations marks ("), and the form option=value when option requires a value.

These options are overridden by those given on the real command line.

Command Line Options

[« Top](#)

Command line options are in the form --option but -option works too.

All options can be abbreviated if the abbreviation is unique. For example, --top can be abbreviated -to but not -t because -t is ambiguous: it could mean --top or --time-each-query.

Some options have explicit aliases, such as --databases which has aliases -db and -D. These explicit aliases are noted in parentheses after the full option name like --databases (-db) (-D).

The most important/frequently used command line option is --log-type. Therefore, it is listed first. The rest are listed in alphabetical order.

--log-type (-lt) TYPE LOGS

[« Top](#)

Parse MySQL LOGS of TYPE. Default none. TYPE must be either slow, general, binary, msl or udl. LOGS is a space-separated list of MySQL log files.

This is the most important/frequently used mysqlsla option and is almost always required because mysqlsla must be told which type of MySQL to parse. As of mysqlsla v2.03, this option is optional; not giving it will cause mysqlsla to automatically detect the log type using the first log file given.

Slow and general log types are what you expect.

msl logs are [microslow patched](#) slow logs.

udl means user-defined log. See [User-Defined Logs](#).

Binary logs are a special case. mysqlsla *cannot* read MySQL binary log directly. You must first "decode" the binary log using [mysqlbinlog](#) *without* the --short-form option. It is only the text output from [mysqlbinlog](#) that mysqlsla can read.

If you want to use the --short-form option with [mysqlbinlog](#) you must then use TYPE udl with mysqlsla.

LOGS can also be - to make mysqlsla read from STDIN. This can be used to pipe the output of another program into mysqlsla. Doing this requires that you give the log type explicitly; mysqlsla cannot automatically detect the log type from STDIN.

Example: `mysqlbinlog bin_log.000001 | mysqlsla -lt binary -`

Example: `zcat slow_log.gz | mysqlsla -lt slow -`

--abstract-in (-Ai) N

[« Top](#)

Abstract IN () clauses further by grouping in groups of N. Disabled by default.

This is a somewhat experimental option. Normally, all IN clauses are condensed as IN (N, N, N) → IN (N3). This option furthers this abstraction by grouping the condensed IN clauses in groups of N where N is the "dividing line."

Example: with -Ai 10, IN (N3) becomes IN (N0-9). Therefore, any IN clause with 0 to 9 values will be condensed and then further abstracted to IN (N0-9). Likewise, any IN clauses with 10 to 19 values will be condensed and further abstracted to IN (10-19).

--abstract-values (-Av)

[« Top](#)

Abstract VALUES () clauses further by removing the number of condensed value sets. Disabled by default.

This is a somewhat experimental option. Normally, all VALUES clauses are condensed as VALUES (NULL, 'foo'), (NULL, 'bar') → VALUES (NULL, 'S')². This option furthers this abstraction by removing that number of condensed value sets: 2.

Example: with -Av, two queries INSERT INTO table VALUES ('S') and INSERT INTO table VALUES ('S'), ('S'), ('S') are first condensed to INSERT ... VALUES ('S')¹ and INSERT ... VALUES ('S')³ then further abstracted to one single query: INSERT INTO table VALUES ('S').

--atomic-statements

[« Top](#)

Treat multi-statement groups atomically when filtering. Disabled by default.

This is a somewhat experimental option. Normally, each statement in a multi-statement group is filtered individually: only those which fail a filter are removed and those which pass are kept. With this option enabled, if any one statement in a group fails, the entire group of statements is removed.

This option does not apply to general logs because general logs never group statements.

--avg (-n) N

[« Top](#)

Average query timing over N executions. Default 1.

For option [time-each-query](#) and the [time-all report](#), this option sets the number of execution times for the query or queries in order to obtain an average.

--databases (-db) (-D) DATABASES

[« Top](#)

Try EXPLAINing queries which have no known database using DATABASES. Default none. DATABASES is a comma-separated list of database names (without spaces). Only used when option [explain](#) is used too.

The [explain](#) option naturally requires knowing against which database a query should be EXPLAINed. Some MySQL logs reliably provide this information, such as general logs; other logs do not, such as user-defined logs.

