RBFS Logging Guide

Version 22.7.1, 25 July 2022

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. Introduction

1.1. Overview

RBFS logging is the process of writing log messages during the execution of an event. RBFS logging provides you reports about the events in the entire RBFS ecosystem at different functional areas. You can configure logging based on different severity levels available.

RBFS has been designed based on microservices architecture. An RBFS ecosystem contains multiple microservices and these microservices can be divided as Brick Daemons (BD) and other (non-BD) daemons. CtrlD and ApiGwD daemons are part of ONL, but they do not reside inside of RBFS container. BDS provides in-built infrastructure for logging which can be used by all BDS applications.

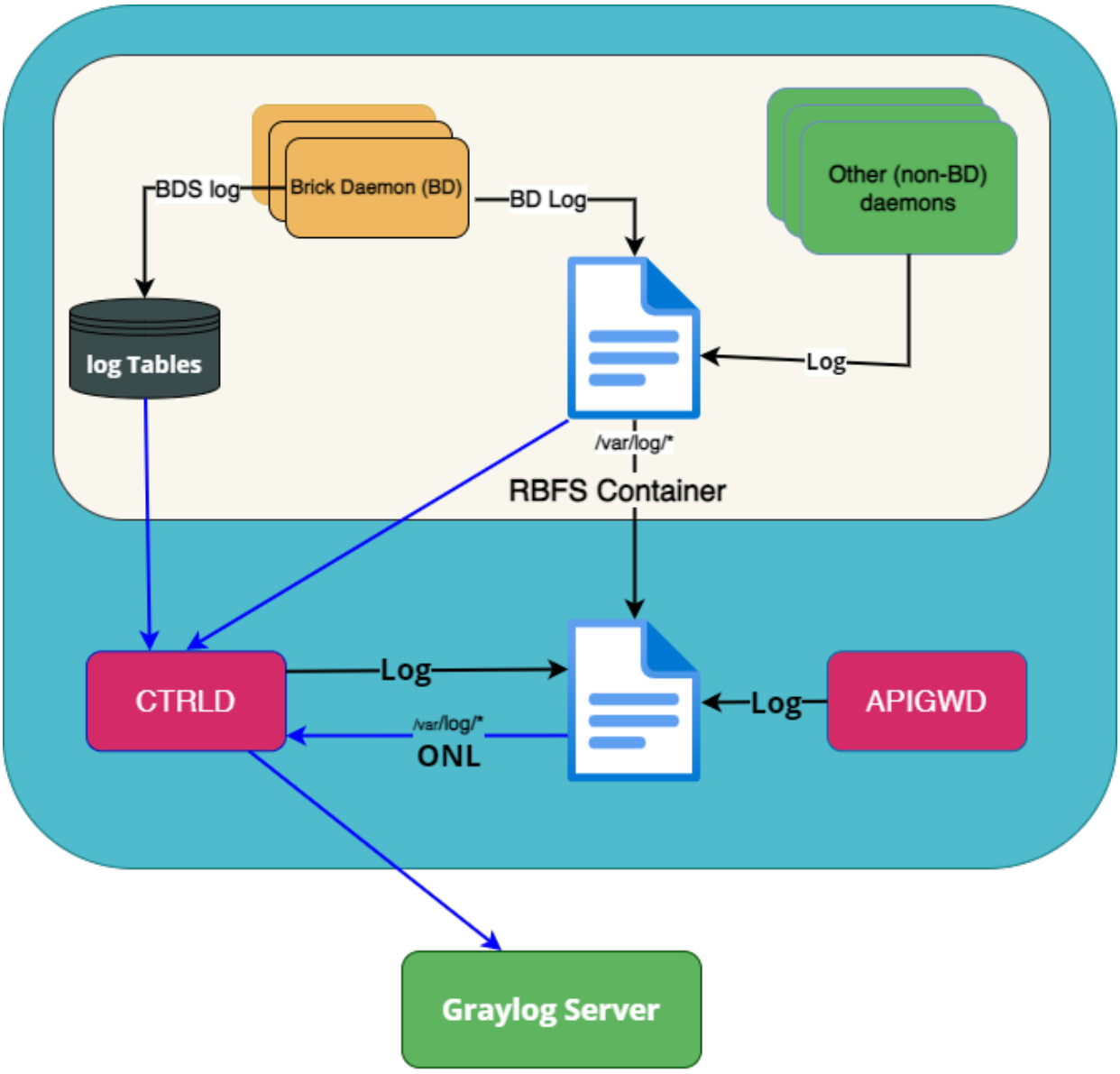
RBFS also allows you to send logs to third party applications such as Graylog where you can view and analyse the real-time data. It provides you the ability to trace out the errors of the applications in real-time.

This document provides you information about logging in the entire RBFS ecosystem. It includes the RBFS container and the host OS which is ONL in case of hardware switches.

1.2. Logging in RBFS Container

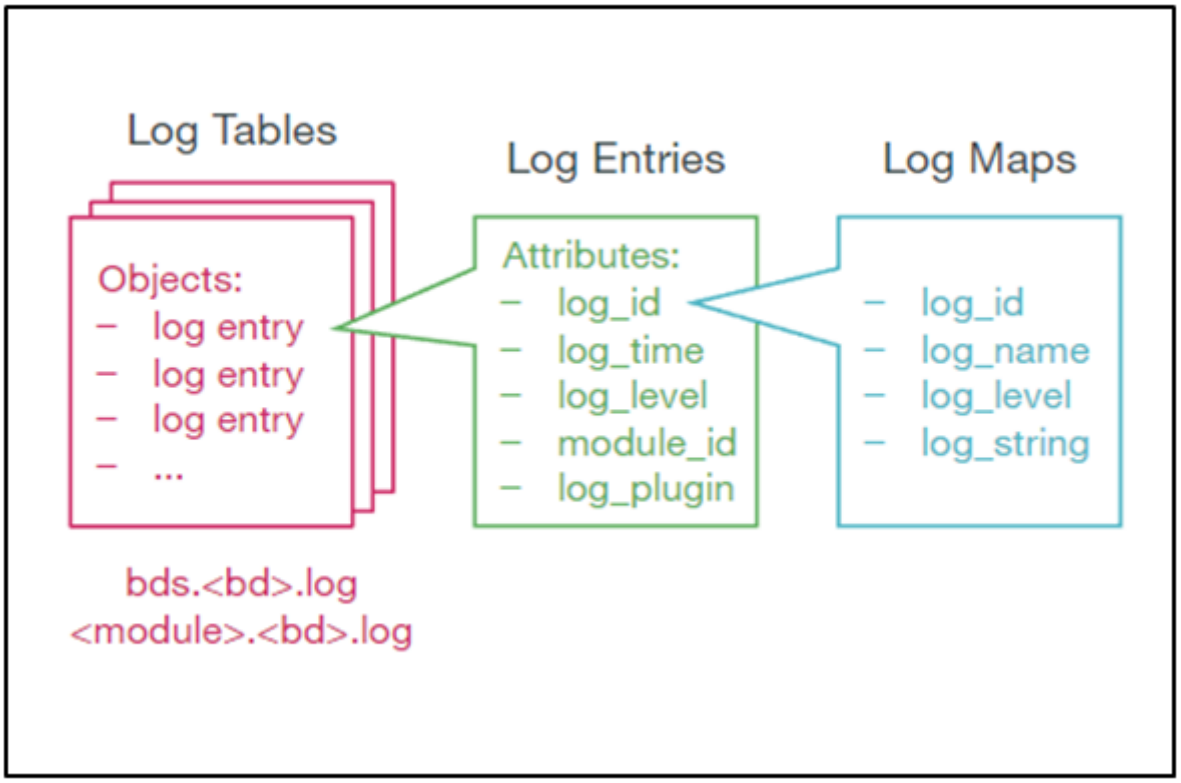
RBFS provides logging for the entire RBFS ecosystem that includes Brick Daemons (BD) and also for other (non-BD) daemons. Brick Daemons are built on top of BDS and other (non-BD) daemons (such as Prometheus) are the ones which are not dependent on BDS.

