



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

## MariaDB MaxScale

## Debug & Diagnostic Support

Mark Riddoch

Last Updated: 24<sup>th</sup> November 2014



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

## Contents

Change History .....	4
Introduction .....	5
Debugger Support.....	5
Command Line Option.....	5
Convenience Functions .....	5
Printing Services .....	5
Printing Sessions .....	6
Printing Servers .....	7
Modules .....	7
Descriptor Control Blocks.....	8
Diagnostic Interface .....	9
Listing Services .....	12
Listing Listeners .....	12
Listing Servers.....	13
Listing Modules .....	13
Showing Services.....	13
Showing Sessions .....	14
Show Servers .....	15
- Version string is available in the output only if the node is running. ....	16
There may be other status description such us:.....	17
Show Server.....	18
Show DCBS .....	18
Show Modules.....	20
Show Polling Statistics .....	21
Show Dbusers .....	21
Show Users .....	22
Show Monitors.....	23



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

Shutdown maxscale .....	24
Shutdown monitor .....	24
Shutdown service .....	25
Restart service .....	26
Restart Monitor .....	26
Set server .....	27
Clear server .....	28
Reload users .....	28
Reload config .....	30
Add user .....	30
Enable/disable log .....	31
Logging facility .....	31
Log contents .....	31
Error log .....	32
Message log .....	32
Trace log .....	33
Managing logs .....	35
Rotating logs .....	35



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

## Change History

Date	Comment
20th June 2013	Initial Version
22nd July 2013	Updated with new naming MaxScale Addition of description of login process for the debug CLI Updates debug CLI output examples Addition of show users, shutdown maxscale, shutdown service, restart service, set server, clear server, reload users, reload config and add user commands.
23rd July 2013	Rename of show users command to show dbusers and addition of the show users command to show the admin users. Addition of example configuration data.
14th November 2013	Added enable/disable log commands details Added Galera Monitor as an example in show monitors
3rd March 2014	Added show users details for MySQL users
27th May 2014	Document the new debugcli mode switch and command changes in the two modes. Added the new show server command.
29th May 2014	Addition of new list command that gives concise tabular output
4th June 2014	Added new 'show monitors' and 'show servers' details
29th August 2014	Added new 'show monitors' details for MySQL/Galera monitors



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

## Introduction

MaxScale is a complex application and as such is bound to have bugs and support issues that occur from time to time. There are a number of things we need to consider for the development stages and long term supportability of MaxScale

- Flexible logging of MaxScale activity
- Support for connecting a debugger to MaxScale
- A diagnostic interface to MaxScale

The topic of logging has already been discussed in another document in this series of documents about MaxScale and will not be covered further here.

## Debugger Support

Beyond the language support for debugging using tools such as gdb, MaxScale will also offer convenience functions for the debugger to call and a command line argument that is useful to run MaxScale under the debugger.

### Command Line Option

Normally when MaxScale starts it will place itself in the background and setup the signal masks so that it is immune to the normal set of signals that will cause the process to exit, SIGINT and SIGQUIT. This behaviour is normally what is required, however if you wish to run MaxScale under the control of a debugger it is useful to suppress this behaviour. A command line option, `-d` is provided to turn off this behaviour.

```
% gdb maxscale
(gdb) run -d
```

### Convenience Functions

A set of convenience functions is provided that may be used within the debugger session to extract information from MaxScale.

### Printing Services

A service within MaxScale provides the encapsulation of the port MaxScale listen on, the protocol it uses, the set of servers it may route to and the routing method to use. Two functions exists that allow you to display the details of the services and may be executed from within a debugger session.

The `printAllServices()` function will print all the defined services within MaxScale and is invoked using the call syntax of the debugger.

```
(gdb) call printAllServices()
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
Service 0x60da20
  Service:      Debug Service
  Router:       debugcli (0x7ffff5a7c2a0)
  Started:      Thu Jun 20 15:13:32 2013
  Backend databases
  Total connections: 1
  Currently connected: 1
Service 0x60d010
  Service:      Test Service
  Router:       readconnroute (0x7ffff5c7e260)
  Started:      Thu Jun 20 15:13:32 2013
  Backend databases
    127.0.0.1:3308 Protocol: MySQLBackend
    127.0.0.1:3307 Protocol: MySQLBackend
    127.0.0.1:3306 Protocol: MySQLBackend
  Total connections: 1
  Currently connected: 1
```

(gdb)

It is possible to print an individual service if you know the memory address of the service.

```
(gdb) call printService(0x60da20)
```

```
Service 0x60da20
  Service:      Debug Service
  Router:       debugcli (0x7ffff5a7c2a0)
  Started:      Thu Jun 20 15:13:32 2013
  Backend databases
  Total connections: 1
  Currently connected: 1
```

(gdb)

## Printing Sessions

Sessions represent the data for a client that is connecting through MaxScale, there will be a session for each client and one for each listener for a specific port/protocol combination.

Similarly there are two calls to print all or a particular session.

```
(gdb) call printAllSessions()
```

```
Session 0x60fdf0
  Service:      Debug Service (0x60da20)
  Client DCB:   0x60f6c0
  Connected:    Thu Jun 20 15:13:32 2013
Session 0x60f620
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
Service:  Test Service (0x60d010)
Client DCB:  0x60ead0
Connected:  Thu Jun 20 15:13:32 2013
(gdb) call printSession(0x60fdf0)
Session 0x60fdf0
Service:  Debug Service (0x60da20)
Client DCB:  0x60f6c0
Connected:  Thu Jun 20 15:13:32 2013
(gdb)
```

## Printing Servers

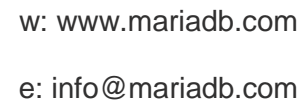
Servers are a representation of the backend database to which MaxScale may route SQL statements. Similarly two calls exist to print server details.

```
(gdb) call printAllServers()
Server 0x60d9a0
Server: 127.0.0.1
Protocol: MySQLBackend
Port: 3308
Server 0x60d920
Server: 127.0.0.1
Protocol: MySQLBackend
Port: 3307
Server 0x60d8a0
Server: 127.0.0.1
Protocol: MySQLBackend
Port: 3306
(gdb) call printServer(0x60d920)
Server 0x60d920
Server: 127.0.0.1
Protocol: MySQLBackend
Port: 3307
(gdb)
```

