MaxAdmin

The MaxScale Administrative & Monitoring Client Application

Mark Riddoch

Last Updated: 26th June 2014

Contents

Overview

MaxAdmin is a simple client interface that can be used to interact with the MaxScale server, it allows the display of internal MaxScale statistics, status and control of MaxScale operations.

MaxAdmin supports

- Interactive user sessions
- Execution of one-off commands via command line arguments
- Execution of command scripts

Running MaxAdmin

The MaxAdmin client application may be run in two different modes, either as an interactive command shell for executing commands against MaxScale or by passing commands on the MaxAdmin command line itself.

Command Line Switches

The MaxAdmin command accepts a number of switches

Switch	**Description**
-u user	Sets the username that will be used for the MaxScale connection. I no -u option is passed on the MaxAdmin command line then the default username of ‘admin’ will be used.
-p password	Sets the user password that will be used. If no -p option is passed on the command line then MaxAdmin will prompt for interactive entry of the password.
-h hostname	The hostname of the MaxScale server to connect to. If no -h option is passed on the command line then MaxAdmin will attempt to connect to the host ‘localhost’.
-P port	The port that MaxAdmin will use to connect to the MaxScale server if no -P option is given then the default port of 6603 will be used.
--help	Print usage information regarding MaxAdmin

When a switch takes a value, this may either be as the next argument on the command line or maybe as part of the switch itself. E.g. `-u me` and `-ume` are treated in the same way.

Interactive Operation

If no arguments other than the command line switches are passed to MaxAdmin it will enter its interactive mode of operation. Users will be prompted to enter commands with a **`MaxScale>`**

prompt. The commands themselves are documented in the sections later in this document. A help system is available that will give some minimal details of the commands available.

Command history is available on platforms that support the libedit library. This allows the use of the up and down arrow keys to recall previous commands that have been executed by MaxAdmin. The default edit mode for the history is to emulate the vi commands, the behaviour of libedit may however be customised using the .editrc file. To obtain the history of commands that have been executed use the inbuilt `history` command.

In interactive mode it is possible to execute a set of commands stored in an external file by using the `source` command. The command takes the argument of a filename which should contain a set of MaxScale commands, one per line. These will be executed in the order they appear in the file.

Command Line Operation

MaxAdmin can also be used to execute commands that are passed on the command line, e.g.

```
-bash-4.1$ maxadmin -hmaxscale list services
Password:
Services.
--------------------------+----------------------+--------+---------------
Service Name              | Router Module        | #Users | Total Sessions
--------------------------+----------------------+--------+---------------
Test Service              | readconnroute        |      1 |     1
Split Service             | readwritesplit       |      1 |     1
Filter Service            | readconnroute        |      1 |     1
QLA Service               | readconnroute        |      1 |     1
Debug Service             | debugcli             |      1 |     1
CLI                       | cli                  |      2 |    27
--------------------------+----------------------+--------+---------------

-bash-4.1$
```

The single command is executed and MaxAdmin then terminates. If the -p option is not given then MaxAdmin will prompt for a password. If a MaxScale command requires an argument which contains whitespace, for example a service name, that name should be quoted. The quotes will be preserved and used in the execution of the MaxScale command.

```
-bash-4.1$ maxadmin show service "QLA Service"
Password:
Service 0x70c6a0
       Service:            QLA Service
       Router:                   readconnroute (0x7ffff0f7ae60)
       Number of router sessions:       0
       Current no. of router sessions:  0
       Number of queries forwarded:     0
       Started:            Wed Jun 25 10:08:23 2014
       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:         0x724340
       Total connections:  1
       Currently connected:1
-bash-4.1$
```

Command files may be executed by either calling MaxAdmin with the name of the file that contains the commands

```
maxadmin listall.ms
```

Or by using the #! mechanism to make the command file executable from the shell. To do this add a line at the start of your command file that contains the #! directive with the path of the MaxAdmin executable. Command options may also be given in this line. For example to create a script file that runs a set of list commands

```
#!/usr/local/bin/maxadmin -hmaxscalehost
list modules
list servers
list services
list listeners
list dcbs
list sessions
list filters
```

Then simply set this file to have execute permissions and it may be run like any other command in the Linux shell.