The RBFS container logging infrastructure provides in-memory (BDS) and traditional (BD) logging support for RBFS applications. The BDS logging is a low-latency and 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.



1.2.1. BDS Logging

BDS logs are stored in a log table. For every unique event, a log ID is created in RBFS. Whenever that particular event is logged, a log entry gets added to the log table. Any file that ends with .log is a log file. A log table is created for a module only when that module has at least one event logged. Every module in RBFS has at least one log table named in this format: <modulename>.<bd-name>.log.



1.2.1.1. Log Tables

BDS logs are stored in a BDS table. BDS creates a log table for each module in a BD. One entry is added to this log table for every log. Older entries are removed from the table when the number of entries exceeds more than 10,000.

1.2.1.2. Log Maps

Every log 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; instead, it stores the log map as an identifier to the actual string message.

Log map and log ID refer to the same entity.

You can access these log maps at the following location:

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

You can see the log maps, organized based on the modules that they belong to, at:

```
supervisor@rtbrick:/usr/share/rtbrick/liblog/logs$ ls
bds  bgp  fib  fwdinfra  ifm  lldpv2  pd  policy  pubsub  resmon  rib
snapshot  static  time_series
```

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

If you want to know more about a particular log map, you can perform a *grep* of the log map in this directory.

1.2.1.3. Log Groups

A log group is a collection of log maps or log IDs. Groups have 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.

1.2.1.4. Log Modules

Every BDS application consists of multiple modules. Logging can be configured for each BDS modules separately.

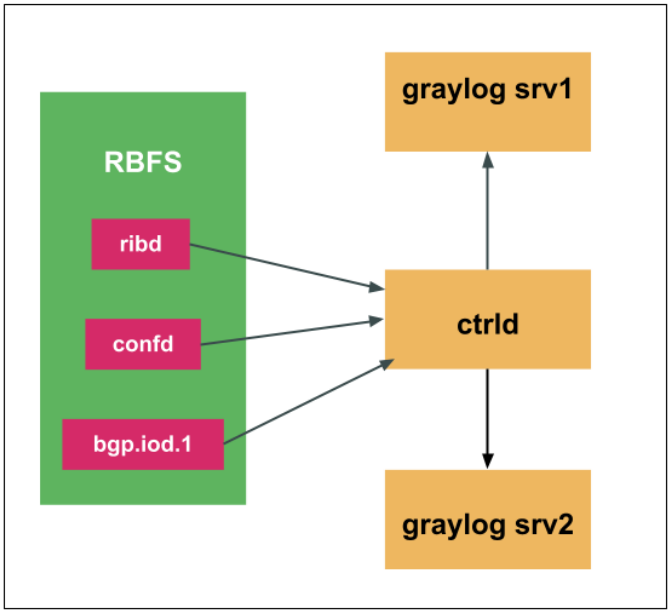
The following are RBFS modules:

```
bgp
ipoe
l2tp
pppoe
lldp
isis
ospf
igmp
```

1.2.1.5. Plugin Alias

Any logs in RBFS can be exported to an external logging destination. Currently, RBFS supports Graylog as external plugin. You need to specify the Graylog endpoint in CtrlD, and you can specify an alias name for that particular endpoint.

CtrlD is the egress node for all the GELF (Graylog Extended Log Format) messages. The brick daemons which are configured to send GELF messages to CtrlD and CtrlD forwards them to the actual GELF endpoint. This is because CtrlD enhances the GELF message with switch-global settings (for example, the serial number of the switch).



1.2.1.6. CLI Access Logs

RBFS supports sending command history log messages to Graylog, a log management software that enables real-time analysis of log messages.

The command history logs help you to understand which user has executed a specific command across multiple CLI sessions.

The log format for CLI command history logs is: *User '%s' executed command '%s'*.

1.2.1.7. Guidelines and Limitations for BDS Logging

- By default, BDS logging is enabled and the log level is set to 'Error'.
- By default, logging for BDS and PUBSUB modules have been disabled. As 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.
- You can configure log levels per BD, per module, or per group.
- Do not keep logging enabled for longer duration in a scaled setup.
- The following log levels are present in the system. Any level above the level Warning indicates that you should perform logging with caution as a scaled environment may cause a system instability.
  - Emergency
  - Alert
  - Critical
  - Error

- Warning
- Notice
- Info
- Debug
- None

If your system becomes unstable, you can remove the logging configuration using the delete log command in configuration mode.

- All log levels lower 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.

1.2.2. BD Logging

BD logging is used for basic operations of the daemons. BD logs are written to respective log files.

