
# RBFS Logging Guide

**Version 20.5.1-rc0, 25 May 2020**

| 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. |
| global log | Aggregation of local log table. This is present in LOGD. |
| 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

- Logging is disabled 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 module has been disabled by default so even if you set the global log level, you will not see any BDS module logs.
  - To get BDS module log, you need to set the log level for BDS module via CLI using the set log bd <bd-name> module bds logmap all level Error command.
- 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
- Following log level are present in the system; anything above Warning indicates that you need to logging with caution as scaled system might get into an unstable situation.
  - None
  - Error

- Warning
- Info
- Detail
- Extensive

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 |

## 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/libbds/logs
```

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

```
ubuntu@bgp_rt1:/usr/share/rtbrick/libbds/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_DETAIL_CONNECT_COLLUSION", you can do the following:

```
ubuntu@bgp_rt1:~$ cd /usr/share/rtbrick/libbds/logs
ubuntu@bgp_rt1:/usr/share/rtbrick/libbds/logs$ grep -ir -B1 -A1
BGP_DETAIL_CONNECT_COLLUSION
bgp/bgp_details.json-            "log_id"     : 16042,
bgp/bgp_details.json:            "log_name"   :
"BGP_DETAIL_CONNECT_COLLUSION",
bgp/bgp_details.json-            "log_string" : "Socket connect collusion.
Instance:%s, peer:%s, src:%s, closed:%s \n",
ubuntu@bgp_rt1:/usr/share/rtbrick/libbds/logs$
```

Here, log_string displays a human-readable interpretation of the log.

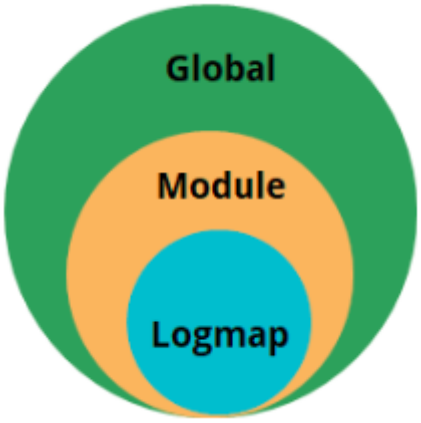
# 2. Configuring Logs

## 2.1. Logging Hierarchy

Logging can be configured at the following three hierarchy:

- Global
- Module
- Logmap

The figure below shows the preference of these levels:




That means, a configuration at log map level will take preference over the module level which in turn would have taken preference over the global level. At the same time, if you configure log at the global level, it will be inherited by the module and logmap levels unless configured specifically under the respective levels.

## 2.2. Global Configuration

- Log level – default: 'none'

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

- None
- Error
- Warning
- Info
- Detail
- Extensive

All log levels lesser than the log level specified are logged. For example, if the

specified log level is "Detail", then all logs that come before "Details" (Error, Warning, Info, and Details) are logged.

## 2.2.1. set system rtbrick log level <log-level>

This command sets log level for system logging.

### Syntax

```
set system rtbrick log level <log-level>
```

### Example

```
root@rtb:confd> set system rtbrick log level Error
[ ]
root@rtb:confd>
```

## 2.2.2. set log bd <bd-name> module <module-name> logmap <map-name> level <log-level>

This command sets BDS log level for BD(s), module(s), and logmap(s).

### Syntax

```
set log bd <bd-name> module <module-name> logmap <map-name> level <log-level>
```

### Example for setting global log level to error for all BD in the system :

```
root@rtb:confd> set log bd all module all logmap all level Error
[ ]
root@rtb:confd>
```

### Example for setting global log level to error for a specific BD in the system :

```
root@rtb:confd> set log bd bgpd module all logmap all level Error
[ ]
root@rtb:confd>
```

### Example for setting global log level to error for a specific module :

```
root@rtb:confd> set log bd all module bds logmap all level Error
[ ]
root@rtb:confd>
```

**Example for setting global log level to error for a specific log map in a module**

```
root@rtb:confd> set log bd all module bds logmap BDS_INVALID_OBJECT_TEMPLATE
level Error
[ ]
root@rtb:confd>
```

## 2.2.3. set log bd <bd-name> module <module-name> logmap <map-name> plugin-type <plugin-type>

This command allows you to set plugin type for BDS logging for a module or logmap(s).

