RBFS Logging Guide

Version 21.1.1, 29 January 2021

Registered Address	Support	Sales
26, Kingston Terrace, Princeton, New Jersey 08540, United States		
		+91 80 4850 5445
http://www.rtbrick.com	support@rtbrick.com	sales@rtbrick.com

Table of Contents

1. RBFS Log Files

RBFS logging infrastructure provides in-memory (BDS) and traditional (BD) logging support for RBFS applications. The BDS logging is a low latency in-memory logging which can be used in a high scale system without compromising much in performance whereas BD logging is a direct write to a file hence CPU heavy.

The BDS logging infrastructure uses the following BDS tables:

Log map	This is a statically defined table, which contains log IDs per module with required detail for each ID (such as format string, plugin, log level, log table). This table is present in all BDs.
local Log	This stores log arguments for each log ID. This is present in all BDs.
config log	Stores logging configuration. This is present in all BDs.
plug-in server	Stores plug-in server logging configuration. This is present in all BDs.

1.1. Guidelines and Limitations

- Logging is enabled and the log-level is set to Error by default (both system and BDS logging).
- You can enable logging via CLI, but follow the guidelines below to enable logging at high scale.
  - Logging for BDS and PUBSUB modules have been disabled by default so even if you set the global log level, you will not see any BDS and PUBSUB module logs. Since these two modules are infrastructure-specific, these logs may not be useful for end-users; however, developers can enable logging for these modules using debug commands.
- Do not enable logging for all BDs; enable logging for the BD which is problematic
- Do not enable BDS logging at the global level, instead enable logging for the module you want to debug
- Do not keep the logging enabled for longer duration in a scaled setup
- The following log levels are present in the system; anything above Warning indicates that you need to logging with caution as a scaled system might get into an unstable situation.
  - EMERGENCY
  - ALERT
  - CRITICAL

- ERROR
- WARNING
- NOTICE
- INFO
- DEBUG
- NONE

If you get into a problematic situation, logging can be disabled by deleting all the logging configuration.

- Logging is enabled if any configuration exists in the BD start-up config files or configured via CLI.
- To disable logging, delete all logging configurations.
- Logging supports log file rotation.

The components of logging are:

Log Tables	Table per module and per BD
Modules	Shared library used by multiple BDs
Log Entries	Objects in log tables
Log Entry Attributes	lod_id, log_time, log_level etc.
Log Maps	Map log ID in log entry to log name and log string
Log Groups	Collection of log maps

1.2. Understanding the Log Maps

Every log map is mapped to one specific event that is logged by the application. For the optimized usage of memory, RBFS does not store the verbose strings; rather, it stores the log map as an identifier to the actual string message.

You can access these log maps by navigating into to the following location:

```
/usr/share/rtbrick/liblog/logs/
```

Here, you can see the log maps organized by the modules that they belong to.

```
ubuntu@bgp_rt1:/usr/share/rtbrick/liblog/logs$ ls
bds  bgp  fib  fwdinfra  ifm  lldpv2  pd  policy  pubsub  resmon  rib
snapshot  static  time-series
```

In the example above, you can see the modules that have registered the log maps.

If you want to understand the significance of a particular log map, you can simply *grep* the logmap in this directory.

By running the **show log status** command, you can find information about all the possible log maps and the modules in the system.

```
ubuntu@bgp_rt1:~$ rtb confd show log status
 Global Log Satus:
  Logging Enabled             :       true
  Logging Level               :       None
   Module Log Status:

Module [ bds ] Level Info
     Log ID Status:
       BDS_ATTRIBUTE_TEMPLATE_EVENT            , [Level] Info      , [Plugin]
None      , [Render] None
 ....
 ....
 ....
```