Example: Basic operations of daemons

```
supervisor@ipmi>user:~ $ tail -50 /var/log/rtbrick-subscriberd.1-service-
out.log
[   Error    ] <2022-05-
03T04:55:46.368396+0000>[bd]bd_ipc_get_process_for_connection: Process (ribd)
found
[   Error    ] <2022-05-
03T04:55:46.368400+0000>[bd]bd_ipcc_setup_connection_for_operational: IPCc
channel with ribd is operational
[   Error    ] <2022-05-
03T04:55:46.368404+0000>[bd]bd_ipcc_connection_created_cb: Trigerring BDS
Registered connect_cb CB2022-05-03T04:55:46.380193+0000 unix.c:139 [error]
unlink at consumer end (fd:40)
2022-05-03T04:55:46.380256+0000 unix.c:139 [error] unlink at consumer end
(fd:41)
2022-05-03T04:55:46.380281+0000 ringbuffer.c:238 [debug] shm size:1048589;
real_size:1052672; rb->word_size:263168
2022-05-03T04:55:46.380333+0000 unix.c:139 [error] unlink at consumer end
(fd:40)
2022-05-03T04:55:46.380379+0000 unix.c:139 [error] unlink at consumer end
(fd:41)
2022-05-03T04:55:46.380405+0000 ringbuffer.c:238 [debug] shm size:1048589;
real_size:1052672; rb->word_size:263168
2022-05-03T04:55:46.380451+0000 unix.c:139 [error] unlink at consumer end
(fd:40)
2022-05-03T04:55:46.380499+0000 unix.c:139 [error] unlink at consumer end
(fd:41)
2022-05-03T04:55:46.380524+0000 ringbuffer.c:238 [debug] shm size:1048589;
real_size:1052672; rb->word_size:263168
2022-05-03T04:55:46.380551+0000 lib_qb_ipcc_fsm.c:237 [info] Client
Transition. (Prev State: Connect) ---> (Event: Connect)
[   Error    ] <2022-05-
03T04:55:46.380562+0000>[bd]bd_ipcc_connection_created_cb: ----
[IPCS_CLIENT]: Step 1: Created ----
[   Error    ] <2022-05-
03T04:55:46.380568+0000>[bd]bd_ipc_get_process_for_connection: Looking up
process for QB identity(igmp.iod.1@ipmi)
[   Error    ] <2022-05-
03T04:55:46.380575+0000>[bd]bd_ipc_get_process_for_connection: Process
(igmp.iod.1) found
[   Error    ] <2022-05-
03T04:55:46.380580+0000>[bd]bd_ipcc_setup_connection_for_operational: IPCc
channel with igmp.iod.1 is operational
[   Error    ] <2022-05-
03T04:55:46.380583+0000>[bd]bd_ipcc_connection_created_cb: Trigerring BDS
Registered connect_cb CB
```

1.2.2.1. BD Log Files

BD logs are traditional log files that are written to a disk and these log files are CPU-intensive. The following table provides log files and their associated modules/services which RBFS uses:

Log file	Associated Module/Service
rtbrick-alertmanager-service-out.log	Alert Manager
rtbrick-clixon-restconf-service-out.log	Clixon restconf daemon
rtbrick-confd-service-out.log	confd daemon
rtbrick-etcd-service-out.log	etcd daemon
rtbrick-fibd-service-out.log	fibd daemon
rtbrick-hostconfd-service-out.log	hostconfd daemon
rtbrick-ifmd-service-out.log	ifmd daemon
rtbrick-iptables-service-out.log	iptables daemon
rtbrick-lldpd-service-out.log	lldpd daemon
rtbrick-mribd-service-out.log	mribd daemon
rtbrick-opsd-service-out.log	opsd daemon
rtbrick-policy.server-service-out.log	policy server daemon
rtbrick-poold-service-out.log	poold daemon
rtbrick-prometheus-service-out.log	prometheus daemon
rtbrick-resmond-service-out.log	resmond daemon
rtbrick-restconfd-service-out.log	restconfd daemon
rtbrick-ribd-service-out.log	ribd daemon
rtbrick-staticd-service-out.log	staticd daemon

1.2.2.2. Guidelines and Limitations for BD Log Files

- By default, BD logging is enabled and the log level is set to 'Error'.

- For scalability, enable logging only for the BD which is problematic. It is not recommended to enable logging for all the BDs.
- If your system becomes unstable, you can disable logging using the delete log command in the configuration mode.
- BD logging supports log file rotation.

1.2.3. Syslog

Syslog is an API based logging mechanism provided by Linux. Some of the open source libraries present in RBFS use Syslog as a logging mechanism. CLI front-end and libqb also use Syslog for logging purpose.

All the RBFS syslog can be found at: /var/log/syslog.

1.2.3.1. System Log Files

The following table provides system log files and their associated modules/services which RBFS uses:

Log file	Associated Module/Service
alternatives.log	Information by the update-alternatives
auth.log	Pluggable Authentication Module (PAM)
btmp	Failed login user information
dpkg.log	Debian Package Manager
fail2ban.log	Fail2ban
fontconfig.log	Fontconfig
kern.log	Kernal
tallylog	pam_tally
ubuntu-advantage-timer.log	The Ubuntu Advantage (UA) client
wtmp	User and accounting information

1.3. Logging in ONL

In RBFS, there are daemons such as CtrlD and ApiGwD that run on the ONL host. RBFS provides logging for these daemons.

1.3.1. CtrlD and ApiGwD Logs

CtrlD logging provide log messages of events related to business, elements, ZTP, and security. ApiGwD logs contain details about who accessed the API and how

they accessed it.

ApiGwD and CtrlD send different log messages about status changes or progress of processes to the GELF endpoint.

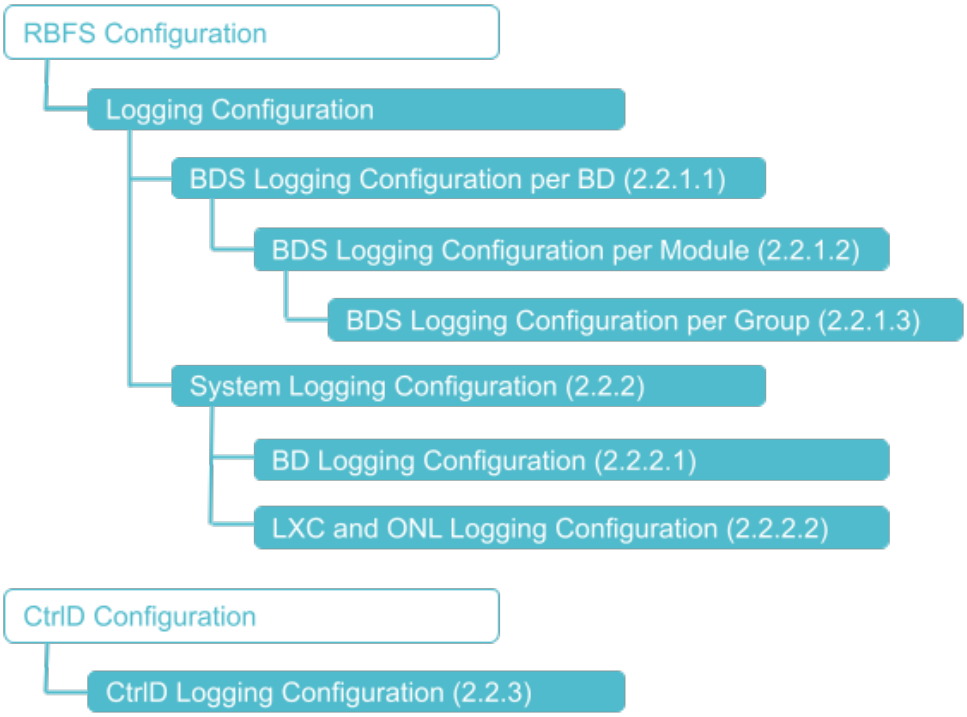
1.4. Supported Platforms

Not all features are necessarily supported on each hardware platform. Refer to the *Platform Guide* for the features and the sub-features that are or are not supported by each platform.