**Syntax**

```
set log bd <bd-name> module <module-name> logmap <map-name> plugin-type
<plugin-type>
```

For examples, see the previous section.

## 2.2.4. set log bd <bd-name> module <module-name> logmap <map-name> rate-limit <rate-limit>

This command allows you to set rate limit of BDS logging for a log map.

**Example**

```
root@rtb:confd> set log bd all module bds logmap BDS_TABLE_EVENT rate-limit
10
[ ]
root@rtb:confd>
```

In this example, the rate-limit 10 indicates that in one second a maximum of 10 entries will be logged for the BDS_TABLE_EVENT, and all the extra entries within that timeframe will be dropped.

Rate-limiting is only supported for log maps.

### 2.2.5. set log bd <bd-name> plugin-server <plugin-server-type> server-ip <server-ip> server-port <server-port>

This command allows you to set plug-in server details logging for all or a single BD in the system.

**Example**

```
root@rtb:confd> set log bd all plugin-server Gray-log server-ip 12.1.1.1
server-port 1110
[ ]
root@rtb:confd>
```

### 2.2.6. set log bd <bd-name> plugin-server <plugin-server-type> server-ip <server-ip> server-port <server-port> server-ep <server-ep>

This command allows you to set plug-in server details and server end-point for logging for all or a single BD in the system.

Currently RBFS supports gray-log and SNMP as as plug-in server.

**Example**

```
root@rtb:confd> set log bd all plugin-server Gray-log server-ip 12.1.1.1
server-port 1110 server-ep /gelf
[ ]
root@rtb:confd>
```

## 2.3. Deleting Log Configurations

The following commands allow you to delete logging configurations.

To delete logging configurations for a BD(s) and plugin-server-type:

```
delete log bd <bd-name> plugin-server <plugin-server-type>
```

To delete logging configurations for a BD(s), module(s) or logmap(s):

```
delete log bd <bd-name> module <module-name> logmap <map-name>
```

To delete plugin type configurations for a BD(s), module(s) or logmap(s):

```
delete  log bd <bd-name> module <module-name> logmap <map-name> plugin-type
```

To delete rate-limit configurations for a BD(s), module or logmap:

```
delete  log bd <bd-name> module <module-name> logmap <map-name> rate-limit
```

# 2.4. Viewing and Rendering Log Details

## 2.4.1. show log status

Displays the status of the log.