Now, if you want to know what a specific log map indicates, for example, "BGP_UPD_NLRI_FAIL", you can do the following:

```
supervisor@dev:/usr/share/rtbrick/liblog/logs/bgp$ cat *.json | jq
'.log_maps[]|select(.log_name=="BGP_UPD_NLRI_FAIL")'
{
  "log_id": 546,
  "log_name": "BGP_UPD_NLRI_FAIL",
  "log_level": "Info",
  "group_name": "message",
  "log_string": "BGP update message processing, peer %s, source %s, hostname
%s, instance %s: Processing NLRI attribute failed\n",
  "args": [
    "string",
    "string",
    "string",
    "string"
  ],
  "expose": [
    "bgp_peer_ip",
    "bgp_source_ip",
    "bgp_host_name",
    "instance"
  ],
  "description": "BGP Update processing: Processing NLRI attribute failed"
}
```

To enable logging for above Log ID, configure log-level as Info for the module "BGP" and log-group "message"

Here, the log_string displays a human-readable interpretation of the log. The description indicates the description of the log string. The log_level indicates the log-level (Info or above) that you need to enable.

1.3. Understanding the Log Groups

A log group is a collection of log maps. This has been introduced to simplify the log configuration tasks. For example, to debug a BGP Peer issue, instead of enabling logs for individual log IDs that are related to BGP peer, you can enable log for a log group BGP Peer.

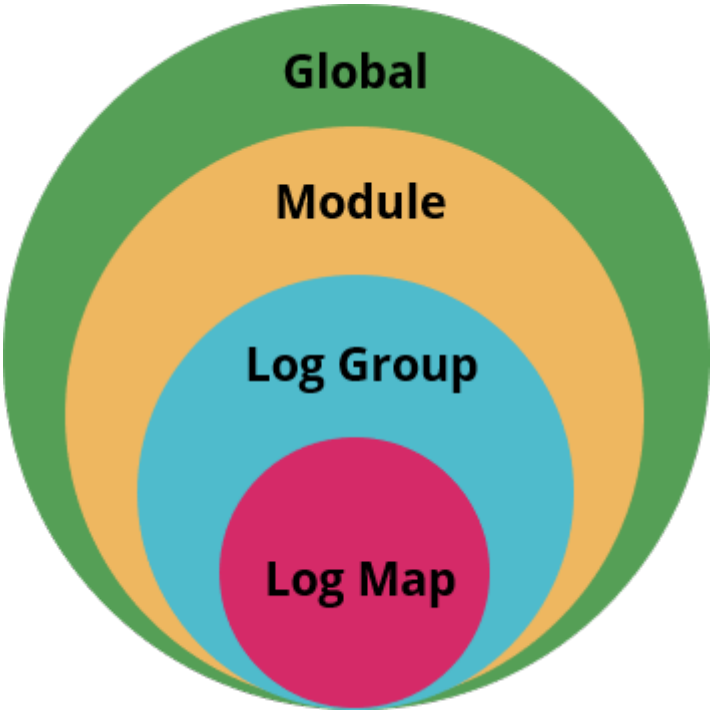
2. Configuring Logs

2.1. Logging Hierarchy

Logging can be configured at the following three hierarchies:

- Global
- Module
- Log Group

The figure below shows the preference of these levels:



If no configuration exists for a hierarchy, then the log-level will be inherited from the upper hierarchy. Otherwise, whatever is configured for the specific hierarchy will take priority.

2.2. Global Configuration

- Log level – default: 'none'

Each individual hierarchy can be configured with the following log levels:

- EMERGENCY
- ALERT
- CRITICAL

- ERROR
- WARNING
- NOTICE
- INFO
- DEBUG
- NONE

- All log levels lesser than the log level specified are logged. For example, if the specified log level is "WARNING", then all logs that come before "WARNING" (EMERGENCY, ALERT, CRITICAL, ERROR, WARNING) are logged.
- When you set the log-level to "NONE", that means log has been disables for the specific module, group, or global.
- Logging configuration for BDS and PUBSUB modules are not supported.