2. Logging Configuration

2.1. Configuration Hierarchy

The diagram illustrates the logging configuration hierarchy.



2.2. Configuration Syntax and Commands

The following sections describe the logging configuration syntax and commands.

2.2.1. BDS Logging Configuration

You can configure BDS logging for a BD, a module, and for a group.

A specific configuration takes priority over a generic configuration. For example, if you have configured a global log level of bgp.iod.1 to "warning", and you have configured a log level of bgp module to "notice", then the final log level of bgp will be "notice".

2.2.1.1. Configuring BDS Logging for a BD

BDS logging can be configured for a BD.

Syntax:

set log bd <bd-name> <option> ...

Attribute	Description
<bd-name>	Configure for the specified BD name.
all	Configure for all BDs.
module <module-name>	Module name. For more information, see the section "Configuring BDS Logging for a Module".
plugin-alias alias-name <alias-name>	Plugin alias name
plugin-alias level <level>	Log severity level. You can filter logs based on the log severity levels for sending to Graylog. This will help you to send only the required log messages to Graylog instead of sending a whole lot of data. For example, if you configure Warning as the severity level, logs with the severity level 'Warning' and all the log levels above Warning such as Error, Critical, Alert, and Emergency will be sent to Graylog.

Example 1: BDS logging for a BD Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {},
    "rtbrick-config:log": {
      "bd": [
        {
          "bd-name": "bgp.iod.1",
          "level": "info",
          "plugin-alias": {
            "alias-name": "ztp",
            "level": "error"
          }
        }
      ]
    }
```

2.2.1.2. Configuring BDS Logging for a Module

Logging can be configured for a module such as BGP, IS-IS, and so on.

Syntax:

set log module <module-name> <option> ...

Attribute	Description
<module-name>	Module name
group <group-name>	BDS log module log-group configuration. For more information, see the section "Configuring BDS Logging for a Group".
level <level>	Log severity level.
plugin-alias alias-name <alias-name>	Plugin alias name
plugin-alias level <level>	Log level. You can filter logs based on the log severity levels for sending to Graylog. This will help you to send only the required log messages to Graylog instead of sending a whole lot of data. For example, if you configure Warning as the severity level, logs with the severity level 'Warning' and all the log levels above Warning such as Error, Critical, Alert, and Emergency will be sent to Graylog.

Example 1: Logging Module Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {},
    "rtbrick-config:log": {
      "module": [
        {
          "module-name": "igmp",
          "level": "info",
          "plugin-alias": {
            "alias-name": "graylog-srv1",
            "level": "warning"
          }
        }
      ]
    }
  }
}
```

2.2.1.3. Configuring BDS Logging for a Group

Logging can be configured at the group hierarchy level.

Syntax:

set log module <module-name> **group** <group-name> <option> ...

Attribute	Description
<group-name>	Group name
level <level>	Specifies the level of the plug-in alias.
rate-limit <rate-limit>	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. The default value is 10.
plugin-alias alias-name <alias-name>	Plugin alias name
plugin-alias level <level>	Specifies the level of the plug-in alias. You can filter logs based on the log severity levels for sending to Graylog. This will help you to send only the required log messages to Graylog instead of sending a whole lot of data. For example, if you configure Warning as the severity level, logs with the severity level 'Warning' and all the log levels above Warning such as Error, Critical, Alert, and Emergency will be sent to Graylog.

Example 1: Logging Group Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:log": {
      "module": [
        {
          "module-name": "bgp",
          "group": [
            {
              "group-name": "interface",
              "level": "warning"
            }
          ]
        }
      ]
    }
  }
}
```

2.2.2. System Logging Configuration

2.2.2.1. BD Logging Configuration

Logging can be configured for a specific BD or all BDs.

Syntax:

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

Attribute	Description
bd <bd-name>	BD logging configuration for a specific BD.
level <level>	BD logging configuration for all BD.

Example: BD Logging Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:log": {
      "system": {
        "bd": [
          {
            "bd-name": "bgp.appd.1",
            "level": "info"
          }
        ]
      }
    }
  }
}
```

2.2.2.2. LXC and ONL Logging Configuration

You can configure an external log management server to transport logs for real-time analysis. Currently, Graylog is the log management server supported by RBFS.

You can send logs from Linux and ONL system facilities such as auth, authpriv, daemon, and kern to Graylog.

Syntax:

set log system facility <facility-name> **plugin-alias** <attribute> <value>

Attribute	Description
<facility-name>	Supported facilities include all, auth, authprive, daemon, and kern.
plugin-alias alias-name <alias-name>	Plugin alias name

Attribute	Description
plugin-alias level <level>	Specifies the level of the plug-in alias. You can filter logs based on the log severity levels for sending to Graylog. This will help you to send only the required log messages to Graylog instead of sending a whole lot of data. For example, if you configure Warning as the severity level, logs with the severity level 'Warning' and all the log levels above Warning such as Error, Critical, Alert, and Emergency will be sent to Graylog.

Example 1: System Log Configuration

```
{
  "ietf-restconf:data": {
    "rtbrick-config:log": {
      "system": {
        "facility": [
          {
            "facility-name": "kern",
            "plugin-alias": {
              "alias-name": "graylog",
              "level": "notice"
            }
          }
        ]
      }
    }
  }
}
```

2.2.3. CtrlD Logging Configuration

When you configure a plugin alias in RBFS, the log message is forwarded to CtrlD. CtrlD forwards it to the corresponding Graylog endpoint alias that you configured.

You must add the Graylog endpoint in the CtrlD start-up configuration before configuring a plugin-alias in RBFS.

If the configured plugin-alias name does not match any of the Graylog endpoint name configured in CtrlD, the log is sent to the default Graylog endpoint ("graylog_url").