## Modules

MaxScale makes significant use of modules, shared objects, that are loaded on demand based on the configuration. A routine exists that will print the currently loaded modules.

```
(gdb) call printModules()
Module Name      | Module Type | Version
-----
```





w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
DCB: 0x60f6c0
  DCB state:          DCB for listening socket
  Queued write data:  0
  Statistics:
    No. of Reads:    0
    No. of Writes:   0
    No. of Buffered Writes: 0
    No. of Accepts: 0
(gdb) call printDCB(0x60ead0)
DCB: 0x60ead0
  DCB state:          DCB for listening socket
  Queued write data:  0
  Statistics:
    No. of Reads:    0
    No. of Writes:   0
    No. of Buffered Writes: 0
    No. of Accepts: 0
(gdb)
```

## Diagnostic Interface

It is possible to configure a service to run within MaxScale that will allow a user to telnet to a port on the machine and be connected to MaxScale. This is configured by creating a service that uses the debugcli routing module and the telnetd protocol with an associated listener. The service does not require any backend databases to be configured since the router never forwards any data, it merely accepts commands and executes them, returning data to the user.

The example below shows the configuration that is required to set-up a debug interface that listens for incoming telnet connections on port 4442.

```
[Debug Service]
type=service
router=debugcli

[Debug Listener]
type=listener
service=Debug Service
protocol=telnetd
port=4442
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

The `Debug Service` section sets up a service with no backend database servers, but with a `debugcli` module as the router. This module will implement the commands and send the data back to the client.

The `debugcli` accepts router options of either `developer` or `user`, these are used to control the mode of the user interface. If no router options are given then the CLI is in user mode by default.

The `Debug Listener` section setups the protocol and port combination and links that to the service.

Assuming a configuration that includes the debug service, with the listening port set to 4442, to connect from the machine that runs MaxScale you must first install telnet and then simply call `telnet` to connect.

```
-bash-4.1$ telnet localhost 4442
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Welcome the SkySQL MaxScale Debug Interface (V1.1.0).
Type help for a list of available commands.

MaxScale login: admin
Password:

MaxScale>
```

As delivered MaxScale uses a default login name of `admin` with the password of `skysql` for connections to the debug interface. Users may be added to the CLI by use of the `add user` command.

This places you in the debug command line interface of MaxScale, there is a help system that will display the commands available to you

```
MaxScale> help
Available commands:
  add user
  clear server
  disable log
  enable log
  list [listeners|modules|services|servers|sessions]
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
reload [config|dbusers]
remove user
restart [monitor|service]
set server
show
[dcbs|dcb|dbusers|epoll|modules|monitors|server|servers|services|service|session|sessions|users]
shutdown [maxscale|monitor|service]
```

Type help command to see details of each command.

Where commands require names as arguments and these names contain whitespace either the \ character may be used to escape the whitespace or the name may be enclosed in double quotes ".

**MaxScale>**

Different command help is shown in user mode and developer mode, in user mode the help for the show command is;

**MaxScale>** help show

Available options to the show command:

dcbs	Show all descriptor control blocks (network connections)
dcb	Show a single descriptor control block e.g. show dcb 0x493340
dbusers	Show statistics and user names for a service's user table. Example : show dbusers <service name>
epoll	Show the poll statistics
modules	Show all currently loaded modules
monitors	Show the monitors that are configured
server	Show details for a named server, e.g. show server dbnode1
servers	Show all configured servers
services	Show all configured services in MaxScale
service	Show a single service in MaxScale, may be passed a service name
session	Show a single session in MaxScale, e.g. show session 0x284830
sessions	Show all active sessions in MaxScale
users	Show statistics and user names for the debug interface

**MaxScale>**

However in developer mode the help is;

**MaxScale>** help show

Available options to the show command:

dcbs	Show all descriptor control blocks (network connections)
dcb	Show a single descriptor control block e.g. show dcb 0x493340
dbusers	Show statistics and user names for a service's user table
epoll	Show the poll statistics



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
modules      Show all currently loaded modules
monitors     Show the monitors that are configured
server       Show details for a server, e.g. show server 0x485390
servers      Show all configured servers
services     Show all configured services in MaxScale
session      Show a single session in MaxScale, e.g. show session 0x284830
sessions     Show all active sessions in MaxScale
users        Show statistics and user names for the debug interface
```

**MaxScale>**

The commands available are very similar to those described above to print things from the debugger, the advantage being that you do not need a debug version or a debugger to use them.

## Listing Services

The `list services` command is designed to give a concise tabular view of the currently configured services within MaxScale along with key data that summarises the use beign made of the service.

```
MaxScale> list services
Service Name      | Router Module      | #Users | Total Sessions
-----
Test Service      | readconnroute      | 1      | 1
Split Service     | readwritesplit     | 1      | 1
Debug Service     | debugcli           | 2      | 2
MaxScale>
```

This provides a useful mechanism to see what is configured and provide the service names that can be passed to a `show service` command.

## Listing Listeners

The `list listeners` command outputs a table that provides the current set of listeners within the MaxScale instance and shows the status of each listener.