Getting Help

A help system is available that describes the commands available via the administration interface. To obtain a list of all commands available simply type the command `help`.

```
MaxScale> help
Available commands:
    add user
    clear server
    disable [heartbeat|log|root]
    enable [heartbeat|log|root]
    list
[clients|dcbs|filters|listeners|modules|monitors|services|servers|sessions]
    reload [config|dbusers]
    remove user
    restart [monitor|service]
    set server
    show
[dcbs|dcb|dbusers|epoll|filter|filters|modules|monitor|monitors|server|servers|ser
vices|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>
```

To see more detail on a particular command, and a list of the sub commands of the command, type help followed by the command name.

```
MaxScale> help list
Available options to the list command:
    clients    List all the client connections to MaxScale
    dcbs       List all the DCBs active within MaxScale
    filters    List all the filters defined within MaxScale
    listeners  List all the listeners defined within MaxScale
    modules    List all currently loaded modules
    monitors   List all monitors
    services   List all the services defined within MaxScale
    servers    List all the servers defined within MaxScale
    sessions   List all the active sessions within MaxScale
MaxScale>
```

Working With Services

A service is a very important concept in MaxScale as it defines the mechanism by which clients interact with MaxScale and can attached to the backend databases. A number of commands exist that allow interaction with the services.

What Services Are Available?

The `list services` command can be used to discover what services are currently available within your MaxScale configuration.

```
MaxScale> list services
Services.
--------------------------+----------------------+--------+---------------
Service Name              | Router Module        | #Users | Total Sessions
--------------------------+----------------------+--------+---------------
Test Service              | readconnroute        |      1 |     1
Split Service             | readwritesplit       |      1 |     1
Filter Service            | readconnroute        |      1 |     1
QLA Service               | readconnroute        |      1 |     1
Debug Service             | debugcli             |      1 |     1
CLI                       | cli                  |      2 |    24
--------------------------+----------------------+--------+---------------

MaxScale>
```

In order to determine which ports services are using then the `list listeners` command can be used.

```
MaxScale> list listeners
Listeners.
---------------------+--------------------+-----------------+-------+--------
Service Name         | Protocol Module    | Address         | Port  | State
---------------------+--------------------+-----------------+-------+--------
Test Service         | MySQLClient        | *               |  4006 | Running
Split Service        | MySQLClient        | *               |  4007 | Running
Filter Service       | MySQLClient        | *               |  4008 | Running
QLA Service          | MySQLClient        | *               |  4009 | Running
Debug Service        | telnetd            | localhost       |  4242 | Running
CLI                  | maxscaled          | localhost       |  6603 | Running
---------------------+--------------------+-----------------+-------+--------

MaxScale>
```

See Service Details

It is possible to see the details of an individual service using the `show service` command. This command should be passed the name of the service you wish to examine as an argument.

Where a service name contains spaces characters there should either be escaped or the name placed in quotes.

```
MaxScale> show service "QLA Service"
Service 0x70c6a0
        Service:                         QLA Service
        Router:                          readconnroute (0x7ffff0f7ae60)
        Number of router sessions:       0
        Current no. of router sessions:  0
        Number of queries forwarded:     0
        Started:                         Wed Jun 25 10:08:23 2014
        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:                      0x724340
        Total connections:               1
        Currently connected:             1
MaxScale>
```

This allows the set of backend servers defined by the service to be seen along with the service statistics and other information.