2.3. Deleting all existing log configurations

delete log

Example

```
root@rtbrick: cfg> delete log
root@rtbrick: cfg> commit
```

2.4. Configure global log level in all BD

set log <bd-name> **all level** <log-level>

Example

```
root@rtbrick: cfg> set log bd all level Warning
root@rtbrick: cfg> commit
```

2.4.1. Configure global log level in a single BD

set log bd <bd-name> **level** <log-level>

Example

```
root@rtbrick: cfg> set log bd pppoed.1 level Notice
root@rtbrick: cfg> commit
```

2.5. Configure module log level in all BD

set log <bd-name> **all module** <module-name> **level** <log-level>

Example

```
root@rtbrick: cfg> set log bd all module pppoed level Info
root@rtbrick: cfg> commit
```

2.6. Configure module log level in a single BD

set log bd <bd-name> **module** <module-name> **level** <log-level>

Example

```
root@rtbrick: cfg> set log bd pppoed.1 module pppoed level Warning
root@rtbrick: cfg> commit
```

2.7. Configure module plugin alias in all BD

set log bd all module <module-name> **plugin-alias** <plugin-alias>

Example

```
root@rtbrick: cfg> set log bd all module pppoed plugin-alias default
root@rtbrick: cfg> commit
```

Before configuring a plugin-alias here, you need to add graylog endpoints in the CTRLD start-up configuration. For more information, refer to the CTRLD configuration guide. If the configured plugin-alias name does not match any of the graylog endpoints name configured in CTRLD, then the log will be sent to the default graylog endpoint ("graylog_url") that is configured in CTRLD configuration.

/etc/rtbrick/ctrld/config.json example

```
{
"rbms_enable": true,
"rbms_host": "http://10.200.32.48",
"rbms_authorization_header": "Bearer THIS IS NOT A REAL KEY",
"rbms_heart_beat_interval": 10,
"graylog_enable": true,
"graylog_url": "http://10.200.32.49:12201/gelf",
"graylog_heart_beat_interval": 10,
"graylog_endpoints": [
{
"name": "ztp",
"url": "http://192.168.202.46:12201/gelf"
}
],
"auth_enabled": false
}
```

2.8. Configure module plugin alias in a single BD

set log bd <bd-name> **module** <module-name> **plugin-alias** <plugin-alias>

Example

```
root@rtbrick: cfg> set log bd pppoed.1 module pppoed plugin-alias pppoed
root@rtbrick: cfg> commit
```

2.9. Configure group log level in all BD

set log bd all module <module-name> **group** <group-name> **level** <log-level>

Example

```
root@rtbrick: cfg> set log bd all module pppoed group subscriber level Info
root@rtbrick: cfg> commit
```

2.10. Configure group log level in a single BD

set log bd <bd-name> **module** <module-name> **group** <group-name> **level** <log-level>

Example

```
root@rtbrick: cfg> set log bd pppoed.1 BD module pppoed group subscriber
level Debug
root@rtbrick: cfg> commit message
```

2.11. Configure group plugin alias in all BD

set log bd all module <module-name> **group** <group-name> **plugin-alias** <plugin-alias>

Example

```
root@rtbrick: cfg> set log bd all module pppoed group subscriber plugin-alias
suball
root@rtbrick: cfg> commit message
```

2.12. Configure group plugin alias in a single BD

set log bd <bd-name> **module** <module-name> **group** <group-name> **plugin-alias** <plugin-alias>

Example

```
root@rtbrick: cfg> set log bd pppoed.1 module pppoed group subscriber plugin-
alias subp
root@rtbrick: cfg> commit
```

2.13. Configure group rate limit in all BD

set log bd all module <module-name> **group** <group-name> **rate-limit** <rate-limit>

Example

```
root@rtbrick: cfg> set log bd all module pppoed group subscriber rate-limit
100
root@rtbrick: cfg> commit
```

Rate-limiting is only supported for log groups. Configuring a higher rate-limit for a whole module may cause system instability due to generation of high volume of logs.