### Example

```
root@rtb:confd> show log status
System Log Status:
  Logging Level: Error

Bds Log Status:
  Logging Enabled: true
  Logging Level: Error

  Module Log Status:
     bds, Level: Error, Plugin: None
     Log ID Status:
       BDS_ATTRIBUTE_TEMPLATE_EVENT, Level: None, Plugin: None, Rate-limit: 0
       BDS_INVALID_OBJECT_TEMPLATE, Level: Error, Plugin: None, Rate-limit: 0
       BDS_INVALID_OBJECT_TEMPLATE_ATTRIBUTE, Level: None, Plugin: None, Rate-limit: 0
       BDS_INVALID_PARAMS, Level: None, Plugin: None, Rate-limit: 0
       BDS_OBJECT_ATTRIBUTE_EVENT, Level: None, Plugin: None, Rate-limit: 0
       BDS_OBJECT_EVENT, Level: None, Plugin: None, Rate-limit: 0
       BDS_OBJECT_TEMPLATE_EVENTS, Level: None, Plugin: None, Rate-limit: 0
       BDS_OBJECT_TEMPLATE_GET_ERROR, Level: None, Plugin: None, Rate-limit: 0
       BDS_PUBSUB_ERROR_STATUS, Level: None, Plugin: None, Rate-limit: 0
       BDS_QUEUE_TABLE, Level: None, Plugin: None, Rate-limit: 0
       BDS_ROOT_EVENT, Level: None, Plugin: None, Rate-limit: 0
       BDS_ROOT_OBJECT_EVENT, Level: None, Plugin: None, Rate-limit: 0
       BDS_TABLE_CHANGE_CALLBACK, Level: None, Plugin: None, Rate-limit: 0
       BDS_TABLE_ERROR_STATUS, Level: None, Plugin: None, Rate-limit: 0
       BDS_TABLE_EVENT, Level: None, Plugin: None, Rate-limit: 10
       BDS_TABLE_EVENTS, Level: None, Plugin: None, Rate-limit: 0
       BDS_TABLE_SEQ_BLOCK, Level: None, Plugin: None, Rate-limit: 0
       BDS_TABLE_TEMPLATE_EVENT, Level: None, Plugin: None, Rate-limit: 0
       BDS_TEST_LOG, Level: None, Plugin: None, Rate-limit: 0
     pubsub, Level: Error, Plugin: None
     Log ID Status:
       PUBSUB_CSN_EVENT, Level: None, Plugin: None, Rate-limit: 0
       PUBSUB_INVALID_MESSAGE, Level: None, Plugin: None, Rate-limit: 0
       PUBSUB_INVALID_PARAMS, Level: None, Plugin: None, Rate-limit: 0
       PUBSUB_LINE_EVENT, Level: None, Plugin: None, Rate-limit: 0
       PUBSUB_OBJECT_EVENT, Level: None, Plugin: None, Rate-limit: 0
       PUBSUB_REGEX_COMPARE_EVENT, Level: None, Plugin: None, Rate-limit: 0
       PUBSUB_REGEX_EVENT, Level: None, Plugin: None, Rate-limit: 0
       PUBSUB_REGEX_SCAN_EVENT, Level: None, Plugin: None, Rate-limit: 0
```

```
       PUBSUB_TABLE_EVENT, Level: None, Plugin: None, Rate-limit: 0
       PUBSUB_TABLE_LINE_EVENT, Level: None, Plugin: None, Rate-limit: 0
       PUBSUB_TABLE_REGEX_EVENT, Level: None, Plugin: None, Rate-limit: 0
       PUBSUB_TABLE_SEQ_EVENT, Level: None, Plugin: None, Rate-limit: 0
     snapshot, Level: None, Plugin: None
     Log ID Status:
       SNAPSHOT_INVALID_PARAMS, Level: None, Plugin: None, Rate-limit: 0
       SNAPSHOT_TABLE_EVENT, Level: None, Plugin: None, Rate-limit: 0
     time-series, Level: None, Plugin: None
     Log ID Status:
       TIME_SERIES_ALERT_CONFIG_TO_HOSTCONFD_SKIPPED, Level: None, Plugin: None, Rate-
limit: 0
       TIME_SERIES_ALERT_INCOMPLETE_CONFIG, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_ALERT_MANAGER_CONFIG_COMPLETE, Level: None, Plugin: None, Rate-limit:
0
       TIME_SERIES_ALERT_MANAGER_CONFIG_EVENT, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_ALERT_MANAGER_CONFIG_INCOMPLETE, Level: None, Plugin: None, Rate-
limit: 0
       TIME_SERIES_ALERT_MANAGER_FILE_EVENT, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_ATTRIBUTE_INCOMPLETE_CONFIG, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_ATTRIBUTE_NOT_DEFINED, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_ATTRIBUTE_TYPE_INVALID, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_BUFFER_NO_SPACE, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_CUSTOM_PRINTER_HIT, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_FILTER_ALLOWED, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_FILTER_ATTRIBUTE_INCOMPLETE_CONFIG, Level: None, Plugin: None, Rate-
limit: 0
       TIME_SERIES_FILTER_DENIED, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_FILTER_EVENT, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_FILTER_LABEL_INCOMPLETE_CONFIG, Level: None, Plugin: None, Rate-
limit: 0
       TIME_SERIES_FILTER_MATCH_ATTRIBUTE_TYPE_INVALID, Level: None, Plugin: None, Rate-
limit: 0
       TIME_SERIES_FILTER_METRIC_INCOMPLETE_CONFIG, Level: None, Plugin: None, Rate-
limit: 0
       TIME_SERIES_INDEX_NAME_INVALID, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_LABEL_ATTRIBUTE_TYPE_INVALID, Level: None, Plugin: None, Rate-limit:
0
       TIME_SERIES_LABEL_INCOMPLETE_CONFIG, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_LABEL_KEY_INVALID_FORMAT, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_LABEL_KEY_SAME_AS_DEFAULT, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_LABEL_NOT_DEFINED, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_LABEL_TYPE_INVALID, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_LABEL_VALUE_INVALID, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_LABEL_VALUE_NOT_DEFINED, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_METRIC_INCOMPLETE_CONFIG, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_METRIC_TYPE_INVALID, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_NO_ATTRIBUTE, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_PROCESS_NOT_FOUND, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_PROMETHEUS_RESTART_EVENT, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_REGEX_EVENT, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_SUBSCRIBED_TABLE, Level: None, Plugin: None, Rate-limit: 0
       TIME_SERIES_TABLE_NOT_FOUND, Level: None, Plugin: None, Rate-limit: 0
[ ]
root@rtb:confd>
```