For Graylog support, you need to configure logging in the CtrlD.

RBFS logs can be sent to Graylog servers. This can be achieved by configuring a plugin alias in CtrlD.

The following section describe the tasks to be performed to configure the plugin alias in CtrlD:

config.json

This section describes the main configuration file of CtrlD. This file can be changed via API. If it is changed on the file system, CtrlD has to be restarted.

/etc/rtbrick/ctrld/config.json example

```
{
  "graylog_enable": true,
  "graylog_url": "http://10.200.32.49:12201/gelf",
  "graylog_endpoints": [
    {
      "name": "ztp",
      "url": "http://192.168.202.46:12201/gelf"
    }
  ]
}
```

Table 1. /etc/rtbrick/ctrld/config.json format

Name	Type	Description
graylog_enable	bool	To Enable all Graylog outgoing messages
graylog_url	string	Graylog url e.g. http://127.0.0.1:12201/gelf
graylog_heart_beat_interval	string	Graylog heartbeat Interval in seconds (0 means deactivated)

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>graylog_endpoints</td><td colspan="2">GraylogEndpoints allows to specify multiple graylog endpoints by name. If a log to a specific endpoint is requested and the endpoint is not available, the log is send to the default Graylog endpoint.

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>name</td><td>string</td><td>Logical name of the entpoint e.g.: ztp for ztp messages.</td></tr>
<tr><td>url</td><td>string</td><td>Graylog url e.g. http://127.0.0.1:12201/gelf</td></tr>
<tr><td>disable</td><td>string</td><td>Disables this endpoint.</td></tr>
</table>

If the default endpoint is disabled, but the specific one is enabled than the message to the specific endpoint will be sent.

If the default endpoint is enabled, but the specific one is disabled than the message to the specific endpoint will not be sent.</td></tr>
</table>

3. Operational Commands

The logging operational commands provide information about the logging operations. They are used to show logs in the system, log configuration status and so on.

3.1. BDS Logging

The BDS logging show commands provide information about the BDS logging operations. With the BDS logging operational commands, you can verify BDS logging configuration status and view BDS logs.

3.1.1. Verifying BDS Logging Configuration Status

3.1.1.1. Show Log Status

This command shows log configuration status for all modules except infrastructure modules in all BDs. The default show log status displays the summary of log status for the whole system and there are options available to show specific module or BD log status.

Syntax:

show log status <attribute> <value>

Option	Description
-	Without any option, the command displays the log configuration status for all modules except infrastructure modules in all BDs.
bd <bd-name>	Displays log status of the all modules including infrastructure in the specified BD.
detail	Displays the log configuration status in detail all the way till log IDs.
module <module-name>	Displays the log status for a given module in all BDs where this module is running.

Example 1: Summary of log status

```
supervisor@rtbrick: cfg> show log status
Module log status:
  bds_mock:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin
```

```
Plugin Level   Rate limit
        generic                 error          None                      none
10
  bgp:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin
Plugin Level   Rate limit
        config                  error          None                      none
10
        general                 error          None                      none
10
        generic                 error          None                      none
10
        instance                error          None                      none
10
        interface               warning        None                      none
10
        message                 error          None                      none
10
        peer                    error          None                      none
10
    bgp.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin
Plugin Level   Rate limit
        config                  error          None                      none
10
        general                 error          None                      none
10
        generic                 error          None                      none
10
        instance                error          None                      none
10
        interface               warning        None                      none
10
        message                 error          None                      none
10
        peer                    error          None                      none
10
    bgp.iod.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin
Plugin Level   Rate limit
        config                  error          None                      none
10
        general                 error          None                      none
10
        generic                 error          None                      none
10
        instance                error          None                      none
10
        interface               warning        None                      none
10
        message                 error          None                      none
```

```
10
        peer                    error          None                    none
10
  fib:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin
Plugin Level   Rate limit
        adjacency               error          None                    none
10
        bds                     error          None                    none
10
        config                  error          None                    none
10
        general                 error          None                    none
10
        generic                 error          None                    none
10
        instance-afi-safi       error          None                    none
10
        interface-events        error          None                    none
10
        route                   error          None                    none
10
  fwdinfra:
    ribd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin
Plugin Level   Rate limit
        generic                 error          None                    none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin
Plugin Level   Rate limit
        generic                 error          None                    none
10
    staticd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin
Plugin Level   Rate limit
        generic                 error          None                    none
10
    ifmd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin
Plugin Level   Rate limit
        generic                 error          None                    none
10
    mribd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin
```

```
Plugin Level   Rate limit
        generic                 error          None                      none
10
  hostconfd:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin
Plugin Level   Rate limit
        bds                     error          None                      none
10
        config                  error          None                      none
10
        functional              error          None                      none
10
        generic                 error          None                      none
10
  ifm:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin
Plugin Level   Rate limit
        bds                     error          None                      none
10
        config                  error          None                      none
10
        general                 error          None                      none
10
        generic                 error          None                      none
10
        interface-events        error          None                      none
10
    ifmd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin
Plugin Level   Rate limit
        bds                     error          None                      none
10
        config                  error          None                      none
10
        general                 error          None                      none
10
        generic                 error          None                      none
10
        interface-events        error          None                      none
10
  igmp:
    pim.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                   Level          Plugin
Plugin Level   Rate limit
        config                  error          None                      none
10
        generic                 error          None                      none
10
```

```
        igmp-interface-events    error          None                      none
10
        igmp-membership-events   error          None                      none
10
        igmp-packet-events       error          None                      none
10
        igmp-route-events        error          None                      none
10
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin
Plugin Level   Rate limit
        config                   error          None                      none
10
        generic                  error          None                      none
10
        igmp-interface-events    error          None                      none
10
        igmp-membership-events   error          None                      none
10
        igmp-packet-events       error          None                      none
10
        igmp-route-events        error          None                      none
10
    igmp.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin
Plugin Level   Rate limit
        config                   error          None                      none
10
        generic                  error          None                      none
10
        igmp-interface-events    error          None                      none
10
        igmp-membership-events   error          None                      none
10
        igmp-packet-events       error          None                      none
10
        igmp-route-events        error          None                      none
10
    pim.iod.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin
Plugin Level   Rate limit
        config                   error          None                      none
10
        generic                  error          None                      none
10
        igmp-interface-events    error          None                      none
10
```

