mysqlreport Documentation

mysqlreport makes an easy-to-read report of important **MySQL status values**. Unlike SHOW STATUS which simply dumps over 100 values to screen in one long list, mysqlreport interprets, formats, and then nicely presents the values in report readable by humans. Numerous example reports are available at the **mysqlreport web page**.

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 is not present in SHOW STATUS; it is an inferred value (the ratio of Key_reads to Key_read_requests).

This documentation outlines all the command line options for mysqlreport. Understanding all the values of a mysqlreport report is covered in **The Guide To Understanding mysqlreport**.

NOTE: As of mysqlreport v3.3, many options were removed because the old --all command line option is now enabled by default. All reports that mysqlreport can create are automatically created.

A Note About 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.

Options

--user USER --password --host ADDRESS --port PORT --socket SOCKET --no-mycnf --help	These options mimic most standard applications. --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. --no-mycnf makes mysqlreport not read ~/.my.cnf which it does by default otherwise. --user and --password always override values from ~/.my.cnf.
--infile FILE	Instead of getting SHOW STATUS values from MySQL, read values from FILE. FILE should contain two things: one, the output of SHOW STATUS (with or without formatting characters such as |, + and -), and two, the output of SHOW VARIABLES. This option is useful for making "offline" reports, often from other people's servers such as those who post their SHOW STATUS and SHOW VARIABLES to forums when looking for help with their MySQL server. For the status values, mysqlreport can read any normal output of SHOW STATUS. In general, it expects three things while reading these values: one, that the values begin with Aborted_clients and end with Uptime; two, that the status value names are separated from their values by only white space characters (spaces, tabs) or |; three, that the status values are integers. For the **system variable values**, there are two methods to include these. Beginning with mysqlreport v3.5, the simplest method is to include the whole output from SHOW VARIABLES. Using this method, mysqlreport expects three things while reading these values: one, that the values begin with backlog and end with wait_timeout; two, that the variable names are separated from their values by only white space characters (spaces, tabs) or |; three, that the variable names are alphanumeric, including periods. The second, older method is to include the following system variable values manually: version, table_cache, max_connections, key_buffer_size, query_cache_size, thread_cache_size, tmp_table_size, log_slow_queries, long_query_time. These values must be added manually at the beginning or end of FILE, one per line, 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). (**NOTE:** log_slow_queries is an exception. Use 1 for ON and 0 for OFF.) For example, to specify an 18M key_buffer_size and a 256 table_cache: key_buffer_size = 18M 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, 0, 0, ?, ?. **NOTE:** For MySQL servers version 5.1.3 and newer, even though the system variable table_cache was renamed to table_open_cache, still use table_cache in an infile.
--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. Then it prints the temp file to screen. Then, if --outfile is specified, the temp file is copied to OUTFILE. After --email, 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 on HOST" where HOST is the host that the mysqlreport was ran on (from the --host option, or "localhost" by default).
--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.
--relative (-r) X	The report that mysqlreport normally generates shows values for the entire uptime of the MySQL server. The --relative option causes mysqlreport to generate reports which are relative to previous reports. If the --relative option is given an integer value for X, then mysqlreport will generate relative reports from the MySQL server which are X seconds apart. The number of reports is controlled by the --report-count option. The default is 1 relative report. For example, given the option --relative 60, mysqlreport will generate 2 reports: the first report will be generated immediately; this is the beginning or baseline report. The second report will be generated 60 seconds later. The values of the second report will be relative to the beginning report. For example, assume that in the beginning report there were 10.00k Questions Total. Then assume that during the 60 second interval the MySQL server answered another 1k Questions. The second report will then show 1.00k Questions Total, not 11.00k.

	If the --relative option is given a list of infiles (like those used for the --infile option), then mysqlreport will generate relative reports from only the given infiles in the order that the infiles are given. Infiles names in the given list should be separated with a space; for example: file1 file2 etc. It is important to give the infiles on the command line in the proper order of time: older infiles first. The first infile should have the system variable values (either from SHOW VARIABLES or included manually). An infile can have one or more sets of MySQL SHOW STATUS values. Note, however, that output from "mysqladmin -r -i N extended" will not work because mysqladmin with the -r option already relativizes the SHOW STATUS values. Because mysqlreport writes reports to temp file first, when --relative is used with an integer (not with infiles), mysqlreport will says which temp file it is writing to. Then one can watch the progress of relative reports as they are written.
--report-count (-c) N	Generate N number of relative reports. This option only works with --relative when --relative is given an integer value for X. mysqlreport actually generates N + 1 reports: the first report is the beginning or baseline report. Then, N number of relative reports are generated.
--detach	This option causes mysqlreport to fork, detach from the terminal, and continue running in the background. After forking, mysqlreport will say which temp file it is writing to. This option also needs either option --outfile or --email. If neither --outfile nor --email are given, the report is simply deleted because, since mysqlreport detached from the terminal, the report cannot be printed to the terminal. This option is meant to be used with option --relative so that mysqlreport can collect relative reports over long periods of time without having to have a controlling terminal (i.e. a user logged in). For example, one can capture a relative report over an hour interval and have the whole report emailed to their self by running mysqlreport like: mysqlreport -r 3600 -detach -email host@domain.com After an hour, mysqlreport will email the relative report, remove its temporary files, and terminate cleanly.
--debug	Print debugging information.

What To Do About Bugs and Errors

Please **report** any bugs or strange behavior.

(Doc rev: April 16 2008)