```
MaxScale> list listeners
Service Name      | Protocol Module    | Address      | Port  | State
-----
Test Service      | MySQLClient        | (null)       | 4006  | Running
Split Service     | MySQLClient        | (null)       | 4007  | Running
Debug Service     | telnetd            | localhost    | 4242  | Running
MaxScale>
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

## Listing Servers

The `list servers` command will display a table that contains a row for every server defined in the configuration file. The row contains the server name that can be passed to the `show server` command, the address and port of the server, its current status and the number of connections to that server from MaxScale.

```
MaxScale> list servers
```

Server	Address	Port	Status	Connections
server1	127.0.0.1	3306	Running	0
server2	127.0.0.1	3307	Slave, Running	0
server3	127.0.0.1	3308	Master, Running	0
server4	127.0.0.1	3309	Slave, Running	0

```
MaxScale>
```

## Listing Modules

The `list modules` command displays a table of all the modules loaded within MaxScale.

```
MaxScale> list modules
```

Module Name	Module Type	Version
telnetd	Protocol	V1.0.1
MySQLClient	Protocol	V1.0.0
mysqlmon	Monitor	V1.1.0
readconnroute	Router	V1.0.2
readwritesplit	Router	V1.0.2
debugcli	Router	V1.1.1

```
MaxScale>
```

## Showing Services

The `show services` command will show all the services configured currently

```
MaxScale> show services
```

Service 0xf44c10

```
Service:          Test Service
Router:           readconnroute (0x7f7fd8afba40)
Number of router sessions:      0
Current no. of router sessions: 0
Number of queries forwarded:    0
Started:           Mon Jul 22 11:24:09 2013
Backend databases
127.0.0.1:3309  Protocol: MySQLBackend
127.0.0.1:3308  Protocol: MySQLBackend
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
127.0.0.1:3307 Protocol: MySQLBackend
127.0.0.1:3306 Protocol: MySQLBackend
Users data:      0xf454b0
Total connections: 1
Currently connected: 1
Service 0xf43910
  Service:      Split Service
  Router:      readwritesplit (0x7f7fd8f05460)
  Number of router sessions: 0
  Current no. of router sessions: 0
  Number of queries forwarded: 0
  Number of queries forwarded to master: 0
  Number of queries forwarded to slave: 0
  Number of queries forwarded to all: 0
  Started:      Mon Jul 22 11:24:09 2013
  Backend databases
    127.0.0.1:3308 Protocol: MySQLBackend
    127.0.0.1:3307 Protocol: MySQLBackend
    127.0.0.1:3306 Protocol: MySQLBackend
  Users data:      0xf449b0
  Total connections: 1
  Currently connected: 1
Service 0xea0190
  Service:      Debug Service
  Router:      debugcli (0x7f7fd910d620)
  Started:      Mon Jul 22 11:24:09 2013
  Backend databases
  Users data:      0xea2d80
  Total connections: 2
  Currently connected: 2
MaxScale>
```

## Showing Sessions

There are two options to show sessions, either an individual session or all sessions

```
MaxScale> show sessions
Session 0x6f8f20
  State:      Session Ready
  Service:      Debug Service (0x649190)
  Client DCB:  0x6f8e20
  Client Address: 0.0.0.0
  Connected:    Mon Jul 22 11:31:56 2013
Session 0x6f83b0
  State:      Session Allocated
  Service:      Split Service (0x6ec910)
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
Client DCB:      0x64b430
Client Address:  127.0.0.1
Connected:      Mon Jul 22 11:31:28 2013
Session 0x6efba0
  State:         Listener Session
  Service:       Debug Service (0x649190)
  Client DCB:    0x64b180
  Connected:     Mon Jul 22 11:31:21 2013
Session 0x64b530
  State:         Listener Session
  Service:       Split Service (0x6ec910)
  Client DCB:    0x6ef8e0
  Connected:     Mon Jul 22 11:31:21 2013
Session 0x618840
  State:         Listener Session
  Service:       Test Service (0x6edc10)
  Client DCB:    0x6ef320
  Connected:     Mon Jul 22 11:31:21 2013
MaxScale> show session 0x6f83b0
Session 0x6f83b0
  State:         Session Allocated
  Service:       Split Service (0x6ec910)
  Client DCB:    0x64b430
  Client Address: 127.0.0.1
  Connected:     Mon Jul 22 11:31:28 2013
MaxScale>
```

## Show Servers

The configured backend databases can be displayed using the `show servers` command.

```
MaxScale> show servers
Server 0x6ec840 (server1)
  Server:        127.0.0.1
  Status:        Running
  Protocol:      MySQLBackend
  Port:          3306
  Number of connections: 0
  Current no. of connections:0

Server 0x6ec770 (server2)
  Server:        127.0.0.1
  Status:        Master, Running
  Protocol:      MySQLBackend
  Port:          3307
  Server Version: 5.5.35-MariaDB-log
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
Node Id: 1
Master Id: -1
Slave Ids: 2,3
Repl Depth: 0
Last Repl Heartbeat: 1417080906
Number of connections: 1
Current no. of connections:1
Server 0x6ec6a0 (server3)
  Server: 127.0.0.1
  Status: Slave, Running
  Protocol: MySQLBackend
  Port: 3308
  Server Version: 5.5.35-MariaDB-log
  Node Id: 2
  Master Id: 1
  Slave Ids:
  Repl Depth: 1
  Slave delay: 8
  Last Repl Heartbeat: 1417080898
  Number of connections: 1
  Current no. of connections:1
Server 0x6ec5d0 (server4)
  Server: 127.0.0.1
  Status: Down
  Protocol: MySQLBackend
  Port: 3309
  Server Version: 5.5.35-MariaDB-log
  Node Id: 3
  Master Id: 1
  Slave Ids:
  Repl Depth: 1
  Number of connections: 0
  Current no. of connections:0
```

**MaxScale>**

- Version string is available in the output only if the node is running.
- Node Id possible values:
  - the value of server-id from MySQL or MariaDB servers in Master/Slave replication setup.
  - the value of 'wsrep\_local\_index' for Galera cluster nodes
  - the -1 value for a failure getting one of these informations
- Repl Depth is the replication depth level found by MaxScale MySQL Monitor



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

- Master Id is the master node, if available, for current server
- Slave Ids is the slave list for the current node, if available

Note, the Master Root Server used for routing decision is the server with master role and with lowest replication depth level. All slaves servers even if they are intermediate master are suitable for read statement routing decisions.