Examining Service Users

MaxScale provides an authentication model by which the client application authenticates with MaxScale using the credentials they would normally use to with the database itself. MaxScale loads the user data from one of the backend databases defined for the service. The `show dbusers` command can be used to examine the user data held by MaxScale.

```
MaxScale> show dbusers "Filter Service"
Users table data
Hashtable: 0x723e50, size 52
        No. of entries:     48
        Average chain length:      0.9
        Longest chain length:      5
User names: pappo@%, rana@%, new_control@%, new_nuovo@%, uno@192.168.56.1,
nuovo@192.168.56.1, pesce@%, tryme@192.168.1.199, repluser@%, seven@%, due@%,
pippo@%, mmm@%, daka@127.0.0.1, timour@%, ivan@%, prova@%, changeme@127.0.0.1,
uno@%, massimiliano@127.0.0.1, massim@127.0.0.1, massi@127.0.0.1,
masssi@127.0.0.1, pappo@127.0.0.1, rana@127.0.0.1, newadded@127.0.0.1,
newaded@127.0.0.1, pesce@127.0.0.1, repluser@127.0.0.1, seven@127.0.0.1,
pippo@127.0.0.1, due@127.0.0.1, nopwd@127.0.0.1, timour@127.0.0.1,
controlla@192.168.56.1, ivan@127.0.0.1, ppp@127.0.0.1, daka@%, nuovo@127.0.0.1,
uno@127.0.0.1, repluser@192.168.56.1, havoc@%, tekka@192.168.1.19,
due@192.168.56.1, qwerty@127.0.0.1, massimiliano@%, massi@%, massim@%
MaxScale>
```

Reloading Service User Data

MaxScale will automatically reload user data if there are failed authentication requests from client applications. This reloading is rate limited and triggered by missing entries in the MaxScale table. If a user is removed from the backend database user table it will not trigger removal from the MaxScale internal table. The `reload dbusers` command can be used to force the reloading of the user table within MaxScale.

```
MaxScale> reload dbusers "Split Service"
Loaded 34 database users for service Split Service.
MaxScale>
```

Stopping A Service

It is possible to stop a service from accepting new connections by using the `shutdown service` command. This will not affect the connections that are already in place for a service, but will stop any new connections from being accepted.

```
MaxScale> shutdown service "Split Service"
MaxScale>
```

Restart A Stopped Service

A stopped service may be restarted by using the `restart service` command.

```
MaxScale> restart service "Split Service"
MaxScale>
```

Working With Servers

The server represents each of the instances of MySQL or MariaDB that a service may use.

What Servers Are Configured?

The command `list servers` can be used to display a list of all the servers configured within MaxScale.

```
MaxScale> list servers
Servers.
-------------------+-----------------+-------+----------------------+------------
Server             | Address         | Port  | Status               | Connections
-------------------+-----------------+-------+----------------------+------------
server1            | 127.0.0.1       |  3306 | Running              |    0
server2            | 127.0.0.1       |  3307 | Master, Running      |    0
server3            | 127.0.0.1       |  3308 | Running              |    0
server4            | 127.0.0.1       |  3309 | Slave, Running       |    0
-------------------+-----------------+-------+----------------------+------------

MaxScale>
```

Server Details

It is possible to see more details regarding a given server using the show `server command`.

```
MaxScale> show server server2
Server 0x70d460 (server2)
       Server:                    127.0.0.1
       Status:                    Master, Running
       Protocol:                  MySQLBackend
       Port:                      3307
       Server Version:            5.5.25-MariaDB-log
       Node Id:                   124
       Number of connections:     0
       Current no. of conns:      0
MaxScale>
```

Setting The State Of A Server

MaxScale maintains a number of status bits for each server that is configured, these status bits are normally maintained by the monitors, there are two commands in the user interface that are used to manually maintain these bits also; the `set server` and `clear server` commands.

The status bit that can be controlled are

Bit Name	**Description**
running	The server is responding to requests, accepting connections and executing database commands
master	The server is a master in a replication setup or should be considered as a destination for database updates.
slave	The server is a replication slave or is considered as a read only database.
synced	The server is a fully fledged member of a Galera cluster
maintenance	The server is in maintenance mode. In this mode no new connections will be established to the server. The monitors will also not monitor servers that are in maintenance mode.

All status bits, with the exception of the maintenance bit, will be set by the monitors that are monitoring the server. If manual control is required the monitor should be stopped.

```
MaxScale> set server server3 maintenance
MaxScale> clear server server3 maintenance
MaxScale>
```

Working With Sessions

The MaxScale session represents the state within MaxScale. Sessions are dynamic entities and not named in the configuration file, this means that sessions can not be easily named within the user interface. The sessions are referenced using ID values, these are actually memory address, however the important thing is that no two session have the same ID.

What Sessions Are Active in MaxScale?