2.14. Configure group rate limit in a single BD

set log bd <bd-name> **module** <module-name> **group** <group-name> **rate-limit** 200

Example

```
root@rtbrick: cfg> set log bd pppoed.1 module pppoed group subscriber rate-
limit 200
root@rtbrick: cfg> commit
```

2.15. Save configuration

save config <filename>

Example

```
root@rtbrick: cfg> save config log_test_config.json
```

2.16. Unconfiguring global log level in all BD

delete log bd all level <log-level>

Example

```
root@rtbrick: cfg> delete log bd all level Warning
root@rtbrick: cfg> commit
```

2.17. Unconfiguring global log level in a single BD

delete log bd <bd-name> **level** <log-level>

Example

```
root@rtbrick: cfg> delete log bd pppoed.1 level Notice
root@rtbrick: cfg> commit
```

2.18. Unconfiguring module log level in a single BD

delete log bd <bd-name> **module** <module-name> **level** <log-level>

Example

```
root@rtbrick: cfg> delete log bd pppoed.1 module pppoed level Warning
root@rtbrick: cfg> commit
```

2.19. Unconfiguring module log level in all BD

delete log bd all module <module-name> **level** <log-level>

Example

```
root@rtbrick: cfg> delete log bd all module pppoed level Info
root@rtbrick: cfg> commit
```

2.20. Unconfiguring module plugin alias in a single BD

delete log bd <bd-name> **module** <module-name> **plugin-alias** <plugin-alias>

Example

```
root@rtbrick: cfg> delete log bd pppoed.1 module pppoed plugin-alias pppoed
root@rtbrick: cfg> commit
```

2.21. Unconfiguring module plugin alias in all BD

delete log bd all module <module-name> **plugin-alias** <plugin-alias>

Example

```
root@rtbrick: cfg> delete log bd all module pppoed plugin-alias default
root@rtbrick: cfg> commit
```

2.22. Unconfiguring group log level in a single BD

delete log bd <bd-name> **module** <module-name> **group** <group-name> **level** <log-level>

Example

```
root@rtbrick: cfg> delete log bd pppoed.1 module pppoed group subscriber
level Debug
root@rtbrick: cfg> commit
```

2.23. Unconfiguring group log level in all BD

delete log bd all module <module-name> **group** <group-name> **level** <log-level>

Example

```
root@rtbrick: cfg> delete log bd all module pppoed group subscriber level
Info
root@rtbrick: cfg> commit
```

2.24. Unconfiguring group plugin alias in a single BD

delete log bd <bd-name> **module** <module-name> **group** <group-name> **plugin-alias** <plugin-alias>

Example

```
root@rtbrick: cfg> delete log bd pppoed.1 module pppoed group subscriber
plugin-alias subs
root@rtbrick: cfg> commit
```

2.25. Unconfiguring group plugin alias in all BD

delete log bd all module <module-name> **group** <group-name> **plugin-alias** <plugin-alias>

Example

```
root@rtbrick: cfg> delete log bd all module pppoed group subscriber plugin-
alias subs
root@rtbrick: cfg> commit
```

2.26. Unconfiguring group rate limit in a single BD

delete log bd <bd-name> **module** <module-name> **group** <group-name> **rate-limit** <rate-limit>

Example

```
root@rtbrick: cfg> delete log bd pppoed.1 module pppoed group subscriber
rate-limit 100
root@rtbrick: cfg> commit
```

2.27. Unconfiguring group rate limit in all BD

delete log bd all module <module-name> **group** <group-name> **rate-limit** <rate-limit>

Example

```
root@rtbrick: cfg> delete log bd all module pppoed group subscriber rate-
limit 100
root@rtbrick: cfg> commit
```

2.28. Loading configuration

load config <file-name>

Example

```
root@rtbrick: cfg> load config log_test_config.json
```

2.29. Viewing and Rendering Log Details

2.29.1. show log <bd-name> status

Displays the status of the log.

Example

```
supervisor@dev: op> show log confd status
System log files:
  Log Level: error
BDS log tables:
  Log status: Enabled
  Log level: error
  Log plugin: None
  Module log status:
    bds, Level: none, Plugin: None
    Log group status:
      Group                    Level          Plugin                   Rate limit
      generic                  none           None                     10
      object                   none           None                     10
      table                    none           None                     10
      trim                     none           None                     10
    bgp, Level: error, Plugin: None
    Log group status:
      Group                    Level          Plugin                   Rate limit
      config                   error          None                     10
      general                  error          None                     10
      generic                  error          None                     10
      instance                 error          None                     10
      interface                error          None                     10
      message                  error          None                     10
      peer                     error          None                     10
    fib, Level: error, Plugin: None
    Log group status:
      Group                    Level          Plugin                   Rate limit
      adjacency                error          None                     10
      bds                      error          None                     10
      config                   error          None                     10
      general                  error          None                     10
      generic                  error          None                     10
      instance-afi-safi        error          None                     10
      interface-events         error          None                     10
      route                    error          None                     10
    fwd_infra, Level: error, Plugin: None
```