- Slave Delay is the found time difference used for Replication Lag support in routing decision, between the value read by the Slave (it was inserted into the master) and maxscale current time  
Value of 0 or less than monitor interval means there is no replication delay.
- Last Repl Heartbeat is the maxScale timestamp read or inserted (if current server is master)

The Replication Heartbeat table is updated by MySQL replication, starting MaxScale when there is a significant slave delay may result that Slave Delay and Last Repl Heartbeat are not available for some time in the slave server details

There may be other status description such us:

Slave of External Server:	the node is slave of a server not configured in MaxScale (by MySQL monitor)
Master Stickiness:	the Master node is kept (by Galera monitor)
Stale Status:	the master node is kept even if the replication is not working (by MySQL monitor)
Auth Error:	Monitor was not able to login into backend

A few examples:

```
Server 0x1a5aae0 (server3)
  Server: 127.0.0.1
  Status: Master, Slave of External Server, Running
  Protocol: MySQLBackend
  Port: 3308
  Server Version: 5.5.35-MariaDB-log
  Node Id: 3
  Master Id: 1
  Slave Ids: 2
  Repl Depth: 1
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
Server 0x2d1b5c0 (server2)
  Server:          192.168.122.142
  Status:          Master, Synced, Master Stickiness, Running
  Protocol:        MySQLBackend
  Port:            3306
  Server Version:  5.5.40-MariaDB-wsrep-log
  Node Id:         1
  Repl Depth:      0
```

## Show Server

Details of an individual server can be displayed by using the show server command. In user mode the show server command is passed the name of the server to display, these names are the section names used in the configuration file.

```
MaxScale> show server server4
Server 0x6ec5d0 (server4)
  Server:          127.0.0.1
  Status:          Down
  Protocol:        MySQLBackend
  Port:            3309
  Number of connections: 0
  Current no. of connections:0
MaxScale>
```

In developer mode the show server command is passed the address of a server structure.

```
MaxScale> show server 0x6ec5d0
Server 0x6ec5d0 (server4)
  Server:          127.0.0.1
  Status:          Down
  Protocol:        MySQLBackend
  Port:            3309
  Number of connections: 0
  Current no. of connections:0
MaxScale>
```

## Show DCBS

There are two forms of the show command that will give you DCB information, the first will display information for all DCBs within the system.

```
MaxScale> show dcbs
DCB: 0x6ef320
  DCB state:      DCB for listening socket
  Service:        Test Service
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
Queued write data: 0
Statistics:
    No. of Reads:      0
    No. of Writes:     0
    No. of Buffered Writes: 0
    No. of Accepts:    0
DCB: 0x6ef8e0
    DCB state:          DCB for listening socket
    Service:            Split Service
    Queued write data: 0
    Statistics:
        No. of Reads:      0
        No. of Writes:     0
        No. of Buffered Writes: 0
        No. of Accepts:    1
DCB: 0x64b180
    DCB state:          DCB for listening socket
    Service:            Debug Service
    Queued write data: 0
    Statistics:
        No. of Reads:      0
        No. of Writes:     0
        No. of Buffered Writes: 0
        No. of Accepts:    1
DCB: 0x64b430
    DCB state:          DCB processing event
    Service:            Split Service
    Connected to:       127.0.0.1
    Queued write data: 0
    Statistics:
        No. of Reads:      2
        No. of Writes:     3
        No. of Buffered Writes: 0
        No. of Accepts:    0
DCB: 0x6f8400
    DCB state:          DCB in the polling loop
    Service:            Split Service
    Queued write data: 0
    Statistics:
        No. of Reads:      3
        No. of Writes:     1
        No. of Buffered Writes: 0
        No. of Accepts:    0
DCB: 0x6f8b40
    DCB state:          DCB in the polling loop
    Service:            Split Service
    Queued write data: 0
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
Statistics:
  No. of Reads:      2
  No. of Writes:     0
  No. of Buffered Writes: 0
  No. of Accepts:    0
DCB: 0x6f8e20
  DCB state:         DCB processing event
  Service:           Debug Service
  Connected to:      0.0.0.0
  Queued write data: 0
  Statistics:
    No. of Reads:      8
    No. of Writes:     133
    No. of Buffered Writes: 0
    No. of Accepts:    0
MaxScale>
```

An individual DCB can be displayed by passing the DCB address to the `show dcb` command

```
MaxScale> show dcb 0x64b430
DCB: 0x64b430
  DCB state:         DCB processing event
  Connected to:      127.0.0.1
  Owning Session:    7308208
  Queued write data: 0
  Statistics:
    No. of Reads:      2
    No. of Writes:     3
    No. of Buffered Writes: 0
    No. of Accepts:    0
MaxScale>
```

## Show Modules

The `show modules` command will display the list of the currently loaded modules

```
MaxScale> show modules
Module Name      | Module Type | Version
-----
MySQLBackend     | Protocol   | V2.0.0
telnetd          | Protocol   | V1.0.1
MySQLClient      | Protocol   | V1.0.0
mysqlmon         | Monitor    | V1.0.0
readconroute     | Router     | V1.0.2
readwritesplit   | Router     | V1.0.2
debugcli         | Router     | V1.1.0
MaxScale>
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

## Show Polling Statistics

Display statistics related to the main polling loop. The `epoll` cycles is the count of the number of times `epoll` has returned with one or more event. The other counters are for each individual events that has been detected.

```
MaxScale> show epoll
Number of epoll cycles:      7928
Number of read events:      2000920
Number of write events:     2000927
Number of error events:      0
Number of hangup events:     0
Number of accept events:     4
MaxScale>
```

## Show Dbusers

The `show dbuser` command allows data regarding the table that holds the database users for a service to be displayed. It does not give the actual user data, but rather details of the hashtable distribution.

The `show dbuser` command takes different arguments in the two modes of MaxScale, in user mode it may be passed the name of a service rather than an address, whilst in developer mode it needs the address of a user structure that has been extracted from a service.

In developer mode the `show users` commands must be passed the address of the user table, this can be extracted from the output of a `show services` command.