There are a number of ways to find out what sessions are active, the most comprehensive being the `list sessions` command.

```
MaxScale> list sessions
Sessions.
-----------------+-----------------+----------------+--------------------------
Session          | Client          | Service        | State
-----------------+-----------------+----------------+--------------------------
0x7267a0         | 127.0.0.1       | CLI            | Session ready for routing
0x726340         |                 | CLI            | Listener Session
0x725720         |                 | Debug Service  | Listener Session
0x724720         |                 | QLA Service    | Listener Session
0x72a750         |                 | Filter Service | Listener Session
0x709500         |                 | Split Service  | Listener Session
0x7092d0         |                 | Test Service   | Listener Session
-----------------+-----------------+----------------+--------------------------

MaxScale>
```

This lists all the sessions for both user connections and for the service listeners.

The `list clients` command will give just the subset of sessions that originate from a client connection.

```
MaxScale> list clients
Client Connections
-----------------+------------+----------------------+------------
 Client          | DCB        | Service              | Session
-----------------+------------+----------------------+------------
 127.0.0.1       |   0x7274b0 | CLI                  |   0x727700
 127.0.0.1       |   0x727900 | QLA Service          |   0x727da0
-----------------+------------+----------------------+------------

MaxScale>
```

Display Session Details

Once the session ID has been determined using one of the above method it is possible to determine more detail regarding a session by using the `show session` command.

```
MaxScale> show session 0x727da0
Session 0x727da0
       State:              Session ready for routing
       Service:            QLA Service (0x70d6a0)
       Client DCB:         0x727900
       Client Address:     127.0.0.1
       Connected:          Wed Jun 25 15:27:21 2014
MaxScale>
```

Descriptor Control Blocks

The Descriptor Control Block or DCB is a very important entity within MaxScale, it represents the state of each connection within MaxScale. A DCB is allocated for every connection from a client, every network listener and every connection to a backend database. Statistics for each of these connections are maintained within these DCB's.

As with session above the DCB's are not named and are therefore referred to by the use of a unique ID, the memory address of the DCB.

Finding DCB's

There are several ways to determine what DCB's are active within a MaxScale server, the most straightforward being the `list dcbs` command.

```
MaxScale> list dcbs
Descriptor Control Blocks
------------+----------------------------+----------------------+----------
 DCB        | State                      | Service              | Remote
------------+----------------------------+----------------------+----------
   0x667170 | DCB for listening socket   | Test Service         |
   0x71a350 | DCB for listening socket   | Split Service        |
   0x724b40 | DCB for listening socket   | Filter Service       |
   0x7250d0 | DCB for listening socket   | QLA Service          |
   0x725740 | DCB for listening socket   | Debug Service        |
   0x726740 | DCB for listening socket   | CLI                  |
   0x7274b0 | DCB in the polling loop    | CLI                  | 127.0.0.1
   0x727900 | DCB in the polling loop    | QLA Service          | 127.0.0.1
   0x72e880 | DCB in the polling loop    | QLA Service          |
------------+----------------------------+----------------------+----------

MaxScale>
```

A MaxScale server that has activity on it will however have many more DCB's than in the example above, making it hard to find the DCB that you require. The DCB ID is also included in a number of other command outputs, depending on the information you have it may be easier to use other methods to locate a particular DCB.

DCB Of A Client Connection

To find the DCB for a particular client connection it may be best to start with the `list clients` command and then look at each DCB for a particular client address to determine the one of interest.

DCB Details

The details of an individual DCB can be obtained by use of the `show dcb` command

```
MaxScale> show dcb 0x727900
DCB: 0x727900
       DCB state:          DCB in the polling loop
       Connected to:       127.0.0.1
       Owning Session:     0x727da0
       Statistics:
             No. of Reads:                     4
             No. of Writes:                    3
             No. of Buffered Writes:           0
             No. of Accepts:                   0
             No. of High Water Events:         0
             No. of Low Water Events:          0
MaxScale>
```

Working with Filters

Filters allow the request contents and result sets from a database to be modified for a client connection, pipelines of filters can be created between the client connection and MaxScale router modules.

What Filters Are Configured?

