



RBFS Logging Guide

Version 20.7.1, 15 July 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

©Copyright 2020 RtBrick, Inc. All rights reserved. The information contained herein is subject to change without notice. The trademarks, logos and service marks ("Marks") displayed in this documentation are the property of RtBrick in the United States and other countries. Use of the Marks are subject to RtBrick's Term of Use Policy, available at <https://www.rtbrick.com/privacy>. Use of marks belonging to other parties is for informational purposes only.

Table of Contents

1. RBFS Log Files	3
1.1. Guidelines	3
1.2. Understanding the Log Maps	4
1.3. Understanding the Log Groups	5
2. Configuring Logs	6
2.1. Logging Hierarchy	6
2.2. Global Configuration	6
2.2.1. set log bd <bd-name> module <module-name> log-group <group-name> level <log-level>	7
2.2.2. set log bd <bd-name> module <module-name> log-group <group-name> plugin-alias <plugin-alias>	8
2.2.3. set log bd <bd-name> module <module-name> log-group <group-name> rate-limit <rate-limit>	8
2.3. Viewing and Rendering Log Details	9
2.3.1. show log status	9
2.3.2. show log status detail	9
2.3.3. render datastore log <table-name> to stdout <format>	12
2.3.4. render datastore log <table-name> to file <file-name> <format>	13
2.3.5. Filters supported by Logging	14
2.3.5.1. render datastore log <table-name> to stdout <format> filter level <log-level>	14
2.3.5.2. render datastore log <table-name> to stdout <format> filter module <module-name>	14
2.3.5.3. render datastore log <table-name> to stdout <format> filter module-regex <regex-string>	15
2.3.5.4. render datastore log <table-name> to stdout <format> filter module <module-name> logmap <map-name>	16
2.3.5.5. render datastore log <table-name> to stdout <format> filter module <module-name> logmap-regex <regex-string>	17

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 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 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
- 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



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	log_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/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.

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.

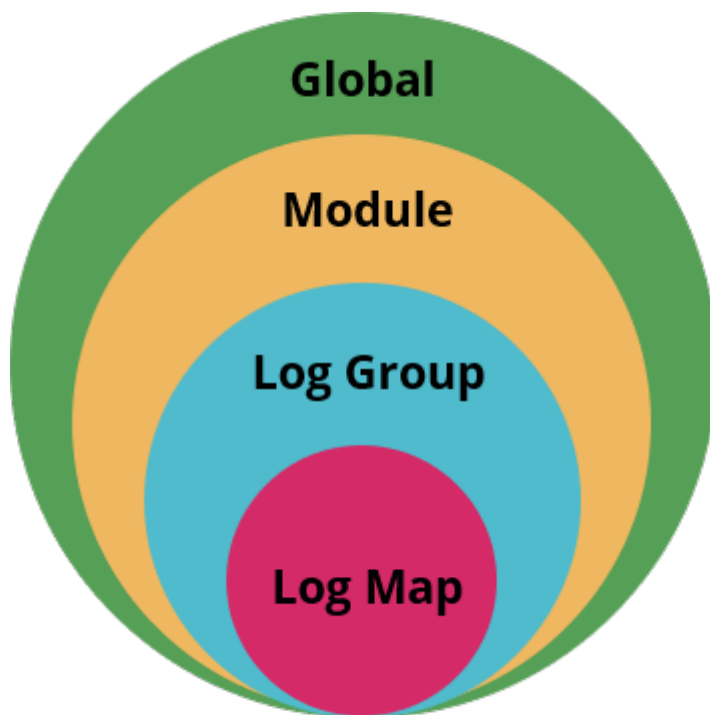
2. Configuring Logs

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

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

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

Syntax

```
set log bd <bd-name> module <module-name> log-group <group-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 log-group 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 log-group 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 log-group all level Error
[ ]
root@rtb:confd>
```

Example for setting log level to error for a specific log group in a module


```

root@dev:confd> set log bd pppoed.1 module pppoed log-group DHCPv6 level
Error
[ ]
root@dev:confd> set log bd pppoed.1 module pppoed log-group
  <group-name>          log group name
  all                   <group-name>
  Config                <group-name>
  DHCPv6                <group-name>
  General               <group-name>
  generic               <group-name>
  PPP                   <group-name>
  PPPoE                 <group-name>
  Subscriber            <group-name>
[ ]

```

2.2.2. set log bd <bd-name> module <module-name> log-group <group-name> plugin-alias <plugin-alias>

This command allows you to set plugin alias for BDS logging for a module or log group(s).

Syntax

```

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

```

For examples, see the previous section.