```
MaxScale> show services
Service 0x6ec910
  Service:          Split Service
  Router:           readwritesplit (0x7ffff1698460)
  Number of router sessions:      1
  Current no. of router sessions: 0
  Number of queries forwarded:    2
  Number of queries forwarded to master: 0
  Number of queries forwarded to slave: 1
  Number of queries forwarded to all: 1
  Started:              Mon Jul 22 11:31:21 2013
  Backend databases
    127.0.0.1:3308 Protocol: MySQLBackend
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
127.0.0.1:3307 Protocol: MySQLBackend
127.0.0.1:3306 Protocol: MySQLBackend
Users data:      0x6ed9b0
Total connections: 2
Currently connected: 1
```

...

The following example shows the MySQL users.

Users are loaded with the host (IPv4 data) as they are created in the backend.

```
MaxScale> show dbusers 0x6ed9b0
Users table data
Hashtable: 0x19243a0, size 52
      No. of entries:      16
      Average chain length: 0.3
      Longest chain length: 4
User names: one@%, new@192.168.56.1, new@127.0.0.1, repluser@%,
seven@127.0.0.1, four@%
MaxScale>
```

In user mode the command is simply passed the name of the service

```
MaxScale> show dbusers "Split Service"
Users table data
Hashtable: 0x19243a0, size 52
      No. of entries:      16
      Average chain length: 0.3
      Longest chain length: 4
User names: one@%, new@192.168.56.1, new@127.0.0.1, repluser@%,
seven@127.0.0.1, four@%
MaxScale>
```

Please note the use of quotes in the name in order to escape the white space character.

## Show Users

The `show users` command lists the users defined for the administration interface. Note that if there are no users defined, and the default admin user is in use, then no users will be displayed.

```
MaxScale> show users
Administration interface users:
Users table data
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
Hashtable: 0x25ef5e0, size 52
  No. of entries:      2
  Average chain length: 0.0
  Longest chain length: 1
User names: admin, mark
MaxScale>
```

## Show Monitors

The `show monitors` show the status of the database monitors. The address of the monitor can be used for the `shutdown monitor` and `restart monitor` commands.

```
MaxScale> show monitors
Monitor: 0x80a510
  Name:      MySQL Monitor
  Monitor running
  Sampling interval: 10000 milliseconds
  Replication lag:   enabled
  Detect Stale Master: disabled
  Connect Timeout:   3 seconds
  Read Timeout:      1 seconds
  Write Timeout:     2 seconds
  Monitored servers: 127.0.0.1:3306, 127.0.0.1:3307, 127.0.0.1:3308,
127.0.0.1:3309
```

```
Monitor: 0x73d3d0
  Name:      Galera Monitor
  Monitor running
  Sampling interval: 7000 milliseconds
  Master Failback:  off
  Connect Timeout:  3 seconds
  Read Timeout:     1 seconds
  Write Timeout:    2 seconds
  Monitored servers: 127.0.0.1:3310, 127.0.0.1:3311, 127.0.0.1:3312
MaxScale>
```

Monitor timeouts used in Galera and MySQL monitor follow the rules of `mysql_real_connect` C API:

- Connect Timeout is the connect timeout in seconds.



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

- Read Timeout is the timeout in seconds for each attempt to read from the server. There are retries if necessary, so the total effective timeout value is three times the option value.
- Write Timeout is the timeout in seconds for each attempt to write to the server. There is a retry if necessary, so the total effective timeout value is two times the option value.

## Shutdown maxscale

The CLI can be used to shutdown the MaxScale server by use of the `shutdown` command, it may be called with the argument either `maxscale` or `gateway`.

```
MaxScale> shutdown maxscale
```

## Shutdown monitor

The `shutdown monitor` command stops the thread that is used to run the monitor and will stop any update of the server status flags. This is useful prior to manual setting of the states of the server using the `set server` and `clear server` commands.

```
MaxScale> show monitors
Monitor: 0x80a510
  Name:          MySQL Monitor
  Monitor running
  Sampling interval: 10000 milliseconds
  Monitored servers: 127.0.0.1:3306, 127.0.0.1:3307, 127.0.0.1:3308,
127.0.0.1:3309
MaxScale> shutdown monitor 0x80a510
```

```
MaxScale> show monitors
Monitor: 0x80a510
  Name:          MySQL Monitor
  Monitor stopped
  Sampling interval: 10000 milliseconds
  Monitored servers: 127.0.0.1:3306, 127.0.0.1:3307, 127.0.0.1:3308,
127.0.0.1:3309
MaxScale>
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

It may take some time before a monitor actually stops following the issuing of a `shutdown monitor` command. Stopped monitors can be restarted by issuing a `restart monitor` command.

## Shutdown service

The `shutdown service` command can be used to stop the listener for a particular service. This will prevent any new clients from using the service but will not terminate any clients already connected to the service.

The `shutdown service` command needs the address of a service to be passed as an argument, this can be obtained by running `show services`.

```
MaxScale> show services
Service 0x6edc10
  Service:          Test Service
  Router:           readconnroute (0x7ffff128ea40)
  Number of router sessions:      257
  Current no. of router sessions: 0
  Number of queries forwarded:    1000193
  Started:             Mon Jul 22 11:31:21 2013
  Backend databases
    127.0.0.1:3309 Protocol: MySQLBackend
    127.0.0.1:3308 Protocol: MySQLBackend
    127.0.0.1:3307 Protocol: MySQLBackend
    127.0.0.1:3306 Protocol: MySQLBackend
  Users data:        0x6ee4b0
  Total connections: 258
  Currently connected: 1
Service 0x6ec910
  Service:          Split Service
  Router:           readwritesplit (0x7ffff1698460)
  Number of router sessions:      1
  Current no. of router sessions: 0
  Number of queries forwarded:    2
  Number of queries forwarded to master: 0
  Number of queries forwarded to slave: 1
  Number of queries forwarded to all: 1
  Started:             Mon Jul 22 11:31:21 2013
  Backend databases
    127.0.0.1:3308 Protocol: MySQLBackend
    127.0.0.1:3307 Protocol: MySQLBackend
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
127.0.0.1:3306 Protocol: MySQLBackend
Users data:      0x6ed9b0
Total connections: 2
Currently connected: 1
Service 0x649190
  Service:      Debug Service
  Router:      debugcli (0x7ffff18a0620)
  Started:      Mon Jul 22 11:31:21 2013
  Backend databases
  Users data:      0x64bd80
  Total connections: 2
  Currently connected: 2
MaxScale> shutdown service 0x6edc10
```