In order to EXPLAIN queries with unknown databases, mysqlsla will try the query against all given DATABASES. The first database that works (which does not cause MySQL to return an error) is used.

--db-inheritance

[« Top](#)

Allow queries to inherit the last database specified in the log. Disabled by default.

Normally, the log must explicitly specify the database for each statement. Or, in the case of general logs, the current database is tracked by other means. Sometimes, however, logs only specify the database explicitly once. If this option is enabled, all statements following an explicit database specification inherit that database.

--debug

[« Top](#)

Enable a *flood* of debugging information from both mysqlsla and MySQL::Log::ParseFilter. Disabled by default. Use with caution.

--dist

[« Top](#)

Calculate distribution frequency of values. Disabled by default. Requires an appropriate [standard report](#) format line.

For meta-properties with all values saved, a distribution of those values is calculated in order to show the frequency of common values.

Example: a slow query has 10 different time values: 3, 3, 4, 3, 10, 18, 4, 2, 3, 3. The distribution frequency of these values is: 50%:3 20%:4 10%:10 10%:18: 10%:2. That means: fifty percent of the time the query took 3 seconds to execute; twenty percent of the time it took 4 seconds; ten percent of the time it took 10 seconds; another ten percent it took 18; and finally another ten percent it took 2 seconds to execute.

In all cases, for every query and every meta-property, dist values will not be calculated if the query occurred only once (c_sum == 1).

--dist-min-percent (-dmin) N

[« Top](#)

Do not display [dist](#) percents less than N. Default 5.

--dist-top (-dtop) N

[« Top](#)

Display at most N [dist](#) percentages. Default 10.

`--dont-save-meta-values`[« Top](#)

Do not save meta-property values from log. Disabled by default (meta-property values are saved).

Enabling this option causes mysqlsla to save only the most basic meta-property values (a sample of the query, its database, and count-related meta-properties). The log can still be filtered by non-aggregate meta-property values (those *not* ending in `_min`, `_max`, `_avg` or `_sum`) using the [meta filter](#).

`--explain (-ex)`[« Top](#)

EXPLAIN each query. Disabled by default. Requires an appropriate [standard report](#) format line.

This option causes mysqlsla to EXPLAIN each unique query. The full output from EXPLAIN is saved and re-printed in the [standard report](#).

`--extra (-x) TINFO`[« Top](#)

Display extra table information. TINFO can be either `tschema`, `tcount` or both like `tschema,tcount`. Default none.

`tschema` will cause mysqlsla to print each tables structure (the output of SHOW CREATE TABLE).

`tcount` will cause mysqlsla to print the number of rows in each table using `SELECT COUNT(*) FROM table`.

`--flush-qc`[« Top](#)

Flush query cache before query execution timing. Disabled by default.

The user with which mysqlsla is made to connect to MySQL must have sufficient privileges to flush the query cache. Otherwise, the flush will fail silently and mysqlsla will report no error.

`--grep PATTERN`[« Top](#)

grep statements for PATTERN and keep only those which match. Default none.

PATTERN is a [Perl regular expressions](#) pattern without `m//`.

`--help (-?)`[« Top](#)

Print basic mysqlsla command line options to screen and exit.

`--host ADDRESS`[« Top](#)

Connect to MySQL at host ADDRESS. Default localhost if no [socket](#) is available.

`--meta-filter (-mf) CONDITIONS`[« Top](#)

Set meta-properties filter using CONDITIONS. Default none. CONDITIONS is a comma-separated list of meta-property conditions (without spaces) in the form: `[meta][op][value]`.

`[meta]` refers to a meta-property name, the list of which is long: see [mysqlsla v2 Filters](#).

`[op]` is either `>` `<` or `=`. `[value]` is the value, numeric or string, against which the value for `[meta]` from the log must be true according to `[op]`.

For string-based `[meta]`, like `db` or `host`, `[op]` can only be `=`.

Meta-property filters are all positive and inclusive: *all* conditions must pass for the statement to pass. A [future feature](#) may introduce negative conditions.

Multiple conditions on the same `[meta]` are allowed, even if they are logically impossible.