Example 2: View of module log status

```
supervisor@rtbrick: cfg>  show log status module bgp
Module log status:
  bgp:
    confd:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin
Plugin Level   Rate limit
        config                   error          None                     none
10
        general                  error          None                     none
10
        generic                  error          None                     none
10
        instance                 error          None                     none
10
        interface                warning        None                     none
10
        message                  error          None                     none
10
        peer                     error          None                     none
10
    bgp.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin
Plugin Level   Rate limit
        config                   error          None                     none
10
        general                  error          None                     none
10
        generic                  error          None                     none
10
        instance                 error          None                     none
10
        interface                warning        None                     none
10
        message                  error          None                     none
10
        peer                     error          None                     none
10
    bgp.iod.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin
Plugin Level   Rate limit
        config                   error          None                     none
10
        general                  error          None                     none
10
        generic                  error          None                     none
10
        instance                 error          None                     none
10
        interface                warning        None                     none
10
        message                  error          None                     none
10
```

```
        peer                    error          None                    none
10
```

Example 3: Log status of the all modules including infrastructure in the specified BD.

```
supervisor@rtbrick: cfg> show log status bd bgp.appd.1
System/File log status:
  Level: error

Module log status:
  bd:
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      Group                   Level          Plugin                  Level
Rate limit
      generic                 none           None                    none
10
      http                    none           None                    none
10
  bds:
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      Group                   Level          Plugin                  Level
Rate limit
      generic                 none           None                    none
10
      object                  none           None                    none
10
      table                   none           None                    none
10
      trim                    none           None                    none
10
  bgp:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                   Level          Plugin                  Level
Rate limit
      config                  error          None                    none
10
      general                 error          None                    none
10
      generic                 error          None                    none
10
      instance                error          None                    none
10
      interface               warning        None                    none
10
      message                 error          None                    none
10
      peer                    error          None                    none
10
  license:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
```

```
      Group                    Level          Plugin                   Level
Rate limit
      generic                  error          None                     none
10
      internal                 error          None                     none
10
      operational              error          None                     none
10
  policy:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                   Level
Rate limit
      BDS                      error          None                     none
10
      Configuration            error          None                     none
10
      Enforcement              error          None                     none
10
      Generation               error          None                     none
10
      generic                  error          None                     none
10
  poold:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                   Level
Rate limit
      generic                  error          None                     none
10
  pubsub:
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                   Level
Rate limit
      generic                  none           None                     none
10
  secure_management:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                   Level
Rate limit
      generic                  error          None                     none
10
  snapshot:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                   Level
Rate limit
      generic                  error          None                     none
10
  time_series:
    Level: error, Plugin: None, Plugin Level: none
    Log group status:
      Group                    Level          Plugin                   Level
Rate limit
      generic                  error          None                     none
10
```

Example 4: Log status for given module in the given BD

```
supervisor@rtbrick: cfg>  show log status module bgp bd bgp.appd.1
Module log status:
  bgp:
    bgp.appd.1:
      Level: error, Plugin: None, Plugin Level: none
      Log group status:
        Group                    Level          Plugin
Plugin Level   Rate limit
        config                   error          None                     none
10
        general                  error          None                     none
10
        generic                  error          None                     none
10
        instance                 error          None                     none
10
        interface                warning        None                     none
10
        message                  error          None                     none
10
        peer                     error          None                     none
10
```

Example 5: Log status for active logs per log ID

```
supervisor@rtbrick: cfg> show log status bd bgp.appd.1 detail
System/File log status:
  Level: error

Module log status:
  bd:
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      generic, Level: none, Plugin: None, Plugin Level: none, Rate limit: 10
      http, Level: none, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID
Level          Plugin                    Level          Rate limit
        HTTP_JWK_FILE_JSON_PARSE_FAILED
none           None                      none           10
        HTTP_JWK_FILE_MEM_ALLOC_FAILED
none           None                      none           10
        HTTP_JWK_FILE_MISSING
none           None                      none           10
        HTTP_JWK_FILE_OPEN_FAILED
none           None                      none           10
        HTTP_JWK_FILE_READ_FAILED
none           None                      none           10
        HTTP_JWK_MISSING_KEY
none           None                      none           10
        HTTP_JWK_MULTIPLE_KEYS
none           None                      none           10
        HTTP_SEND
none           None                      none           10
        HTTP_WRITE_BUFFER_MEM_ALLOC_FAILED
none           None                      none           10
        HTTP_WRITE_PRINTF_FAILED
none           None                      none           10
  bds:
    Level: none, Plugin: None, Plugin Level: none
    Log group status:
      generic, Level: none, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID
Level          Plugin                    Level          Rate limit
        BDS_ATTRIBUTE_TEMPLATE_EVENT
none           None                      none           10
        BDS_INVALID_PARAMS
none           None                      none           10
        BDS_PUBSUB_ERROR_STATUS
none           None                      none           10
        BDS_QUEUE_TABLE
none           None                      none           10
        BDS_ROOT_EVENT
none           None                      none           10
        BDS_TEST_LOG
none           None                      none           10
      object, Level: none, Plugin: None, Plugin Level: none, Rate limit: 10
     Log ID status:
        LOG ID
Level          Plugin                    Level          Rate limit
```

3.1.2. Viewing BDS Logs

3.1.2.1. Show Log

This command shows BDS logs in the log tables. By default, the command show log shows all logs present in the log tables. Various command options are available to filter and display logs. Also, a command option is available to send logs into a file.

Syntax:

show log <option>

Option	Description
-	Without any option, the command displays all logs present in the log tables.
filter <level> <module>	Render output of the log can be filtered for the specified module or the log level. You can specify filter level or module with any of the view log command, so that the logs are filtered based on the specified level or module.
format <abstract |summary |detailed>	Choose any of the three formats: abstract, summary, or detailed. This command provides output in the specified output format. You can specify any of the options at end of any of the view log commands to show logs in a specific output format. By default, summary is the log format. Abstract: Shows logs without metadata Detailed: Shows logs in detail with metadata Summary: Shows logs in summary view with metadata
table <name>	Displays logs from a specified log table. Every BD includes multiple log tables. By default, log will be rendered from every log table, if not specified.
to file <filename>	Name of the file in which logs are transported. You can specify to file and the file name at end of any view log command to transport the log to the file.