Filters are configured in the configuration file for MaxScale, they are given names and may be included in the definition of a service. The `list filters` command can be used to determine which filters are defined.

```
MaxScale> list filters
Filters
--------------------+-----------------+----------------------------------------
Filter              | Module          | Options
--------------------+-----------------+----------------------------------------
counter             | testfilter      |
QLA                 | qlafilter       | /tmp/QueryLog
Replicate           | tee             |
QLA_BLR             | qlafilter       | /tmp/QueryLog.blr0
regex               | regexfilter     |
MySQL5.1            | regexfilter     |
top10               | topfilter       |
--------------------+-----------------+----------------------------------------

MaxScale>
```

Retrieve Details Of A Filter Configuration

The command show filter can be used to display information related to a particular filter.

```
MaxScale> show filter QLA
Filter 0x719460 (QLA)
       Module:      qlafilter
       Options:     /tmp/QueryLog
              Limit logging to connections from 127.0.0.1
              Include queries that match       select.*from.*user.*where
MaxScale>
```

Filter Usage

The `show session` command will include details for each of the filters in use within a session. First use `list sessions` or `list clients` to find the session of interest and then run the `show session` command

```
MaxScale> list clients
Client Connections
-----------------+------------+----------------------+------------
```

```
 Client          | DCB        | Service              | Session
-----------------+------------+----------------------+------------
 127.0.0.1       |   0x7361a0 | Split Service        |   0x736680
 127.0.0.1       |   0x737ec0 | Plumbing             |   0x7382b0
 127.0.0.1       |   0x73ab20 | DigitalOcean         |   0x73ad90
 127.0.0.1       |   0x7219e0 | CLI                  |   0x721bd0
-----------------+------------+----------------------+------------

MaxScale> show session 0x736680
Session 0x736680
       State:              Session ready for routing
       Service:            Split Service (0x719f60)
       Client DCB:         0x7361a0
       Client Address:            127.0.0.1
       Connected:          Thu Jun 26 10:10:44 2014
       Filter: top10
             Report size                10
             Logging to file /tmp/Query.top10.1.
             Current Top 10:
             1 place:
                    Execution time: 23.826 seconds
                    SQL: select sum(salary), year(from_date) from salaries s,
(select distinct year(from_date) as y1 from salaries) y where (makedate(y.y1, 1)
between s.from_date and s.to_date) group by y.y1 ("1988-08-01?
             2 place:
                    Execution time: 5.251 seconds
                    SQL: select d.dept_name as "Department", y.y1 as "Year",
count(*) as "Count" from departments d, dept_emp de, (select distinct
year(from_date) as y1 from dept_emp order by 1) y where d.dept_no = de.dept_no and
(makedate(y.y1, 1) between de.from_date and de.to_date) group by y.y1, d.dept_name
order by 1, 2
             3 place:
                    Execution time: 2.903 seconds
                    SQL: select year(now()) - year(birth_date) as age, gender,
avg(salary) as "Average Salary" from employees e, salaries s where e.emp_no =
s.emp_no and ("1988-08-01"  between from_date AND to_date) group by year(now()) -
year(birth_date), gender order by 1,2
             4 place:
                    Execution time: 2.138 seconds
                    SQL: select dept_name as "Department", sum(salary) / 12 as
"Salary Bill" from employees e, departments d, dept_emp de, salaries s where
e.emp_no = de.emp_no and de.dept_no = d.dept_no and ("1988-08-01"  between
de.from_date AND de.to_date) and ("1988-08-01"  between s.from_date AND s.to_date)
and s.emp_no = e.emp_no group by dept_name order by 1
             5 place:
                    Execution time: 0.839 seconds
                    SQL: select dept_name as "Department", avg(year(now()) -
year(birth_date)) as "Average Age", gender from employees e, departments d,
dept_emp de where e.emp_no = de.emp_no and de.dept_no = d.dept_no and
("1988-08-01"  between from_date AND to_date) group by dept_name, gender
```

```
             6 place:
                    Execution time: 0.662 seconds
                    SQL: select year(hire_date) as "Hired", d.dept_name, count(*)
as "Count" from employees e, departments d, dept_emp de where de.emp_no = e.emp_no
and de.dept_no = d.dept_no group by d.dept_name, year(hire_date)
             7 place:
                    Execution time: 0.286 seconds
                    SQL: select moves.n_depts As "No. of Departments",
count(moves.emp_no) as "No. of Employees" from (select de1.emp_no as emp_no,
count(de1.emp_no) as n_depts from dept_emp de1 group by de1.emp_no) as moves group
by moves.n_depts order by 1
             8 place:
                    Execution time: 0.248 seconds
                    SQL: select year(now()) - year(birth_date) as age, gender,
count(*) as "Count" from employees group by year(now()) - year(birth_date), gender
order by 1,2@
             9 place:
                    Execution time: 0.182 seconds
                    SQL: select year(hire_date) as "Hired", count(*) as "Count"
from employees group by year(hire_date)
             10 place:
                    Execution time: 0.169 seconds
                    SQL: select year(hire_date) - year(birth_date) as "Age",
count(*) as Count from employees group by year(hire_date) - year(birth_date) order
by 1

MaxScale>
```