`--microsecond-symbol (-us) STRING`[« Top](#)

Use STRING to denote microsecond values. Default `µs`.

This option affects values from a [standard report](#) which are microsecond formatted. mysqlsla uses character number 230 (decimal) on Windows and 181 on everything else to denote the micro symbol: `µ`. Sometimes, however, this symbol confuses programs like `less`.

`--no-mycnf`[« Top](#)

Do not read `~/my.cnf` when initializing. Does not apply to Windows servers.

```
--nthp-min-values (nthpm) N
```

[« Top](#)

Do not calculate **Nth percent values** if there are less than N values. Default 10.

This value applies to each query and each meta-property individually. For example, if one query has 10 time values, **Nth percent** will be calculated. And if another query has only 9 time values, **Nth percent** will not be calculated.

In all cases, for every query and every meta-property, **Nth percent values** will not be calculated if the query occurred only once (`c_sum == 1`).

```
--nth-percent (-nthp) N
```

[« Top](#)

Calculate Nth percent values. Disabled by default or 95 if used but no N is given. Requires an appropriate **standard report** format line.

For meta-properties with all values saved, the total, min, max and average values are re-calculated for only the Nth percent of values on the low (small) end. In other words, the top (100 - N) percent of values are removed.

This is useful to see more realistic total, min, max and average values when a one-time event, like a backup running, caused queries to take longer than normal.

```
--password PASS
```

[« Top](#)

Use PASS as MySQL **user** password. If PASS is omitted, the password will be prompted for (on STDERR).

```
--percent
```

[« Top](#)

Display a basic percentage complete indicator while timing all queries for the **time-all report**. Disabled by default.

```
--port PORT
```

[« Top](#)

Connect to MySQL on PORT. Default none (relies on system default which will be 3306).

```
--post-analyses-replay FILE
```

[« Top](#)

Save a post-analyses replay as FILE. See [mysqlsla v2 Replays](#).

```
--post-parse-replay FILE
```

[« Top](#)

Save a post-parse replay as FILE. See [mysqlsla v2 Replays](#).

```
--post-sort-replay FILE
```

[« Top](#)

Save a post-sort replay as FILE. See [mysqlsla v2 Replays](#).

```
--replay FILE
```

[« Top](#)

Load unique queries from replay FILE. Default none. See [mysqlsla v2 Replays](#).

```
--report-format (-rf) FILE
```

[« Top](#)

Use FILE to format the standard report. Default internal report format.

See [mysqlsla v2 Reports](#) for information on what report format files are and how to create your own.

```
--reports (-R) REPORTS
```

[« Top](#)

Print REPORTS. Default standard. REPORTS is a comma-separated list of report names (without spaces).

Available reports are: standard, time-all, print-unique, print-all, dump. See [mysqlsla v2 Reports](#).

WARNING: A safety **SQL statement filter** of "+SELECT,USE" is automatically set when using **time-each-query** or the **time-all report**. Overriding the safety **SQL statement filter** by explicitly setting another with **statement-filter** can permit *real changes* to databases. Use with caution!

```
--save-all-values
```

[« Top](#)

Save extra "all values" for some meta-properties. Disabled by default.

For binary logs this causes all execution time (ext) values to be saved, and for microslow (msl) logs all iorops, iorbytes, iorwait, reclwait, qwait and pages values. (See [mysqlsla v2 Filters](#).)

This option is useful mainly for msl logs when you want also to calculate [Nth percent values](#) for those meta-properties. It is disabled by default to save 6 arrays worth of memory and the computational overhead of calculating [Nth percent values](#).

This option does not affect user-defined logs and full aggregate values.

```
--silent
```

[« Top](#)

Do not print any [reports](#). Disabled by default. [Debug](#) messages will still be printed.