## 2.4.2. show log-plugin server stats all

```
[ ]
root@rtb:confd> show log-plugin server stats all
 Connection Stats : Gray Log
  Server Ip                  : 12.1.1.1
  Server Port                : 1110
  Server EP                  : /gelf
  Server Connected           : Yes
  Server Connect Status      : Down
  Total Send Count           : 0

  Rtbrick Host Name          : rtb
  Rtbrick Pod Name           : rtbrick-pod
 Connection Stats : SNMP Server
  Server Ip                  : 0.0.0.0
  Server Port                : 0
  Server Connected           : No
  Server Connect Status      : Down
  Total Send Count           : 0

  Rtbrick Host Name          : rtb
  Rtbrick Pod Name           : rtbrick-pod
[ ]
root@rtb:confd>
```

## 2.4.3. show log-plugin server stats plugin <plugin>

```
root@rtb:confd> show log-plugin server stats plugin gray-log
 Connection Stats : Gray Log
  Server Ip                  : 12.1.1.1
  Server Port                : 1110
  Server EP                  : /gelf
  Server Connected           : Yes
  Server Connect Status      : Down
  Total Send Count           : 0

  Rtbrick Host Name          : rtb
  Rtbrick Pod Name           : rtbrick-pod
[ ]
root@rtb:confd>
```

## 2.4.4. render datastore log <table-name> to stdout <format>

```
root@rtb:logd> render datastore log bds.logd.log to stdout detailed
[  Error  ] <2020-03-
06T10:56:45.645532+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bd.log.config.table] Table Type [bd_log_config_table] event - Add
failed - Skip index set by user
[  Error  ] <2020-03-
06T10:56:45.648885+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bds.pub.all] Table Type [pub.all.table] event - Add failed - Skip
index set by user
[  Error  ] <2020-03-06T10:56:45.648998+0530>[bds]{pubsub_ipc_send_table:
380}-> Table [global.bds.sub.all.logd] Table Type [sub.all] event - Tx for
uuid is not up
[  Error  ] <2020-03-06T10:56:45.649106+0530>[bds]{pubsub_ipc_send_table:
380}-> Table [global.bds.sub.all.logd] Table Type [sub.all] event - Tx for
uuid is not up
[  Error  ] <2020-03-
06T10:56:45.650522+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bds.pub.all.logd] Table Type [pub.table] event - Add failed - Skip
index set by user
[  Error  ] <2020-03-
06T10:56:45.654264+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bds.sub.all.logd] Table Type [sub.all] event - Add failed - Skip
index set by user
[  Error  ] <2020-03-06T10:56:45.654411+0530>[bds]{pubsub_ipc_send_table:
380}-> Table [global.bds.sub.all.logd] Table Type [sub.all] event - Tx for
uuid is not up
[  Error  ] <2020-03-06T10:56:45.654464+0530>[bds]{pubsub_ipc_send_table:
380}-> Table [global.bds.sub.all.logd] Table Type [sub.all] event - Tx for
uuid is not up
[  Error  ] <2020-03-
06T10:56:45.655838+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bds.pub.all.logd] Table Type [pub.table] event - Add failed - Skip
index set by user
[  Error  ] <2020-03-06T10:56:45.655986+0530>[pubsub]{pubsub_subscribe_table:
696}-> Table [bds.logd.log] to/from [None] event - Subscribing own table
[  Error  ] <2020-03-06T10:56:45.670388+0530>[bds]{pubsub_ipc_send_table:
380}-> Table [global.bds.sub.all.logd] Table Type [sub.all] event - Tx for
uuid is not up
 ...
 ...
 ...
```

## 2.4.5. render datastore log <table-name> to file <file-name> <format>

```
root@rtb:logd> render datastore log bds.p1.log to file bds_log.log detailed
[ ]
root@rtb:logd>
```

In this example, the log details are written to a file named "bds_log.log" (located at /var/log/rtbrick/bds_log.log).

## 2.4.6. Filters supported by Logging

All the filters explained below are also supported for rendering to files.