Example 1: View of logs

```
supervisor@rtbrick: op> show log
[  Error   ] <2021-07-09T04:35:53.184694+0000> Table
[global.hostconfd.table.config] - event Failed to open file
[  Error   ] <2021-07-09T04:35:53.184771+0000> Table
[global.hostconfd.table.config] - event Could not create snapshot block
[  Error   ] <2021-07-09T04:35:53.184849+0000> Table
[global.hostconfd.table.config] - event Failed to open file
[  Error   ] <2021-07-09T04:35:53.184866+0000> Table
[global.hostconfd.table.config] - event Could not create snapshot block
[  Error   ] <2021-07-09T04:35:53.201029+0000> Table [global.time-
series.config] - event Failed to open file
[  Error   ] <2021-07-09T04:35:53.201052+0000> Table [global.time-
series.config] - event Could not create snapshot block
[  Error   ] <2021-07-09T04:35:53.201106+0000> Table [global.time-
series.config] - event Failed to open file
[  Error   ] <2021-07-09T04:35:53.201125+0000> Table [global.time-
series.config] - event Could not create snapshot block
[  Error   ] <2021-07-09T04:35:53.222660+0000> Table
[secure.global.system.table.config] - event Failed to open file
[  Error   ] <2021-07-09T04:35:53.222679+0000> Table
[secure.global.system.table.config] - event Could not create snapshot block
[  Error   ] <2021-07-09T04:36:00.720574+0000> Table [global.tacacs.config]
Object [name - tacacs_config_object] attribute - tacacs_server_ip not found
event TACACS Server Hostconfd Config
supervisor@ldev: op>
```

Example 2: Summary view for the show log table

```
supervisor@rtbrick: op> show log 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>
```

Example 3: View of applied filters on all logs from a single table

```
supervisor@: op> show log 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.confrtbrickig.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
```

Example 4: Show Log to File

```
supervisor@rtbrick: op> show log to file test.log
supervisor@rtbrick: op> exit
supervisor@rtbrick:~ $ cat test.log
[   info   ] <2022-05-10T11:53:15.399613+0000> Global config for
Instance(default) is added
[   info   ] <2022-05-10T11:53:15.400666+0000> Global address family(ipv4,
unicast) is added in Instance(default)
[   info   ] <2022-05-10T11:53:15.400711+0000> Global address family(ipv4,
multicast) is added in Instance(default)
[   info   ] <2022-05-10T11:53:15.400729+0000> Global address family(ipv4,
labeled-unicast) is added in Instance(default)
[   info   ] <2022-05-10T11:53:15.400744+0000> Global address family(ipv6,
unicast) is added in Instance(default)
[   info   ] <2022-05-10T11:53:15.400758+0000> Global address family(ipv6,
multicast) is added in Instance(default)
[   info   ] <2022-05-10T11:53:15.400773+0000> Global address family(ipv6,
labeled-unicast) is added in Instance(default)
[   info   ] <2022-05-10T11:53:15.400787+0000> Global address family(mpls,
unicast) is added in Instance(default)
[   info   ] <2022-05-10T11:53:07.757687+0000> User 'supervisor' executed
command 'show log'
[   info   ] <2022-05-10T11:53:10.751083+0000> User 'supervisor' executed
command 'show log'
[   info   ] <2022-05-10T11:53:15.338391+0000> Failed due to bds events-
{table:global_ntp_config_tbl}
[   info   ] <2022-05-10T11:53:15.338442+0000> Failed due to bds events-
{table:ipmi_user_config_table}
[   info   ] <2022-05-10T11:53:15.338469+0000> Failed due to bds events-
{table:lum_config_table}
[   info   ] <2022-05-10T11:53:15.338491+0000> Failed due to bds events-
{table:ipmi_interface_config_table}
[   info   ] <2022-05-10T11:53:15.338509+0000> Failed due to bds events-
{table:authorization_config_table}
[   info   ] <2022-05-10T11:53:07.644676+0000> Commit Success
[   info   ] <2022-05-10T11:53:15.337573+0000> Table -
global_tacacs_config_tbl object not found event secure_hostconfd_write_config
[   info   ] <2022-05-10T11:53:15.337602+0000> Table [global.tacacs.config]
Object [name - tacacs_config_object] attribute - tacacs_server_ip not found
event TACACS Server Hostconfd Config
[   info   ] <2022-05-10T11:53:15.337241+0000> No objects present in alert
configuration table to send to hostconfd
```

Example 5: Logs for filter level

```
supervisor@rtbrick: op> show log 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
```

Example 6: Logs for specified module

```
supervisor@rtbrick: op> show log 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>
```

Example 7: View of the logs in abstract format

```
supervisor@rtbrick: op> show log 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
```

3.1.3. BDS Logging Clear Commands

Clear commands allow you to delete existing logs.

3.1.4. Clear Logs

This commands resets all logs.

Syntax:

clear log <option> ...

Option	Description
bd <bd-name>	Clear all BDS logs from the given BD.
table <table-name>	Clears the specified log table.

3.2. BD Logging

The BD logging provides information about the BD logging operations. You can view BD logs at its specified locations.

3.2.1. Viewing BD Log Files

BD log files are available at the following directory:

/var/log

For information about all the log files in var/log, refer to the section, *BD Log Files*.

3.3. Viewing ONL Log Files

ONL log files are available at the following directory:

CtrlD log files are available at:

/var/log/rtbrick-ctrld.log`

Example: CtrlD Logs

```
supervisor@onl>ufi06.q2c.u25.r4.nbg.rtbrick.net:/var/log $ tail -10 rtbrick-
ctrld.log
2022-05-04 08:31:24 UTC INF HTTP request completed host=10.200.134.29:19091
method=GET
path=/api/v1/rbfs/elements/ufi06.q2c.u25.r4.nbg.rtbrick.net/services/promethe
us/proxy/federate remote_addr=10.200.32.49:41508 request_id=R4n9-tadb
statuscode=200 user_name= user_subject=
2022-05-04 08:31:24 UTC INF HTTP request completed host=10.200.134.29:19091
method=GET path=/api/v1/ctrld/info remote_addr=10.200.128.121:34158
request_id=yFn9Ptadb statuscode=200 user_name= user_subject=
2022-05-04 08:31:24 UTC INF HTTP request completed host=10.0.3.1:19091
method=GET path=/api/v1/ctrld/system/clock remote_addr=10.0.3.10:57824
request_id=_cno-gadu statuscode=200 user_name= user_subject=
2022-05-04 08:31:25 UTC INF HTTP request completed host=10.200.134.29:19091
method=GET
path=/api/v1/rbfs/elements/rtbrick/services/PROMETHEUS/proxy/federate
remote_addr=192.168.202.54:39654 request_id=DXSoPgaHu statuscode=404
user_name= user_subject=
2022-05-04 08:31:29 UTC INF HTTP request completed host=10.200.134.29:19091
method=GET
path=/api/v1/rbfs/elements/ufi06.q2c.u25.r4.nbg.rtbrick.net/services/promethe
us/proxy/federate remote_addr=10.200.32.49:41508 request_id=iQqo-gaHu
statuscode=200 user_name= user_subject=
2022-05-04 08:31:34 UTC INF HTTP request completed host=10.200.134.29:19091
method=GET
path=/api/v1/rbfs/elements/ufi06.q2c.u25.r4.nbg.rtbrick.net/services/promethe
us/proxy/federate remote_addr=10.200.32.49:41508 request_id=LWM9PgrHb
statuscode=200 user_name= user_subject=
```