2.29.2. show log <bd-name> status detail

```
supervisor@dev: op> show log confd status detail
System log files:
  Log Level: error
BDS log tables:
  Log status: Enabled
  Log level: error
  Log plugin: None
  Module log status:
    bds, Level: none, Plugin: None
    Log group status:
      generic, Level: none, Plugin: None, Rate limit: 10
      Log ID status:
        LOG ID                                                          Level
Plugin                  Rate limit
        BDS_ATTRIBUTE_TEMPLATE_EVENT                                    none
None                    10
        BDS_INVALID_PARAMS                                              none
None                    10
        BDS_PUBSUB_ERROR_STATUS                                         none
None                    10
        BDS_QUEUE_TABLE                                                 none
None                    10
        BDS_ROOT_EVENT                                                  none
None                    10
        BDS_TEST_LOG                                                    none
None                    10
      object, Level: none, Plugin: None, Rate limit: 10
      Log ID status:
        LOG ID                                                          Level
Plugin                  Rate limit
        BDS_INVALID_OBJECT_TEMPLATE                                     none
None                    10
        BDS_INVALID_OBJECT_TEMPLATE_ATTRIBUTE                           none
None                    10
        BDS_OBJECT_ATTRIBUTE_EVENT                                      none
None                    10
        BDS_OBJECT_EVENT                                                none
None                    10
        BDS_OBJECT_TEMPLATE_EVENTS                                      none
None                    10
        BDS_OBJECT_TEMPLATE_GET_ERROR                                   none
None                    10
        BDS_ROOT_OBJECT_EVENT                                           none
None                    10
      table, Level: none, Plugin: None, Rate limit: 10
      Log ID status:
        LOG ID                                                          Level
Plugin                  Rate limit
        BDS_TABLE_CHANGE_CALLBACK                                       none
None                    10
        BDS_TABLE_ERROR_STATUS                                          none
None                    10
        BDS_TABLE_EVENT                                                 none
None                    10
        BDS_TABLE_EVENTS                                                none
None                    10
        BDS_TABLE_SEQ_BLOCK                                             none
None                    10
        BDS_TABLE_TEMPLATE_EVENT
```

2.29.3. render log <bd-name>

This command renders all the available logs in a specified BD.

```
supervisor@leaf: op> render log confd
[  Error   ] <Tue Nov 10 19:44:31 GMT +0000 2020> Table
[/var/rtbrick/commit_registry/global.commit.registry.snap] - event Could not open file
for reading
[   Info   ] <Thu Nov 12 11:20:29 GMT +0000 2020> Commit Success
[  Debug   ] <Thu Nov 12 11:21:08 GMT +0000 2020> CLI candidate config deletion begin
[  Debug   ] <Thu Nov 12 11:21:08 GMT +0000 2020> CLI candidate config deletion ends,
status : success
[  Debug   ] <Thu Nov 12 11:21:08 GMT +0000 2020> CLI candidate config addition begin
[   Info   ] <Thu Nov 12 11:21:08 GMT +0000 2020> Advertise:true | Snapshot type:2 |
Table name:global.system.config.table | Table type:system_config_table | Deferred:false
| Interval:0 | Type:0 | Consume:false
[   Info   ] <Thu Nov 12 11:21:08 GMT +0000 2020> No keys to inherit, yang node
identifier: table-type system_config_table,  table-getter symbol name :
confd_system_config_tbl_tmpl_get , libname : libconfd.so
[  Debug   ] <Thu Nov 12 11:21:08 GMT +0000 2020> Setting attribute > Table name :
global.system.config.table, object : system_config_object, command-token-name : name,
attribute-name : configuration_name, value : rtbrick, type : string
[  Debug   ] <Thu Nov 12 11:21:08 GMT +0000 2020> BDS object found
[  Debug   ] <Thu Nov 12 11:21:08 GMT +0000 2020> Processing TARGET transaction and
replaying ADD, xml_name : system
[  Debug   ] <Thu Nov 12 11:21:08 GMT +0000 2020> Setting attribute > Table name :
global.system.config.table, object : system_config_object, command-token-name : name,
attribute-name : configuration_name, value : rtbrick, type : string
supervisor@leaf: op>
```