### 2.4.6.1. render datastore log <table-name> to stdout <format> filter level <log-level>

This command allows you to filter logs based on log level.

The following example shows how to filter logs for the log level "**Warning**".

```
root@rtb:logd> render datastore log global.log.table to stdout detailed
filter level Warning
[ Warning ] <2020-03-
06T11:26:47.215953+0530>[bds]{bds_hierarchy_remove_from_bplus:1672}-> Root
gc-index object-seq [1] [Ref 4 bitmap 8] event -  Object is not in tree
[ Warning ] <2020-03-
06T11:26:47.230902+0530>[bds]{bds_hierarchy_remove_from_bplus:1672}-> Root
gc-index object-seq [36] [Ref 4 bitmap 8] event -  Object is not in tree
[ Warning ] <2020-03-
06T11:26:47.243486+0530>[bds]{bds_hierarchy_remove_from_bplus:1672}-> Root
gc-index object-seq [530] [Ref 3 bitmap 0] event -  Object is not in tree
[ Warning ] <2020-03-
06T11:26:47.315598+0530>[bds]{bds_table_delete_all_index:1891}-> Table
[global.bd.log.config.table.delta.102] - All objects are getting deleted
[ Warning ] <2020-03-
06T11:26:47.315740+0530>[bds]{bds_deferred_object_plugins: 465}-> Table
[global.bd.log.config.table.delta.102] Object-seq [3] [Ref:3 Consume:0
Pubmap:0 Objflood:0] event - Getting deleted
[ Warning ] <2020-03-
06T11:26:47.315794+0530>[bds]{bds_table_delete_all_index:1976}-> Table
[global.bd.log.config.table.delta.102] - All objects are deleted
 ...
 ...
 ...
```

### 2.4.6.2. render datastore log <table-name> to stdout <format> filter module <module-name>

This command allows you to filter logs based on module.

The following example shows how to filter logs for the module "**pubsub**".

```
root@rtb:logd> render datastore log bds.p1.log to stdout detailed filter
module pubsub
[  Info   ] <2020-03-06T11:26:47.231369+0530>[pubsub]{pubsub_ipc_rcv_table:
636}-> Table [global.bd.log.config.table] to/from [confd] event - Process
table
[  Info   ] <2020-03-
06T11:26:47.237297+0530>[pubsub]{pubsub_retrans_csn_expiry: 569}-> Table
[global.bd.log.config.table] to/from [confd] Seq [2-0] interval [0] event -
CSN send start
[  Info   ] <2020-03-
06T11:26:47.237358+0530>[pubsub]{pubsub_retrans_csn_expiry: 798}-> Table
[global.bd.log.config.table] to/from [confd] Seq [0-8] interval [0] event -
CSN send end
[  Info   ] <2020-03-
06T11:27:01.394435+0530>[pubsub]{pubsub_retrans_csn_expiry: 569}-> Table
[global.bd.log.config.table] to/from [confd] Seq [8-0] interval [0] event -
CSN send start
[  Info   ] <2020-03-
06T11:27:01.394538+0530>[pubsub]{pubsub_retrans_csn_expiry: 798}-> Table
[global.bd.log.config.table] to/from [confd] Seq [0-9] interval [0] event -
CSN send end
[  Info   ] <2020-03-
06T11:27:14.457574+0530>[pubsub]{pubsub_retrans_csn_expiry: 569}-> Table
[global.bd.log.server.config.table] to/from [confd] Seq [2-0] interval [0]
event - CSN send start
[  Info   ] <2020-03-
06T11:27:14.457809+0530>[pubsub]{pubsub_retrans_csn_expiry: 798}-> Table
[global.bd.log.server.config.table] to/from [confd] Seq [0-2] interval [0]
event - CSN send end
[  Info   ] <2020-03-
06T11:27:51.997378+0530>[pubsub]{pubsub_retrans_rcv_csn:1526}-> Table
[global.startup.status.p1] to/from [etcd] event - CSN full table rcvd
 ...
 ...
 ...
```

### 2.4.6.3. render datastore log <table-name> to stdout <format> filter module-regex <regex-string>

This command allows you to filter logs based on regular expression for module(s).