In user mode the `shutdown service` command may be passed the name of the service as defined in configuration file.

```
MaxScale> shutdown service Split\ Service
```

## Restart service

The `restart service` command can be used to restart a previously stopped listener for a service. In developer mode the address of the service must be passed.

```
MaxScale> restart service 0x6edc10
MaxScale>
```

In user mode the name of the service may be passed.

```
MaxScale> restart service Test\ Service
MaxScale>
```

As with `shutdown service` the address of the service should be passed as an argument.

## Restart Monitor

The `restart monitor` command will restart a previously stopped monitor.

```
MaxScale> show monitors
Monitor: 0x80a510
  Name:      MySQL Monitor
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
Monitor stopped
Monitored servers: 127.0.0.1:3306, 127.0.0.1:3307, 127.0.0.1:3308,
127.0.0.1:3309
MaxScale> restart monitor 0x80a510
MaxScale>
```

## Set server

The `set server` command can be used to set the status flags of a server directly from the user interface. The command should be passed a server address that has been obtained from the output of a `show servers` command.

```
MaxScale> show servers
Server 0x6ec840 (server1)
  Server:      127.0.0.1
  Status:      Running
  Protocol:    MySQLBackend
  Port:        3306
  Server Version: 10.0.11-MariaDB-log
  Node Id:     29
  Number of connections: 0
  Current n. of conns: 0
Server 0x6ec770 (server2)
  Server:      127.0.0.1
  Status:      Master, Running
  Protocol:    MySQLBackend
  Port:        3307
  Server Version: 5.5.35-MariaDB-log
  Node Id:     1
  Number of connections: 1
  Current n. of conns: 0
Server 0x6ec6a0 (server3)
  Server:      127.0.0.1
  Status:      Slave, Running
  Protocol:    MySQLBackend
  Port:        3308
  Server Version: 5.5.35-MariaDB-log
  Node Id:     2
  Number of connections: 258
  Current n. of conns: 0
Server 0x6ec5d0 (server4)
  Server:      127.0.0.1
  Status:      Down
  Protocol:    MySQLBackend
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
Port: 3309
Node Id: -1
Number of connections: 0
Current n. of conns: 0
MaxScale> set server 0x6ec840 slave
```

Valid options that are recognised by the `set server` command are `running`, `master` and `slave`. Please note that if the monitor is running it will reset the flags to match reality, this interface is really for use when the monitor is disabled.

In user mode there is no need to find the address of the server structure, the name of the server from the section header in the configuration file may be given.

```
MaxScale> set server server1 slave
```

## Clear server

The `clear server` command is the complement to the `set server` command, it allows status bits related to a server to be cleared.

```
MaxScale> clear server 0x6ec840 slave
```

Likewise in user mode the server name may be given.

```
MaxScale> clear server server1 slave
```

## Reload users

The `reload users` command is used to force a service to go back and reload the table of database users from the backend database. This is the data used in the transparent authentication mechanism in the MySQL protocol. The command should be passed the address of the service as shown in the output of the `show services` command.

```
MaxScale> show services
Service 0x6edc10
  Service: Test Service
  Router: readconroute (0x7ffff128ea40)
  Number of router sessions: 257
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
Current no. of router sessions: 0
Number of queries forwarded: 1000193
Started: Mon Jul 22 11:31:21 2013
Backend databases
    127.0.0.1:3309 Protocol: MySQLBackend
    127.0.0.1:3308 Protocol: MySQLBackend
    127.0.0.1:3307 Protocol: MySQLBackend
    127.0.0.1:3306 Protocol: MySQLBackend
Users data: 0x6ee4b0
Total connections: 258
Currently connected: 1
Service 0x6ec910
    Service: Split Service
    Router: readwritesplit (0x7ffff1698460)
    Number of router sessions: 1
    Current no. of router sessions: 0
    Number of queries forwarded: 2
    Number of queries forwarded to master: 0
    Number of queries forwarded to slave: 1
    Number of queries forwarded to all: 1
    Started: Mon Jul 22 11:31:21 2013
    Backend databases
        127.0.0.1:3308 Protocol: MySQLBackend
        127.0.0.1:3307 Protocol: MySQLBackend
        127.0.0.1:3306 Protocol: MySQLBackend
    Users data: 0x6ed9b0
    Total connections: 2
    Currently connected: 1
Service 0x649190
    Service: Debug Service
    Router: debugcli (0x7ffff18a0620)
    Started: Mon Jul 22 11:31:21 2013
    Backend databases
    Users data: 0x64bd80
    Total connections: 2
    Currently connected: 2
MaxScale> reload users 0x6edc10
Loaded 34 users.
MaxScale>
```

If user mode is in use then the service name may be given.

```
MaxScale> reload users "Test Service"
Loaded 34 users.
MaxScale>
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

## Reload config

The `reload config` command can be used to force MaxScale to re-read the `MaxScale.cnf` and update itself to the latest configuration defined in that configuration file. It is also possible to force the reading of the configuration file by sending a HangUp signal (SIGHUP) to the `maxscale` process.