ApiGwD log files are available at:

/var/log/rtbrick-apigwd.log

Example: API Gateway Logs

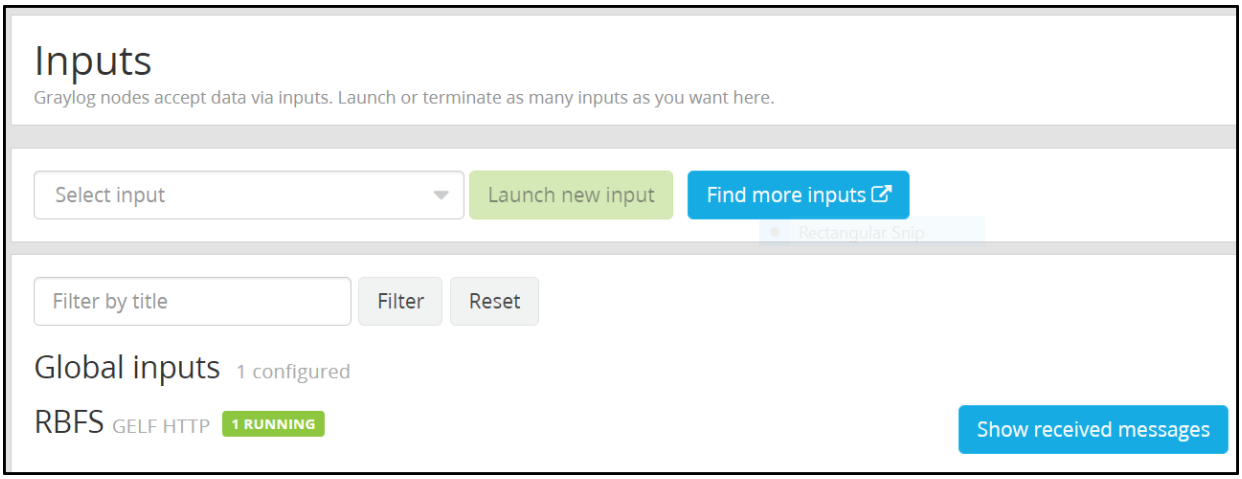
```
supervisor@onl>ufi06.q2c.u25.r4.nbg.rtbrick.net:/var/log $ cat rtbrick-
apigwd.log
Tue May  3 23:27:12 UTC 2022 Starting rtbrick apigwd service
Version: v0.10.0-internal.20220222110916+Bdevelopment.C2a896336 (built with
go1.17.7)
2022-05-03 23:27:12 UTC INF development/apigwd/pkg/options/options.go:158 >
watching for file change /etc/rtbrick/apigwd/config.json
2022-05-03 23:27:12 UTC INF development/apigwd/cmd/apigwd/server.go:29 >
listening on listen_addr=:12321
2022-05-03 23:27:12 UTC INF development/apigwd/cmd/apigwd/server.go:89 >
certman: certificate and key loaded
2022-05-03 23:27:12 UTC INF development/apigwd/cmd/apigwd/server.go:89 >
certman: watching for cert and key change
2022-05-03 23:28:15 UTC INF development/apigwd/pkg/options/options.go:165 >
watch event: {/etc/rtbrick/apigwd/config.json 2}
2022-05-03 23:28:15 UTC INF development/apigwd/pkg/options/options.go:165 >
watch event: {/etc/rtbrick/apigwd/config.json 2}
2022-05-03 23:28:15 UTC INF development/apigwd/cmd/apigwd/routes.go:62 >
reloaded request limiter config
2022-05-03 23:28:15 UTC INF development/apigwd/cmd/apigwd/routes.go:62 >
reloaded request limiter config
```

3.4. Viewing Logs in Graylog

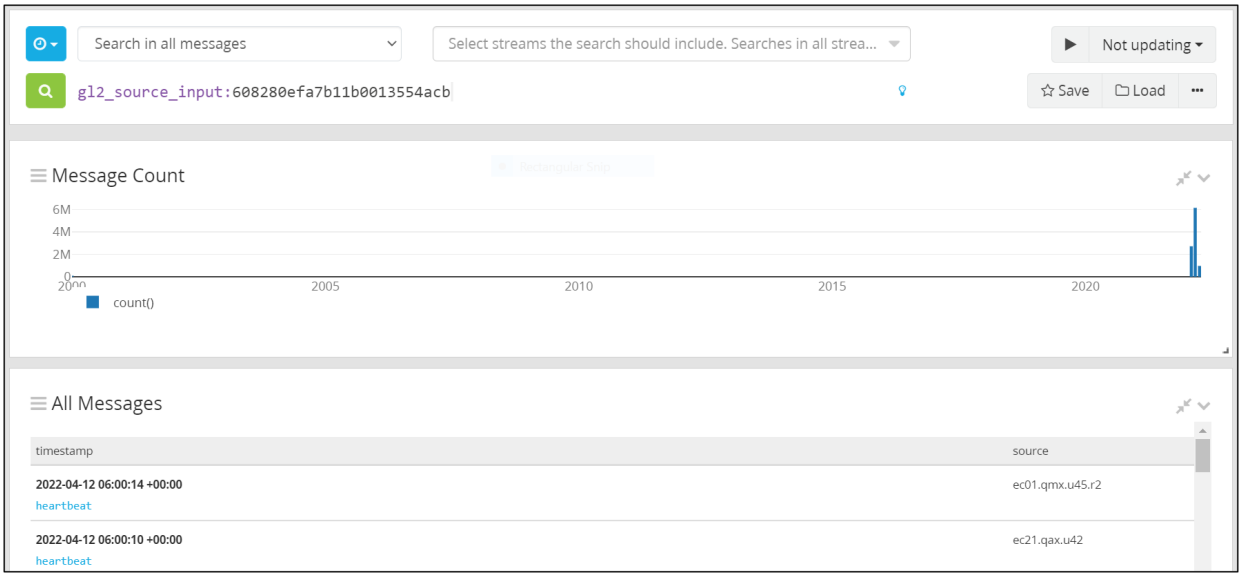
For viewing your log data on Graylog, perform the following steps:

1. Open the Graylog webpage.
2. Log in using your user credentials.

3. Click **System** and select **Input**.



4. Click the **Show received message** tab.



The log messages page appears.