2.2.3. set log bd <bd-name> module <module-name> log-group <group-name> rate-limit <rate-limit>

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

Example

```

root@rtb:confd> set log bd all module bds log-group 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.

2.3. Viewing and Rendering Log Details

2.3.1. show log status

Displays the status of the log.

Example

```
root@rtbrick:confd> show log status
System Log Status:
  Logging Level: Emergency

Bds Log Status:
  Logging Enabled: true
  Logging Level: None

Module Log Status:
  bds, Level: Debug, Plugin-alias: default
    Log group Status:
      generic, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
      object, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
      table, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
      trim, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
    12tpd, Level: NotConfigured, Plugin-alias: None
    Log group Status:
      General, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
      Session, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
      Subscriber, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
      Tunnel, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
    ngaccess_subscriberd, Level: NotConfigured, Plugin-alias: None
    Log group Status:
      BDS, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
      Config, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
      General, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
      Subscriber, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
      generic, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
  [ ]
```

2.3.2. show log status detail

```
root@rtbrick:confd> show log status detail
System Log Status:
  Logging Level: Emergency

Bds Log Status:
  Logging Enabled: true
  Logging Level: None

Module Log Status:
  bds, Level: Debug, Plugin-alias: default
    Log group Status:
      generic, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
    Log ID Status:
```

```

BDS_ATTRIBUTE_TEMPLATE_EVENT, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
BDS_INVALID_PARAMS, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
BDS_PUBSUB_ERROR_STATUS, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
BDS_QUEUE_TABLE, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
BDS_ROOT_EVENT, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
BDS_TEST_LOG, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
object, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
Log ID Status:
BDS_INVALID_OBJECT_TEMPLATE, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
BDS_INVALID_OBJECT_TEMPLATE_ATTRIBUTE, Level: NotConfigured, Plugin-alias:
None, Rate-limit: 10
BDS_OBJECT_ATTRIBUTE_EVENT, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
BDS_OBJECT_EVENT, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
BDS_OBJECT_TEMPLATE_EVENTS, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
BDS_OBJECT_TEMPLATE_GET_ERROR, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
BDS_ROOT_OBJECT_EVENT, Level: NotConfigured, Plugin-alias: None, Rate-limit:
10
table, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
Log ID Status:
BDS_TABLE_CHANGE_CALLBACK, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
BDS_TABLE_ERROR_STATUS, Level: NotConfigured, Plugin-alias: None, Rate-limit:
10
BDS_TABLE_EVENT, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
BDS_TABLE_EVENTS, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
BDS_TABLE_SEQ_BLOCK, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
BDS_TABLE_TEMPLATE_EVENT, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
trim, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
Log ID Status:
BDS_TRIM_COUNT, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
BDS_TRIM_ENTRY_ADD_FAILED, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
BDS_TRIM_INFRA_TABLE_ADD_FAILED, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
BDS_TRIM_NOT_SUPPORTED, Level: NotConfigured, Plugin-alias: None, Rate-limit:
10
BDS_TRIM_OBJECT_DELETE_FAILED, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
BDS_TRIM_SUBSCRIBED_TABLE, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
BDS_TRIM_TABLE_NOT_PRESENT, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
l2tpd, Level: NotConfigured, Plugin-alias: None
Log group Status:
General, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
Log ID Status:
L2TP_ERROR, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
L2TP_INVALID_PACKET_RECEIVED_WARNING, Level: NotConfigured, Plugin-alias:
None, Rate-limit: 10
Session, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
Log ID Status:
L2TP_SESSION_RCV_INFO, Level: NotConfigured, Plugin-alias: None, Rate-limit:
10
Subscriber, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
Log ID Status:
L2TP_NO_POOL_WARNING, Level: NotConfigured, Plugin-alias: None, Rate-limit:
10

```