2.29.4. render log <bd-name> to file <file-name>

This command renders all the available logs in a specified BD to a specified file.

```
supervisor@leaf: op> render log confd to file confd_log.txt
supervisor@leaf: op>
```

2.29.5. render log <bd-name> format <summary|abstract|detailed>

The rendered output of the log (from the BD) can be formatted as follows:

- summary
- abstract
- detailed

```
supervisor@leaf: op> render log confd format abstract
Table [/var/rtbrick/commit_registry/global.commit.registry.snap] - event Could not open
file for reading
Commit Success
CLI candidate config deletion begin
CLI candidate config deletion ends, status : success
CLI candidate config addition begin
Advertise:true | Snapshot type:2 | Table name:global.system.config.table | Table
type:system_config_table | Deferred:false | Interval:0 | Type:0 | Consume:false
No keys to inherit, yang node identifier: table-type system_config_table,  table-getter
symbol name : confd_system_config_tbl_tmpl_get , libname : libconfd.so
Setting attribute > Table name : global.system.config.table, object :
system_config_object, command-token-name : name, attribute-name : configuration_name,
value : rtbrick, type : string
BDS object found
Processing TARGET transaction and replaying ADD, xml_name : system
Setting attribute > Table name : global.system.config.table, object :
system_config_object, command-token-name : name, attribute-name : configuration_name,
value : rtbrick, type : string
Table name global.system.config.table, object name system_config_object
Table name global.system.config.table, object name system_config_object, status success
Advertise:true | Snapshot type:2 | Table name:global.rtbrick.hostname.config | Table
type:global_rtbrick_hostname_tbl | Deferred:false | Interval:0 | Type:0 | Consume:false
No keys to inherit, yang node identifier: table-type global_rtbrick_hostname_tbl,
table-getter symbol name : confd_rtbrick_hostname_config_tbl_tmpl_get , libname :
libconfd.so
```

2.29.6. render log <bd-name> <*> to file <filename>

The rendered output of the log can be saved to a file by adding the keyword "to file <filename" to the end of the command.

2.29.7. render log <bd-name> filter level <log-level>

```
supervisor@leaf: op> render log confd filter level Error
[  Error   ] <Tue Nov 10 19:44:31 GMT +0000 2020> Table
[/var/rtbrick/commit_registry/global.commit.registry.snap] - event Could not open file
for reading
[  Error   ] <Tue Nov 10 19:44:48 GMT +0000 2020> Table [global.tacacs.config] Object
[name - tacacs_config_object] attribute - tacacs_server_ip not found event TACACS Server
Hostconfd Config
```

2.29.8. render log <bd-name> filter module <module-name>

The rendered output of the log can be filtered for the specified module.

```
supervisor@leaf: op> render log confd filter module secure_management
[  Error   ] <Tue Nov 10 19:44:48 GMT +0000 2020> Table [global.tacacs.config] Object
[name - tacacs_config_object] attribute - tacacs_server_ip not found event TACACS Server
Hostconfd Config
supervisor@leaf: op>
```

2.29.9. render log <bd-name> filter <*>

The rendered output of the log can be saved to a file by adding the keyword "to file <filename" to the end of the command.

2.29.10. render log <bd-name> filter <*> format <summary|abstract|detailed>

The log can be rendered in the following formats by adding "format <format-name>" to the end of the command.

- summary
- abstract
- detailed

2.29.11. render log <bd-name> filter <*> to file <filename>

Any rendered log in the specified format can be saved to a file by adding the keyword "to file <filename" to the end of the command.

