

# Open Tools for MySQL Administrators

[MySQL](#) provides some tools to monitor and troubleshoot a MySQL server, but they don't always suit a MySQL developer or administrator's common needs, or may not work in some scenarios, such as remote or over-the-web monitoring. Fortunately, the MySQL community has created a variety of free tools to fill the gaps. On the other hand, many of these are hard to find via web searches. In fact, web searches can be frustrating because they uncover abandoned or special-purpose, not ready-to-use projects. You could spend hours trying to find tools for monitoring and troubleshooting your MySQL servers. What's a tool-seeker to do?

Relax! I've already done the work, so you won't have to. I'll point you to the tools I've actually found useful. At the end of this article I'll also list those I didn't find helpful.

This article is about tools to discover and monitor the state of your server, so I won't discuss programs for writing queries, designing tables, and the like. I'm also going to focus exclusively on free and open source software.

## Tools to Monitor Queries and Transactions

The classic tool for monitoring queries is Jeremy Zawodny's `mytop`. It is a Perl program that runs in a terminal and displays information about all connections in a tabular layout, similar to the Unix `top` program's process display. Columns include the connection ID, the connection's status, and the text of the current query. From this display you can select a query to `EXPLAIN`, kill a query, and a few other tasks. A header at the top of the display gives information about the server, such as version, uptime, and some statistics like the number of queries per second. The program also has some other functions, but I never found myself using them much.

There are `mytop` packages for various GNU/Linux distributions, such as Gentoo and Fedora Core, or you can install one from Jeremy's website. It is very small and has minimal dependencies. On the downside, it hasn't been maintained actively for a while and doesn't work correctly with MySQL 5.x.

A similar tool is `mtop`. It has a tabular process display much like `mytop`, and although it lacks some features and adds others, the two programs are very similar. It is also a Perl script and there are installation packages for some operating systems, or you can download it from SourceForge. Unfortunately, it is not actively maintained and does not work correctly on newer versions of MySQL.

Some programmers have also created scripts to output MySQL's process list for easy consumption by other scripts. An example is this `SHOW FULL PROCESSLIST` script, available from the always-useful MySQL Forge.

My own contribution is `innotop`, a MySQL and InnoDB monitor. As MySQL has become increasingly popular, InnoDB has become the most widely used transactional MySQL storage engine. InnoDB has many differences from other MySQL storage engines, so it requires different monitoring methods. It exposes internal status by dumping a potentially huge amount of semi-formatted text in response to the `SHOW INNODB STATUS` command. There's a lot of raw data in this text, but it's unusable for real-time monitoring, so I wrote `innotop` to format and display it conveniently. It is the main monitoring tool at my current employer.

`Innotop` is much more capable than the other tools I've mentioned, and can replace them completely. It has a list of processes and status information, and offers the standard functions to kill and explain queries. It also offers many features that are not in any other tool, including being able to list current transactions, lock waits, deadlock information, foreign key errors, I/O and log statistics, InnoDB row operation and semaphore statistics, and information on the InnoDB buffer pool, memory usage, insert buffer, and adaptive hash index. It also displays more standard MySQL information than `mytop` and its clones, such as compact, tabular displays of current and past status information snapshots. It is very configurable and has interactive help.

Installation is simple, because `innotop` is a ready-to-run Perl script, but there are no installation packages yet, so you must download it from my website.

There are also some web-based tools. There are two web-based `mytop` clones, `phpMyTop` and `ajaxMyTop`. These are useful when you don't have shell access and can't connect remotely to your database server, but can connect from a web server. `ajaxMyTop` is more recent and seems to be more actively developed. It also feels more like a traditional GUI program, because thanks to Ajax, the entire page does not constantly refresh itself.

Another web-based tool is the popular `phpMyAdmin` package. `phpMyAdmin` is a Swiss Army Knife, with features to design tables, run queries, manage users and more. Its focus isn't on monitoring queries and processes, but it has some of the features I've mentioned earlier, such as showing a process list.

Finally, if you need to monitor what's happening inside a MySQL server and don't care to--or can't--use a third-party tool, MySQL's own `mysqladmin` command-line program works. For example, to watch incremental changes to the query cache, run the command:

```
$ mysqladmin extended -r -i 10 | grep Qcache
```

Of course, `innotop` can do that for you too, only better. Take a look at its "V" mode. Still, this can be handy when you don't have any way to run `innotop`.

## Tools to Monitor a MySQL Server

Sometimes, rather than monitoring the queries running in a MySQL server, you need to analyze other aspects of the system's performance. You could use standard command-line utilities to monitor the resources used by the MySQL process on GNU/Linux, or you could run Giuseppe Maxia's helpful script to measure MySQL resource consumption. This tool recursively examines the processes associated with the MySQL server's process ID, and prints a report on what it finds. For more information, read Giuseppe's own article on the O'Reilly Databases blog.

The MySQL Forge website is an excellent place to discover tips, tricks, scripts, and code snippets for daily MySQL administration and programming tasks. For example, there's an entry to help you measure replication speed, a "poor man's query profiler" to capture queries as they fly by on the network interface, and much more.

Another excellent resource is `mysqlreport`, a well-designed program that turns MySQL status information into knowledge. It prints out a report of relevant variables, sensibly arranged for an experienced MySQL user. I find this tool indispensable when I have to troubleshoot a server without knowing anything about it in advance. For example, if someone asks me to help reduce load on a MySQL server that's running at 100 percent CPU, the first thing I do is to run `mysqlreport`. I can get more information by glancing at its output than I could in 10 minutes of talking to the customer. It immediately tells me where to focus my efforts. If I see a high key read ratio and a high percentage of index scans, I can immediately look for large indexes and a key buffer that's too small. That intuition could take many minutes to develop just by examining `SHOW STATUS`.

The `mysqlreport` website has full information on how to install and use the program, but better yet, there are excellent tutorials on how to interpret its output, with real examples. Some of these go into detail on MySQL internals, and I recommend them to any MySQL developer.

Another common task is setting up automated systems to monitor your server and let you know if it's alive. You could write your own monitor, or you could just plug in a ready-made one. According to a MySQL poll, Nagios is the most popular tool for doing this. There's also a Watchdog mysql monitor plugin for mon, the Linux scheduling and alert management tool. We currently use a home-grown system at my employer, but we're looking at using Nagios soon.

**DISCLAIMER:** The content provided in this article is not warranted or guaranteed by DBA Guys. The content provided is intended for entertainment and/or educational purposes in order to introduce to the reader key ideas, concepts, and/or product reviews. As such it is incumbent upon the reader to employ real-world tactics for security and implementation of best practices. We are not liable for any negative consequences that may result from implementing any information covered in our articles or tutorials. If this is a hardware review, it is not recommended to open and/or modify your hardware.