The data displayed varies from filter to filter, the example above is the top filter. This filter prints a report of the current top queries at the time the `show session` command is run.

Working With Monitors

Monitors are used to monitor the state of databases within MaxScale in order to supply information to other modules, specifically the routers within MaxScale.

What Monitors Are Running?

To see what monitors are running within MaxScale use the `list monitors` command.

```
MaxScale> list monitors
+----------------------+---------------------
| Monitor              | Status
+----------------------+---------------------
| MySQL Monitor        | Running
+----------------------+---------------------
MaxScale>
```

Details Of A Particular Monitor

To see the details of a particular monitor use the `show monitor` command.

```
MaxScale> show monitor "MySQL Monitor"
Monitor: 0x71c370
       Name:        MySQL Monitor
       Monitor running
       Sampling interval:  10000 milliseconds
       MaxScale MonitorId: 24209641
       Replication lag:    disabled
       Monitored servers:  127.0.0.1:3306, 127.0.0.1:3307, 127.0.0.1:3308, 
127.0.0.1:3309
MaxScale>
```

Controlling Replication Heartbeat

Some monitors provide a replication heartbeat mechanism that monitors the delay for data that is replicated from a master to slaves in a tree structured replication environment. This can be enabled or disabled using the commands `enable heartbeat` and `disable heartbeat`.

```
MaxScale> disable heartbeat "MySQL Monitor"
MaxScale> enable heartbeat "MySQL Monitor"
MaxScale>
```

Please note that changes made via this interface will not persist across restarts of MaxScale. To make a permanent change edit the MaxScale.cnf file.

Enabling the replication heartbeat mechanism will add the display of heartbeat information in the show server output

```
MaxScale> show server server4
Server 0x719800 (server4)
       Server:                    127.0.0.1
       Status:                    Slave, Running
       Protocol:                  MySQLBackend
       Port:                      3309
       Server Version:            5.5.25-MariaDB-log
       Node Id:                   4
       Number of connections:     0
       Current no. of conns:      0
MaxScale> enable heartbeat "MySQL Monitor"
MaxScale> show server server4
Server 0x719800 (server4)
       Server:                    127.0.0.1
       Status:                    Slave, Running
       Protocol:                  MySQLBackend
       Port:                      3309
       Server Version:            5.5.25-MariaDB-log
       Node Id:                   4
       Slave delay:               0
       Last Repl Heartbeat:       Thu Jun 26 17:04:58 2014
       Number of connections:     0
       Current no. of conns:      0
MaxScale>
```

Shutting Down A Monitor

A monitor may be shutdown using the `shutdown monitor` command. This allows for manual control of the status of servers using the `set server` and `clear server` commands.

```
MaxScale> shutdown monitor "MySQL Monitor"
MaxScale> list monitors
+-----------------------+----------------------
| Monitor               | Status
+-----------------------+----------------------
| MySQL Monitor         | Stopped
+-----------------------+----------------------
MaxScale>
```

Restarting A Monitor

A monitor that has been shutdown may be restarted using the `restart monitor` command.

```
MaxScale> restart monitor "MySQL Monitor"
MaxScale> show monitor "MySQL Monitor"
Monitor: 0x71a310
       Name:        MySQL Monitor
       Monitor running
       Sampling interval:  10000 milliseconds
       MaxScale MonitorId: 24201552
       Replication lag:    enabled
       Monitored servers:  127.0.0.1:3306, 127.0.0.1:3307, 127.0.0.1:3308,
127.0.0.1:3309
MaxScale>
```

Working With Administration Interface Users

A default installation of MaxScale allows connection to the administration interface using the username of `admin` and the password `skysql`. This username and password stay in effect as long as no other users have been created for the administration interface. As soon as the first user is added the use of admin/skysql as login credentials will be disabled.