This option is used primarily when making [replays](#) to suppress the standard report.

```
--socket SOCKET
```

[« Top](#)

Connect to MySQL through SOCKET. Default none (relies on system default which is compiled into the MySQL client library).

```
--sort META
```

[« Top](#)

Sort queries according to META. Default t_sum for slow and msl logs, c_sum for all others. META is any meta-property name; see [mysqlsla v2 Filters](#).

mysqlsla currently does not check that the meta-property name META actually exists. Therefore, if a non-existent meta-property name is given, mysqlsla will print copious errors.

```
--statement-filter (-sf) CONDITIONS
```

[« Top](#)

Set SQL statement filter using CONDITIONS. Default none. CONDITIONS is a comma-separated list of [SQL statement types](#) in the form: [+][TYPE],[TYPE],etc.

The [+] is given only once before the first [TYPE]. A + indicates a positive filter: keep only SQL statements of [TYPE]. A - indicates a negative filter: remove only SQL statements of [TYPE]. If neither is given, - is default.

[TYPE] is a [SQL statement type](#): SELECT, CREATE, DROP, UPDATE, INSERT, etc.

```
--time-each-query (-te)
```

[« Top](#)

Time each query by actually executing it on the MySQL server. Disabled by default. Requires an appropriate [standard report](#) format line.

This option causes mysqlsla to actually execute each unique query on the MySQL server and record its effective execution time. This is not the execution time as reported by MySQL but rather the time required by Perl and DBI to execute the query. The amount of overhead is insignificant.

WARNING: A safety [SQL statement filter](#) of "+SELECT,USE" is automatically set when using [time-each-query](#) or the [time-all report](#). Overriding the safety [SQL statement filter](#) by explicitly setting another with [statement-filter](#) can permit *real changes* to databases. Use with caution!

```
--top N
```

[« Top](#)

After [sorting](#) display only the top N queries. Default 10.

```
--udl-format (-uf) FILE
```

[« Top](#)

Use FILE to define the format of the user-defined log (udl) instead of the default. Default is ";\n" record separator and no headers.

It is almost always necessary to use this option with the udl [log type](#) because the default is only useful for logs that are flat lists of semicolon newline separated SQL statements and nothing else. To have mysqlsla read headers, save meta-property values, etc., it necessary to define the format of the udl. This topic is covered in [User-Defined Logs](#).

```
--user USER
```

[« Top](#)

Connect to MySQL as USER. Default user of mysqlsla process.

Limitations

[« Top](#)

- Abstracting queries to determine unique queries is not an exact science. Unique queries are, at best, "unique enough."
- Not all [meta filters](#) are applied to the queries when using the [time-all report](#). Generally, none of the aggregate

filters (min, max, avg, grand totals, etc.) are applied.

- Database inheritance only applies to new queries. If `SELECT * FROM foo;` already uses database `foo`, then `USE bar;` is read, `SELECT * FROM foo;` will not inherit `bar`. Bug?
- `UPDATEs` are not transformed into `SELECTs` in order to `EXPLAIN` them.
- Calling an SP is untested. It may or may not work.
- No check to see if **sort META** is valid. It's possible but not pretty to do so.
- All users cannot be recreated from **replays**.
- **Replays** only save queries, not grand totals or all users (but grand totals can be **recalculated**).
- Cannot merge replays with replays, only replays with logs.

Future Features

[« Top](#)

- Always trying to gracefully overcome any limitation
- Optimize `_all` arrays to avoid thousands of useless 0 vals while simultaneously calc'ing dist
- Read logs from `general_log` and `slow_log` tables in the MySQL database
- Negative meta filters
- `-Ai` like grouping for `-dist: 5s, 6s` → `5-6s`
- Analyses after sort (to avoid needlessly analyzing queries which will not make the `-top N`)
- Separate `mysqlsla` debug info from that of `MySQL::Log::ParseFilter`
- Grand totals as either "what is seen" (what passes filtering) or true log-wide grand total (considering every query regardless of filtering)
- Ability to merge replays with replays
- Option to split multi-statements and save each individually
- Calc total queries while parsing/saving stmts to prevent first `q_h` loop in `calc_final_sums()`

Bugs

[« Top](#)

I follow the zero known bugs release policy in releasing new versions of `mysqlsla`. Certainly, however, bugs still exist somewhere. So when you find one, send me a **message** with a copy of its output including any error messages. Expect that I will ask for at least a part of your log file because it is easiest to debug `mysqlsla` with the log file that is causing it to break.