2.29.12. render log <bd-name> table <log-table>

This command renders logs from a specified table from the specified BD. Every BD will have multiple log tables. By default, log will be rendered from every log table if not specified.

```
supervisor@leaf: op> render log confd table secure_management.confd.log
[  Error   ] <Tue Nov 10 19:44:48 GMT +0000 2020> Table [global.tacacs.config] Object
[name - tacacs_config_object] attribute - tacacs_server_ip not found event TACACS Server
Hostconfd Config
supervisor@leaf: op>
```

2.29.13. render log <bd-name> table <log-table> to file <filename>

Any rendered log in the specified table can be saved to a file by adding the keyword "to file <filename" to the end of the command.

2.29.14. render log <bd-name> table <log-table> format <summary|abstract|detailed>

Any log from the specified table can be rendered in the following formats by adding "format <format-name>" to the end of the command.

- summary

- abstract
- detailed

2.29.15. render log <bd-name> table <log-table> format <summary|abstract|detailed> to file <filename>

Any rendered log (from a specified table) in the specified format can be saved to a file by adding the keyword "to file <filename" to the end of the command.

2.29.16. render log <bd-name> table <log-table> filter level <log-level>

You can apply filters on all logs from a single table.

```
supervisor@leaf: op> render log confd table rtbrick-cli.confd.log filter level Info
[   Info   ] <Thu Nov 12 11:20:29 GMT +0000 2020> Commit Success
[   Info   ] <Thu Nov 12 11:21:08 GMT +0000 2020> Advertise:true | Snapshot type:2 |
Table name:global.system.config.table | Table type:system_config_table | Deferred:false
| Interval:0 | Type:0 | Consume:false
[   Info   ] <Thu Nov 12 11:21:08 GMT +0000 2020> No keys to inherit, yang node
identifier: table-type system_config_table,  table-getter symbol name :
confd_system_config_tbl_tmpl_get , libname : libconfd.so
[   Info   ] <Thu Nov 12 11:21:08 GMT +0000 2020> Advertise:true | Snapshot type:2 |
Table name:global.rtbrick.hostname.config | Table type:global_rtbrick_hostname_tbl |
Deferred:false | Interval:0 | Type:0 | Consume:false
[   Info   ] <Thu Nov 12 11:21:08 GMT +0000 2020> No keys to inherit, yang node
identifier: table-type global_rtbrick_hostname_tbl,  table-getter symbol name :
confd_rtbrick_hostname_config_tbl_tmpl_get , libname : libconfd.so
[   Info   ] <Thu Nov 12 11:21:08 GMT +0000 2020> Commit Success
```

2.29.17. render log <bd-name> table <log-table> filter module <module-name>

```
supervisor@leaf: op> render log confd table rtbrick-cli.confd.log filter module rtbrick-
cli
[   Info   ] <Thu Nov 12 11:20:29 GMT +0000 2020> Commit Success
[  Debug   ] <Thu Nov 12 11:21:08 GMT +0000 2020> CLI candidate config deletion begin
[  Debug   ] <Thu Nov 12 11:21:08 GMT +0000 2020> CLI candidate config deletion ends,
status : success
[  Debug   ] <Thu Nov 12 11:21:08 GMT +0000 2020> CLI candidate config addition begin
[   Info   ] <Thu Nov 12 11:21:08 GMT +0000 2020> Advertise:true | Snapshot type:2 |
Table name:global.system.config.table | Table type:system_config_table | Deferred:false
| Interval:0 | Type:0 | Consume:false
```

2.29.18. render log <bd-name> table <log-table> filter <*> to file <filename>

The log rendered output from the "render log <bd-name> table <log-table> filter <*>" command can be saved to a file by adding the keyword "to file <filename" to

the end of the command.

2.29.19. render log <bd-name> table <log-table> filter <*> format <summary|abstract|detailed>

The log rendered output from the "render log <bd-name> table <log-table> filter <*>" command can be displayed displayed in the following formats by adding "format <format-name>" to the end of the command.

- summary
- abstract
- detailed

2.29.20. render log <bd-name> table <log-table> filter <*> format <summary|abstract|detailed> to file <filename>

Any rendered log (from a specified table) in the specified format can be saved to a file by adding the keyword "to file <filename" to the end of the command.