

NAME

mysqlreport – Makes a friendly report of important MySQL status values

SYNTAX

mysqlreport [*options*]

DESCRIPTION

mysqlreport makes a friendly report of important MySQL status values. Actually, it makes a friendly report of nearly every status value from SHOW STATUS. Unlike SHOW STATUS which simply dumps over 100 values to screen in one long list, mysqlreport interprets and formats the values and presents the basic values and many more inferred values in a human-readable format. Numerous example reports are available at the mysqlreport web page at <http://hackmysql.com/mysqlreport>.

The benefit of mysqlreport is that it allows you to very quickly see a wide array of performance indicators for your MySQL server which would otherwise need to be calculated by hand from all the various SHOW STATUS values. For example, the Index Read Ratio is an important value but it's not present in SHOW STATUS; it's an inferred value (the ratio of Key_reads to Key_read_requests).

This documentation outlines all the command line options in mysqlreport, most of which control which reports are printed. This document does not address how to interpret these reports; that topic is covered in the document Guide To Understanding mysqlreport at <http://hackmysql.com/mysqlreportguide>.

OPTIONS

Technically, command line options are in the form --option, but -option works too. All options can be abbreviated if the abbreviation is unique. For example, option --host can be abbreviated --ho but not --h because --h is ambiguous: it could mean --host or --help.

--help Output help information and exit.

--user USER

--password

As of version 2.3 --password can take the password on the command line like "--password FOO". Using --password alone without giving a password on the command line causes mysqlreport to prompt for a password.

--host ADDRESS

--port PORT

--socket SOCKET

--no-mycnf

--no-mycnf makes mysqlreport not read ~/.my.cnf which it does by default otherwise. --user and --password always override values from ~/.my.cnf.

--dtq Print Distribution of Total Queries (DTQ) report (under Total in Questions report). Queries (or Questions) can be divided into four main areas: DMS (see --dms below), Com_ (see --com below), COM_QUIT (see COM_QUIT and Questions at http://hackmysql.com/com_quit), and Unknown. --dtq lists the number of queries in each of these areas in descending order.

- dms** Print Data Manipulation Statements (DMS) report (under DMS in Questions report). DMS are those from the MySQL manual section 13.2. Data Manipulation Statements. (Currently, mysqlreport considers only SELECT, INSERT, REPLACE, UPDATE, and DELETE.) Each DMS is listed in descending order by count.

- com N**
Print top N number of non-DMS Com_ status values in descending order (after DMS in Questions report). If N is not given, default is 3. Such non-DMS Com_ values include Com_change_db, Com_show_tables, Com_rollback, etc.

- sas** Print report for Select_ and Sort_ status values (after Questions report). See MySQL Select and Sort Status Variables at <http://hackmysql.com/selectandsort>.

- tab** Print Threads, Aborted, and Bytes status reports (after Created temp report). As of mysqlreport v2.3 the Threads report reports on all Threads_ status values.

- qcache**
Print Query Cache report.

- all** Equivalent to "--dtq --dms --com 3 --sas --qcache". (Notice --tab is not invoked by --all.)

- infile FILE**
Instead of getting SHOW STATUS values from MySQL, read values from FILE. FILE is often a copy of the output of SHOW STATUS including formatting characters (|, +, -). mysqlreport expects FILE to have the format " value number " where value is only alpha and underscore characters (A-Z and _) and number is a positive integer. Anything before, between, or after value and number is ignored. mysqlreport also needs the following MySQL server variables: version, table_cache, max_connections, key_buffer_size, query_cache_size. These values can be specified in INFILE in the format "name = value" where name is one of the aforementioned server variables and value is a positive integer with or without a trailing M and possible periods (for version). For example, to specify an 18M key_buffer_size: key_buffer_size = 18M. Or, a 256 table_cache: table_cache = 256. The M implies Megabytes not million, so 18M means 18,874,368 not 18,000,000. If these server variables are not specified the following defaults are used (respectively) which may cause strange values to be reported: 0.0.0, 64, 100, 8M, 0.

- outfile FILE**
After printing the report to screen, print the report to FILE too. Internally, mysqlreport always writes the report to a temp file first: /tmp/mysqlreport.PID on *nix, c:\sqlreport.PID on Windows (PID is the script's process ID). Then it prints the temp file to screen. Then if --outfile is specified, the temp file is copied to OUTFILE. After --email (below), the temp file is deleted.

- email ADDRESS**
After printing the report to screen, email the report to ADDRESS. This option requires sendmail in /usr/sbin/, therefore it does not work on Windows. /usr/sbin/sendmail can be a sym link to qmail, for example, or any MTA that emulates sendmail's -t command line option and operation. The FROM: field is "mysqlreport", SUBJECT: is "MySQL status report".

- flush--status**
Execute a "FLUSH STATUS;" after generating the reports. If you do not have permissions in MySQL to do this an error from DBD::mysql::st will be printed after the reports.

AUTHORS

Daniel Nichter

If mysqlreport breaks, send me a message from <http://hackmysql.com/feedback> with the error.

SEE ALSO

mytop(1)

The comprehensive Guide To Understanding mysqlreport at <http://hackmysql.com/mysqlreportguide>.