```
MaxScale> reload config
Reloading configuration from file.
MaxScale>
```

Note, not all configuration elements can be changed dynamically currently. This mechanism can be used to add new services, servers to services, listeners to services and to update passwords. It can not be used to remove services, servers or listeners currently.

## Add user

The `add user` command is used to add new users to the debug CLI of MaxScale. The default behaviour of the CLI for MaxScale is to have a login name of `admin` and a fixed password of `skysql`. Adding new users will disable this default behaviour and limit the login access to the users that are added.

```
MaxScale> add user admin july2013
User admin has been successfully added.
MaxScale> add user mark hambleden
User mark has been successfully added.
MaxScale>
```

User names must be unique within the debug CLI, this excludes the `admin` default user, which may be redefined.

```
MaxScale> add user mark 22july
User admin already exists.
MaxScale>
```

If you should forget or lose the the account details you may simply remove the `passwd` file in `$MAXSCALE_HOME/etc` and the system will revert to the default behaviour with `admin/skysql` as the account.



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

## Enable/disable log

The `enable/disable log` command is used to enable/disable the log facility of MaxScale. The default behaviour for MaxScale is to have all logs enabled in DEBUG version, and only error log in production release.

Examples:

```
MaxScale> help enable log
Available options to the enable command:
    log          Enable Log options for MaxScale, options trace | error | message
E.g. enable log message.

MaxScale> help disable log
Available options to the disable command:
    log          Disable Log for MaxScale, Options: debug | trace | error |
message E.g. disable log debug

MaxScale> disable log trace
MaxScale>
```

No output for these commands in the debug interface, but in the affected logs there is a message:

```
2013 11/14 16:08:33 ---      Logging is disabled  --
```

## Logging facility

MaxScale generates output of its behavior to four distinct logs, error, messages, trace and debug log. Error and message logs are enabled by default but all logs can be dynamically enabled and disabled by using [maxadm](#) utility, debug client interface (telnet) or optionally by using your own application through the client API.

### Log contents

By default all log files are located in : `$MAXSCALE_HOME/log` and named as :



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

skygw\_errW.log, skygw\_msgX.log, skygw\_traceY.log and skygw\_debugZ.log

where W, X, Y, and Z are sequence numbers for files within the same directory,

## Error log

Error log includes errors and warnings; things that occur during runtime and of which the user should at least be aware of. An entry in error log doesn't necessarily mean a problem - it may be a notification of failed backend server, for example.

Example:

```
MariaDB Corporation MaxScale      /home/jdoe/bin/develop/log/skygw_err1.log Mon Dec  8
13:28:15 2014
```

```
-----
---      Logging is enabled.
2014-12-08 13:28:15   Error : Unable to find filter 'testfi' for service 'RW Split
Router'
2014-12-08 13:28:26   MaxScale received signal SIGINT. Shutting down.
2014-12-08 13:28:27   MaxScale is shut down.
-----
```

## Message log

The content of message log consists of information about loaded modules, opened listen ports and other information about file locations etc. The amount of data written in message log is typically very small.

Example:

```
MariaDB Corporation MaxScale      /home/jdoe/bin/develop/log/skygw_msg1.log Tue Dec  9
14:47:05 2014
```

```
-----
---      Logging is enabled.
2014-12-09 14:47:05   Home directory      : /home/jdoe/bin/develop
2014-12-09 14:47:05   Data directory       : /home/jdoe/bin/develop/data/data5398
2014-12-09 14:47:05   Log directory        : /home/jdoe/bin/develop/log
2014-12-09 14:47:05   Configuration file   : /home/jdoe/bin/develop/etc/MaxScale.cnf
2014-12-09 14:47:05   Initialise CLI router module V1.0.0.
2014-12-09 14:47:05   Loaded module cli: V1.0.0 from
/home/jdoe/bin/develop/modules/libcli.so
2014-12-09 14:47:05   Initialise debug CLI router module V1.1.1.
2014-12-09 14:47:05   Loaded module debugcli: V1.1.1 from
/home/jdoe/bin/develop/modules/libdebugcli.so
2014-12-09 14:47:05   Loaded module testroute: V1.0.0 from
/home/jdoe/bin/develop/modules/libtestroute.so
2014-12-09 14:47:05   Initialise readconnroute router module V1.1.0.
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
2014-12-09 14:47:05   Loaded module readconnroute: V1.1.0 from
/home/jdoe/bin/develop/modules/libreadconnroute.so
2014-12-09 14:47:05   Initializing statemend-based read/write split router module.
2014-12-09 14:47:05   Loaded module readwritesplit: V1.0.2 from
/home/jdoe/bin/develop/modules/libreadwritesplit.so
2014-12-09 14:47:05   Initialise the MySQL Monitor module V1.4.0.
2014-12-09 14:47:05   Loaded module mysqlmon: V1.4.0 from
/home/jdoe/bin/develop/modules/libmysqlmon.so
2014-12-09 14:47:05   MariaDB Corporation MaxScale 1.0.1-beta (C) MariaDB Corporation
Ab 2013-2014
2014-12-09 14:47:05   MaxScale is running in process  5398
2014-12-09 14:47:05   Encrypted password file /home/jdoe/bin/develop/etc/.secrets
can't be accessed (No such file or directory). Password encryption is not used.
2014-12-09 14:47:05   Loaded 6 MySQL Users for service [RW Split Router].
2014-12-09 14:47:05   Loaded module MySQLClient: V1.0.0 from
/home/jdoe/bin/develop/modules/libMySQLClient.so
2014-12-09 14:47:05   Loaded 8 MySQL Users for service [Read Connection Router].
2014-12-09 14:47:05   Loaded module telnetd: V1.0.1 from
/home/jdoe/bin/develop/modules/libtelnetd.so
2014-12-09 14:47:05   Loaded module maxscaled: V1.0.0 from
/home/jdoe/bin/develop/modules/libmaxscaled.so
2014-12-09 14:47:05   Info: A Master Server is now available: 127.0.0.1:3000
2014-12-09 14:47:10   MaxScale is shut down.
```

-----

## Trace log

Trace log includes information about available servers and their states, client sessions, queries being executed, routing decisions and other routing related data. Trace log can be found from the same directory with other logs but it is physically stored elsewhere, to OSs shared memory to reduce the latency caused by logging. The location of physical file is :

`/dev/shm/<pid>/skygw_traceX.log` where 'X' is the same sequence number as in the file name in the `$MAXSCALE_HOME/log` directory.

Individual trace log entry looks similar to those in other logs but there is some difference too. Some log entries include a number within square brackets to specify which client session they belong to. For example:

```
2014-12-09 14:52:36   [6] Session write, routing to all servers.
```

Writing trace log for each client may produce so much data that it seriously affects on the performance of MaxScale. It may also be difficult to follow a specific session if the log is flooded with data from other sessions. While it is possible to dynamically enable and disable trace log as



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

a whole, one can also choose to explicitly enable trace logging for a specific session by first enabling trace log, finding out the session id of the interesting session, disabling trace log and finally enabling trace log only for a given session:

1. enable log trace (and examine the session id)
2. disable log trace
3. enable sessionlog trace <session id>

### Example of trace log:

```
MariaDB Corporation MaxScale      /dev/shm/5420/skygw_trace1.log Tue Dec  9 14:51:29
2014
-----
---   Logging is disabled.
2014-12-09 14:51:52   ---           Logging is enabled           --
2014-12-09 14:52:03   Servers and router connection counts:
2014-12-09 14:52:03   current operations : 0 in      127.0.0.1:3003 RUNNING SLAVE
2014-12-09 14:52:03   current operations : 0 in      127.0.0.1:3002 RUNNING SLAVE
2014-12-09 14:52:03   current operations : 0 in      127.0.0.1:3000 RUNNING MASTER
2014-12-09 14:52:03   Selected RUNNING SLAVE in      127.0.0.1:3003
2014-12-09 14:52:03   Selected RUNNING SLAVE in      127.0.0.1:3002
2014-12-09 14:52:03   Selected RUNNING MASTER in     127.0.0.1:3000

...

2014-12-09 14:52:28   [6] > Autocommit: [enabled], trx is [not open], cmd: COM_QUERY,
type: QUERY_TYPE_READ, stmt: select count(*) from user
2014-12-09 14:52:28   [6] Route query to slave      127.0.0.1:3003 <
2014-12-09 14:52:36   [6] Disable autocommit : implicit START TRANSACTION before
executing the next command.
2014-12-09 14:52:36   [6] > Autocommit: [disabled], trx is [open], cmd: COM_QUERY,
type: QUERY_TYPE_GSYSVAR_WRITE|QUERY_TYPE_BEGIN_TRX|QUERY_TYPE_DISABLE_AUTOCOMMIT,
stmt: set autocommit=0
2014-12-09 14:52:36   [6] Session write, routing to all servers.
2014-12-09 14:52:36   [6] Route query to slave      127.0.0.1:3003
2014-12-09 14:52:36   [6] Route query to slave      127.0.0.1:3002
2014-12-09 14:52:36   [6] Route query to master     127.0.0.1:3000 <
2014-12-09 14:52:39   [6] > Autocommit: [disabled], trx is [open], cmd: COM_QUERY,
type: QUERY_TYPE_BEGIN_TRX, stmt: begin
2014-12-09 14:52:39   [6] Route query to master     127.0.0.1:3000 <
2014-12-09 14:52:51   [6] > Autocommit: [disabled], trx is [open], cmd: COM_QUERY,
type: QUERY_TYPE_READ, stmt: select count(*) from user
2014-12-09 14:52:51   [6] Route query to master     127.0.0.1:3000 <
2014-12-09 14:52:59   Servers and router connection counts:
```



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

```
2014-12-09 14:52:59 current operations : 0 in 127.0.0.1:3003 RUNNING SLAVE
2014-12-09 14:52:59 current operations : 0 in 127.0.0.1:3002 RUNNING SLAVE
2014-12-09 14:52:59 current operations : 0 in 127.0.0.1:3000 RUNNING MASTER
2014-12-09 14:52:59 Selected RUNNING SLAVE in 127.0.0.1:3003
2014-12-09 14:52:59 Selected RUNNING SLAVE in 127.0.0.1:3002
2014-12-09 14:52:59 Selected RUNNING MASTER in 127.0.0.1:3000
2014-12-09 14:52:59 Started RW Split Router client session [7] for 'maxuser' from
127.0.0.1
```

In the log, session's life cycle is covered by annotating its beginning and the end. During the session, each statement is surrounded with '>' and '<' characters. Entries without '[<id>]' belong to other than client sessions, monitor, for example.

## Managing logs

The log files are located in

```
$MAXSCALE_HOME/log
```

by default. If, however, trace and debug logs are enabled, only a soft link is created there. MaxScale process creates a directory under

```
/dev/shm/<pid>
```

where it stores the physical trace and debug log files. Link and physical files share the same name. These logs consume the main memory of the host they run on so it is important to archive or remove them periodically to avoid unnecessary main-memory consumption.

## Rotating logs

Log files may grow very large over time and that is why users may want to split log files into several smaller ones. That is called log rotation. User can rotate all logs by executing

```
flush logs
```

Specific log file can be rotated by executing

```
flush log [error|message|trace|debug]
```

The commands above can be executed either by using `maxadmin` utility or via debug client API with telnet. The sequence number included in the log filename is used to separate files from each other. The logic behind sequence numbering is such that if the log directory is empty when



w: [www.mariadb.com](http://www.mariadb.com)

e: [info@mariadb.com](mailto:info@mariadb.com)

MaxScale is started, new log files will be created with sequence number 1 in their names, `skygw_errr1.log`, for example. If files of the same type already exist, the new MaxScale process opens the file with largest sequence number and applies into it. If existing log file isn't writable for the user that MaxScale runs on, new log file will be created with bigger sequence number.

More information about log files and administering them can be found from **MaxScale Administration Tutorial**.