The following example shows how to filter logs for the module "**bd***".

```
root@rtb:logd> render datastore log bds.p1.log to stdout detailed filter
module-regex bd*
[  Error  ] <2020-03-
06T10:56:45.636020+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bd.log.config.table] Table Type [bd_log_config_table] event - Add
failed - Skip index set by user
[  Error  ] <2020-03-
06T10:56:45.641012+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bds.pub.all.p1] Table Type [pub.table] event - Add failed - Skip
index set by user
[  Error  ] <2020-03-
06T10:56:45.659555+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bds.pub.all.p1] Table Type [pub.table] event - Add failed - Skip
index set by user
[  Error  ] <2020-03-
06T10:56:45.659638+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bds.pub.all] Table Type [pub.all.table] event - Add failed - Skip
index set by user
[  Error  ] <2020-03-
06T10:56:45.664751+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bds.pub.all] Table Type [pub.all.table] event - Add failed - Skip
index set by user
[  Info   ] <2020-03-
06T11:27:51.997378+0530>[pubsub]{pubsub_retrans_rcv_csn:1526}-> Table
[global.startup.status.p1] to/from [etcd] event - CSN full table rcvd
[  Info   ] <2020-03-
06T11:27:51.997464+0530>[pubsub]{pubsub_retrans_rcv_csn:1647}-> Table
[global.startup.status.p1] to/from [etcd] event - CSN process completed
[  Info   ] <2020-03-
06T11:28:09.723773+0530>[pubsub]{pubsub_retrans_rcv_csn:1526}-> Table
[global.bds.sub.all.p1] to/from [etcd] event - CSN full table rcvd
[  Info   ] <2020-03-
06T11:28:09.723875+0530>[pubsub]{pubsub_retrans_rcv_csn:1647}-> Table
[global.bds.sub.all.p1] to/from [etcd] event - CSN process completed
 ...
 ...
 ...
```

### 2.4.6.4. render datastore log <table-name> to stdout <format> filter module <module-name> logmap <map-name>

This command allows you to filter logs for a log map.

The following example shows how to filter logs for the logmap "**BDS_TABLE_EVENT**".

```
root@rtb:logd> render datastore log global.log.table to stdout detailed
filter module bds logmap BDS_TABLE_EVENT
[ Warning ] <2020-03-
06T11:26:47.315598+0530>[bds]{bds_table_delete_all_index:1891}-> Table
[global.bd.log.config.table.delta.102] - All objects are getting deleted
[ Warning ] <2020-03-
06T11:26:47.315794+0530>[bds]{bds_table_delete_all_index:1976}-> Table
[global.bd.log.config.table.delta.102] - All objects are deleted
[ Warning ] <2020-03-06T11:26:47.315857+0530>[bds]{bds_table_free :2391}->
Table [global.bd.log.config.table.delta.102] - Table is getting freed
[ Warning ] <2020-03-
06T11:26:47.316195+0530>[bds]{bds_table_delete_all_index:1891}-> Table
[global.bd.log.config.table.interm.102] - All objects are getting deleted
[ Warning ] <2020-03-
06T11:26:47.316800+0530>[bds]{bds_table_delete_all_index:1976}-> Table
[global.bd.log.config.table.interm.102] - All objects are deleted
```

### 2.4.6.5. render datastore log <table-name> to stdout <format> filter module <module-name> logmap-regex <regex-string>

This command allows you to filter logs based on regular expression for a logmap(s).

The following example shows how to filter logs for the logmap "**BDS_TABLE_***".

```
[ ]
root@ROHIT:logd> render datastore log global.log.table to stdout detailed
filter module bds logmap-regex BDS_TABLE_*
[  Error  ] <2020-03-
06T10:56:45.489867+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bds.pub.all.confd] Table Type [pub.table] event - Add failed - Skip
index set by user
[  Error  ] <2020-03-
06T10:56:45.490261+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bds.sub.all.confd] Table Type [sub.all] event - Add failed - Skip
index set by user
[  Error  ] <2020-03-
06T10:56:45.559413+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bds.sub.all.confd] Table Type [sub.all] event - Add failed - Skip
index set by user
[  Error  ] <2020-03-
06T10:56:45.569778+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.bds.pub.all.confd] Table Type [pub.table] event - Add failed - Skip
index set by user
[  Error  ] <2020-03-
06T10:56:45.569808+0530>[bds]{bds_create_object_index_map:3982}-> Table
[global.startup.status.confd] Table Type [bds_startup_stat_table] event - Add
failed - Skip index set by user
```