What Users Have Been Defined?

In order to see the current users that have been defined for the administration interface use the command `show users`.

```
        MaxScale> show users
        Administration interface users:
        Users table data
        Hashtable: 0x734470, size 52
                No. of entries:           5
                Average chain length:     0.1
                Longest chain length:     2
        User names: vilho, root, dba, massi, mark
        MaxScale>
```

Please note that if no users have been configured the default admin/skysql user will not be shown.

```
        MaxScale> show users
        Administration interface users:
        No administration users have been defined.
        MaxScale>
```

Add A New User

To add a new administrative user to the MaxScale server use the command add user. This command is passed a user name and a password.

```
        MaxScale> add user maria dtbse243
        User maria has been successfully added.
        MaxScale>
```

Delete A User

To remove a user the command `remove user` is used, it must also be called with the username and password of the user. The password will be checked.

```
MaxScale> remove user maria des
Failed to remove user maria. Authentication failed
MaxScale> remove user maria dtbse243
User maria has been successfully removed.
MaxScale>
```

Administration Commands

What Modules Are In use?

In order to determine what modules are in use, and the version and status of those modules the `list modules` command can be used.

```
MaxScale> list modules
Modules.
----------------+-------------+---------+-------+-------------------------
Module Name     | Module Type | Version | API   | Status
----------------+-------------+---------+-------+-------------------------
tee             | Filter      | V1.0.0  | 1.1.0 | Alpha
qlafilter       | Filter      | V1.1.1  | 1.1.0 | Alpha
topfilter       | Filter      | V1.0.1  | 1.1.0 | Alpha
MySQLBackend    | Protocol    | V2.0.0  | 1.0.0 | Alpha
maxscaled       | Protocol    | V1.0.0  | 1.0.0 | Alpha
telnetd         | Protocol    | V1.0.1  | 1.0.0 | Alpha
MySQLClient     | Protocol    | V1.0.0  | 1.0.0 | Alpha
mysqlmon        | Monitor     | V1.2.0  | 1.0.0 | Alpha
readconnroute   | Router      | V1.0.2  | 1.0.0 | Alpha
readwritesplit  | Router      | V1.0.2  | 1.0.0 | Alpha
debugcli        | Router      | V1.1.1  | 1.0.0 | Alpha
cli             | Router      | V1.0.0  | 1.0.0 | Alpha
----------------+-------------+---------+-------+-------------------------

MaxScale>
```

This command provides important version information for the module. Each module has two versions; the version of the module itself and the version of the module API that it supports. Also included in the output is the status of the module, this may be “In Development”, “Alpha”, “Beta”, “GA” or “Experimental”.

Change MaxScale Logging Options

Two commands are provided to change the logging levels within MaxScale, `disable log` and `enable log`. Using these commands the various log levels can be turned on and off, the supported levels are trace, debug and message. The error log level can not be turned off.

```
MaxScale> enable log trace
MaxScale> disable log debug
MaxScale>
```

Please note that changes made via this interface will not persist across restarts of MaxScale. To make a permanent change edit the MaxScale.cnf file.

Reloading The Configuration

A command, `reload config`, is available that will cause MaxScale to reload the MaxScale.cnf configuration file.

Shutting Down MaxScale

The MaxScale server may be shutdown using the `shutdown maxscale` command.

Configuring MaxScale to Accept MaxAdmin Connections

In order to allow the use of the MaxAdmin client interface the service must be added to the MaxScale.cnf file of the Maxscale server. The CLI service itself must be added and a listener for the maxscaled protocol.

The default entries required are shown below.

```
[CLI]
type=service
router=cli

[CLI Listener]
type=listener
service=CLI
protocol=maxscaled
address=localhost
port=6603
```

Note that this uses the default port of 6603 and confines the connections to localhost connections only. Remove the `address=` entry to allow connections from any machine on your network. Changing the port from 6603 will mean that you must allows pass a -p option to the MaxAdmin command.