```

L2TP_POOL_WARNING, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
L2TP_RADIUS_TUNNEL_WARNING, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
L2TP_SESSION_CREATING_INFO, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
L2TP_SESSION_ERROR, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
L2TP_SESSION_FSM_STATE_CHANGE_INFO, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
L2TP_SESSION_REQUEST_ADD_INFO, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
L2TP_TUNNEL_CREDENTIAL_WARNING, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
Tunnel, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
Log ID Status:
L2TP_AVP_DECODE_ERROR, Level: NotConfigured, Plugin-alias: None, Rate-limit:
10
L2TP_AVP_ENCODE_ERROR, Level: NotConfigured, Plugin-alias: None, Rate-limit:
10
L2TP_DROP_PACKET_WARNING, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
L2TP_MBIT_UNKNOWN_ERROR, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
L2TP_PACKET_ENCODE_ERROR, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
L2TP_TUNNEL_AUTH_ERROR, Level: NotConfigured, Plugin-alias: None, Rate-limit:
10
L2TP_TUNNEL_AVP_MH_MISMATCH_ERROR, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
L2TP_TUNNEL_DECODE_ERROR, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
L2TP_TUNNEL_DELETED_INFO, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
L2TP_TUNNEL_FSM_STATE_CHANGE_INFO, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
L2TP_TUNNEL_MAX_RETRANSMISSION_ERROR, Level: NotConfigured, Plugin-alias:
None, Rate-limit: 10
L2TP_TUNNEL_MESSAGE_WARNING, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
L2TP_TUNNEL_PACKET_TRY_SENDING_INFO, Level: NotConfigured, Plugin-alias:
None, Rate-limit: 10
L2TP_TUNNEL_RECEIVE_ERROR, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
L2TP_TUNNEL_RECEIVE_INFO, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
L2TP_TUNNEL_REQUEST_ADD_INFO, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
L2TP_TUNNEL_SEND_INFO, Level: NotConfigured, Plugin-alias: None, Rate-limit:
10
L2TP_TX_QUEUE_ADDED_INFO, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
L2TP_TX_QUEUE_ADD_ERROR, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
ngaccess_subscriberd, Level: NotConfigured, Plugin-alias: None
Log group Status:
BDS, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
Log ID Status:
SUBSMGMT_BDS_OBJECT_ADD_ERROR, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
SUBSMGMT_BDS_OBJECT_DEL_ERROR, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
SUBSMGMT_BDS_TABLE_CREATED, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
SUBSMGMT_BDS_TABLE_CREATE_ERROR, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10

```

```

SUBSMGMT_BDS_TABLE_GET_ERROR, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
SUBSMGMT_BDS_TEMPLATE_GET_ERROR, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
SUBSMGMT_BDS_TEMPLATE_POPULATE_ERROR, Level: NotConfigured, Plugin-alias:
None, Rate-limit: 10
Config, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
Log ID Status:
SUBSMGMT_AAA_PRF_SUBSCRIBER_WARNING, Level: NotConfigured, Plugin-alias:
None, Rate-limit: 10
SUBSMGMT_RADIUS_PRF_MISSING_IN_AAA_WARNING, Level: NotConfigured, Plugin-
alias: None, Rate-limit: 10
SUBSMGMT_SERVICE_PROFILE_GET_FAILED_WARNING, Level: NotConfigured, Plugin-
alias: None, Rate-limit: 10
General, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
Log ID Status:
SUBSMGMT_LOG_ERROR, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
SUBSMGMT_LOG_INFO, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
Subscriber, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
Log ID Status:
SUBSMGMT_QOS_EVENT_DEBUG, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
SUBSMGMT_QOS_POLICER_WARNING, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
SUBSMGMT_QOS_SHAPER_WARNING, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
SUBSMGMT_SUBSCRIBER_ACCT_STOP_PROGRESS_DEBUG, Level: NotConfigured, Plugin-
alias: None, Rate-limit: 10
SUBSMGMT_SUBSCRIBER_ADD_INFO, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
SUBSMGMT_SUBSCRIBER_DEL_INFO, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
SUBSMGMT_SUBSCRIBER_IFL_ERROR, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
SUBSMGMT_SUBSCRIBER_LOG_ERROR, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
SUBSMGMT_SUBSCRIBER_LOG_INFO, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10
SUBSMGMT_SUBSCRIBER_LOG_WARNING, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
SUBSMGMT_SUBSCRIBER_NOT_FOUND_INFO, Level: NotConfigured, Plugin-alias: None,
Rate-limit: 10
SUBSMGMT_SUBSCRIBER_NOT_FOUND_WARNING, Level: NotConfigured, Plugin-alias:
None, Rate-limit: 10
generic, Level: NotConfigured, Plugin-alias: None, Rate-limit: 10
Log ID Status:
SUBSMGMT_ACCT_SESSION_ID_NOT_FOUND_INFO, Level: NotConfigured, Plugin-alias:
None, Rate-limit: 10
SUBSMGMT_TRAP_ADD_ERROR, Level: NotConfigured, Plugin-alias: None, Rate-
limit: 10

```

2.3.3. 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
...
...
...

```

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

```

root@rtb:logd> render datastore log bds.pl.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/rtrbrick/bds_log.log).

2.3.5. Filters supported by Logging



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

2.3.5.1. 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
...
...
...
```

2.3.5.2. 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.pl.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.pl] to/from [etcd] event - CSN full table rcvd
...
...
...

```

2.3.5.3. 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.pl.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.pl] 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.pl] 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.pl] 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.pl] 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.pl] 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.pl] to/from [etcd] event - CSN process completed
...
...
...
```

2.3.5.4. 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 log map "**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

```

2.3.5.5. 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.conf] 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.conf] 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.conf] 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.conf] 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.conf] Table Type [bds_startup_stat_table] event - Add
failed - Skip index set by user

```