MaxScale

Debug & Diagnostic Support

Mark Riddoch
Last Updated: 3rd March 2014

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

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()
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
        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)
```

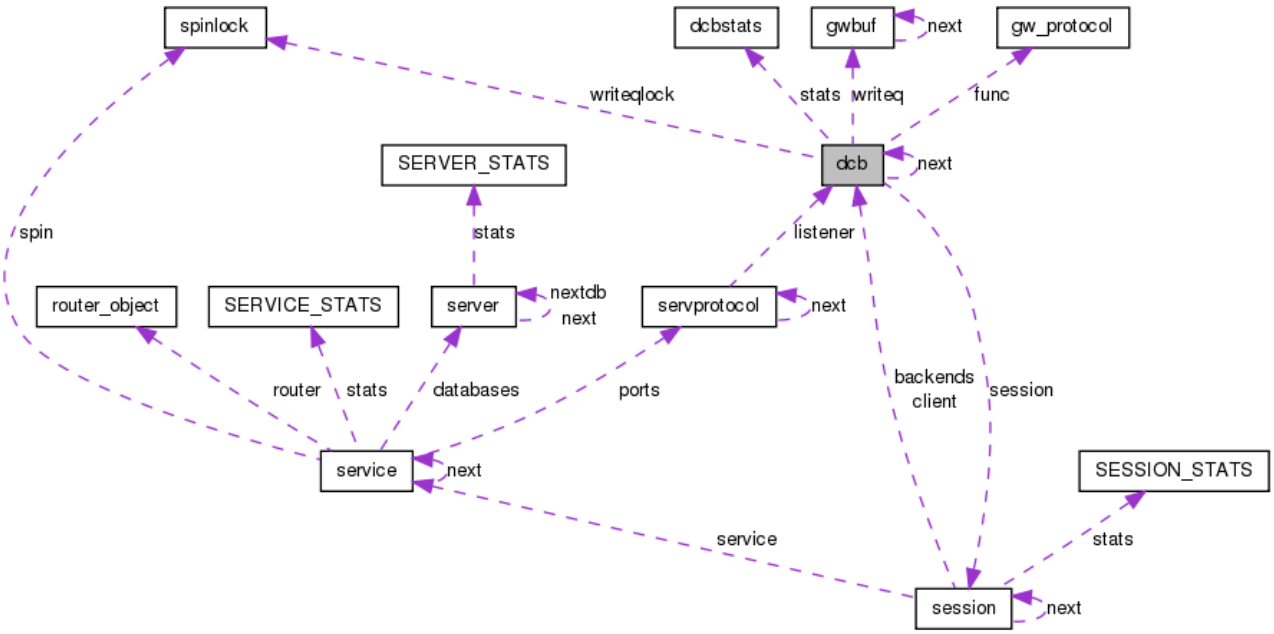
Modules

MaxScale makes significant use of moules, 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
--------------------------------------------------
telnetd          | Protocol    | V1.0.0
MySQLClient      | Protocol    | V1.0.0
testroute        | Router      | V1.0.0
debugcli         | Router      | V1.0.0
readconnroute    | Router      | V1.0.0
(gdb)
```

Descriptor Control Blocks

The Descriptor Control Block (DCB) is an important concept within MaxScale since it is this block that is passed to the polling system, when an event occurs it is that structure that is available and from this structure it must be possible to navigate to all other structures that contain state regarding the session and protocol in use.



Similar print routines exist for the DCB

```
(gdb) call printAllDCBs()
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
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
```

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 `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 (V1.0.1).
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[1], there is a help system that will display the commands available to you

```
MaxScale> help
Available commands:
    add user
    clear server
    restart monitor
    restart service
    set server
    show dcbs
    show dcb
    show dbusers
    show epoll
```

[1] Currently there is no authentication on this interface

```
    show modules
    show monitors
    show server
    show servers
    show services
    show session
    show sessions
    show users
    shutdown maxscale
    shutdown monitor
    shutdown service
    reload config
    reload users
    enable log
    disable log

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
    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.

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
          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)
        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
        Server:              127.0.0.1
        Status:                    Running
        Protocol:       MySQLBackend
        Port:           3306
        Number of connections:     0
        Current no. of connections:     0
Server 0x6ec770
        Server:              127.0.0.1
        Status:                    Master, Running
        Protocol:       MySQLBackend
        Port:           3307
        Number of connections:     1
        Current no. of connections:     1
Server 0x6ec6a0
        Server:              127.0.0.1
        Status:                    Slave, Running
        Protocol:       MySQLBackend
        Port:           3308
        Number of connections:     1
        Current no. of connections:     1
Server 0x6ec5d0
        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
        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
        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
```

```
readconnroute   | Router      | V1.0.2
readwritesplit  | Router      | V1.0.2
debugcli        | Router      | V1.0.1
MaxScale>
```

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 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
          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>
```

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
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
```

```
      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
      Monitored servers:   127.0.0.1:3310, 127.0.0.1:3311,
127.0.0.1:3312
MaxScale>
```

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
      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
      Monitored servers:   127.0.0.1:3306, 127.0.0.1:3307,
127.0.0.1:3308, 127.0.0.1:3309
MaxScale>
```

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
          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
```

Restart service

The `restart service` command can be used to restart a previously stopped listener for a service.

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

```
        Server:               127.0.0.1
        Status:                     Running
        Protocol:       MySQLBackend
        Port:           3306
        Number of connections:     0
        Current no. of connections:     0
Server 0x6ec770
        Server:               127.0.0.1
        Status:                     Master, Running
        Protocol:       MySQLBackend
        Port:           3307
        Number of connections:     1
        Current no. of connections:     0
Server 0x6ec6a0
        Server:               127.0.0.1
        Status:                     Slave, Running
        Protocol:       MySQLBackend
        Port:           3308
        Number of connections:     258
        Current no. of connections:     0
Server 0x6ec5d0
        Server:               127.0.0.1
        Status:                     Down
        Protocol:       MySQLBackend
        Port:           3309
        Number of connections:     0
        Current no. of connections:     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.

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
```

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:             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
                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>
```

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.

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 --