



RBFS CLI Overview

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

©Copyright 2022 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 CLI Overview	3
1.1. Starting CLI	3
1.2. Switching different modes	3
1.3. Committing CLI configurations	4
1.4. Adding a Configuration Description	4
1.5. Viewing configuration	5
1.6. Viewing uncommitted changes	5
1.7. Saving configuration	5
1.8. Loading configuration	5
1.9. Viewing configuration in set format	6
1.10. Viewing configuration in a given hierarchy	6
1.11. Turning Paging on/off	6
1.12. Discarding the uncommitted diff	6
1.13. Deleting entire running-config at a time	6
1.14. Rollback to previous committed configurations	7
1.14.1. Viewing commit log	7
1.14.2. Viewing the configuration differences in the current and previous versions	7
1.14.3. Rollback to a previously committed configuration	7
1.14.4. Rollback to a specific version of the software configuration	8
1.15. Displaying Configurations in a Specific Format	8
1.16. Displaying Core Files	9
1.17. Displaying Date And Time	10
1.18. Rebooting Containers and Hosts	10
1.19. Displaying Command History	11
1.20. Supported Platforms	12

1. RBFS CLI Overview

The RBFS command-line interface (CLI) provides a set of commands that you can use to configure and monitor devices running RBFS.

This document contains information about the RBFS CLI.

1.1. Starting CLI

The following example shows how to start the RBFS CLI.

```
supervisor@rtbrick>LEAF01:~$ cli
supervisor@rtbrick>LEAF01: op>
```



'op>' shows you are in operational mode.

CLI Prompt

The RBFS CLI prompt reflects the static hostname and host OS hostname. In RBFS, the static hostname is the container name and the dynamic hostname is derived from DHCP.

The format of the RBFS CLI prompt is as follows:

`<username> @ <static_hostname> > <hostname.host-os>: <mode>`

Example:

```
supervisor@rtbrick>LEAF01: op>
```

1.2. Switching different modes

The example below shows how to switch between modes.

```
supervisor@rtbrick>LEAF01: op> switch-mode
config           Enter a given mode
debug            Enter a given mode
operation        Enter a given mode
```

The example below shows how to switch from the **operation** mode to the **config** mode.

```
supervisor@rtbrick>LEAF01: op> switch-mode config
supervisor@rtbrick>LEAF01: cfg>
```

1.3. Committing CLI configurations

The example below shows how to commit your changes.

```
supervisor@rtbrick>LEAF01:~$ cli
supervisor@rtbrick>LEAF01: op> switch-mode config
supervisor@rtbrick>LEAF01: cfg> <cli command goes here>
supervisor@rtbrick>LEAF01: cfg> commit
```

When you leave the CLI configuration mode with uncommitted changes, you see a reminder that you may have forgotten to commit, as shown in the following example.

```
supervisor@rtbrick>LEAF01: cfg> exit
Uncommitted changes are present
1. Discard the changes and exit
2. Commit the changes and exit
3. Keep the changes and exit [Default behavior]
Enter one of the above choice to proceed :
```

1.4. Adding a Configuration Description

You can add an in-line description or comment in a system configuration to include a description about that configuration.

set system config-description <description>

Example:

```
supervisor@rtbrick>LEAF01: cfg> set system config-description "This is sample
test configuration"
supervisor@rtbrick>LEAF01: cfg> commit
supervisor@rtbrick>LEAF01: cfg> show config
{
  "ietf-restconf:data": {
    "rtbrick-config:system": {
      "config-description": "This is sample test configuration"
    }
  }
}
```

1.5. Viewing configuration

The example below shows how to view configuration.

```
supervisor@rtbrick>LEAF01: cfg> show config
```

1.6. Viewing uncommitted changes

To view the uncommitted changes, enter the following command:

```
supervisor@rtbrick>LEAF01: cfg> show diff
```

1.7. Saving configuration

To save configuration, enter the following command:

```
supervisor@rtbrick>LEAF01: cfg> save config my_config.json
```



- Ensure that you use .json at the end of the filename.
- The configuration will be saved to the current working directory of CLI executable.

1.8. Loading configuration

To load configuration, enter the following command:

load config <filename> <option>

The options include 'merge' and 'replace'. You can specify 'merge' after the file name to merge the configuration with the running configuration. Specify 'replace' to replace the running configuration with the new one. Without any option, it replaces the running configuration, by default.

```
supervisor@rtbrick>LEAF01: cfg> load config <filename>
```



- Ensure that you use '.json' at the end of the filename.
- Remember to commit your changes after loading.

1.9. Viewing configuration in set format

To view configuration in a set format, enter the following command:

```
supervisor@rtbrick>LEAF01: cfg> show config set
```

1.10. Viewing configuration in a given hierarchy

To view configuration in a given hierarchy, enter the following command:

```
supervisor@rtbrick>LEAF01: cfg> show config instance red address-family
```

1.11. Turning Paging on/off

To turn the paging on or off, enter the following command:

paging [on | off]

- off - Pagination will be turned off for the commands that span more than screen length
- on - Pagination will be turned on for the commands that span more than screen length

```
supervisor@rtbrick>LEAF01: cfg> paging on
```

1.12. Discarding the uncommitted diff

To discard the uncommitted diff, enter the following command:

```
supervisor@rtbrick>LEAF01: cfg> discard
```

1.13. Deleting entire running-config at a time

To delete the entire running-configuration at a time, enter the following command:

```
supervisor@rtbrick>LEAF01: cfg> discard all
```

1.14. Rollback to previous committed configurations

You can rollback to a previously committed configuration file if you need to revert to a previous configuration. RBFS saves the last 50 committed configurations, including the rollback ID, commit ID, and time.

1.14.1. Viewing commit log

To view the commit log, enter the following command:

```
show commit
```

```
supervisor@rtbrick>LEAF01: op> show commit
rollback ID          Commit-ID
Timestamp
0                    29d5db038c1920fdsdsdsdsdsd323232    2021-01-
28T09:44:25.175048+0000
1                    29d8998a8c1920fdsdsdsdsdsd323232    2021-01-
26T05:44:25.175048+0000
2                    59d5db038c1920fdsdsdsdsdsd323232    2021-01-
22T09:40:25.175048+0000
49                   29d5d988999q20fdsdsdsdsdsd323232    2021-01-
2T01:44:25.175048+0000
```

1.14.2. Viewing the configuration differences in the current and previous versions

In RBFS, you can view the configuration differences between the current and the previous versions.

```
show diff <number>
```

```
supervisor@rtbrick>LEAF01: cfg> show diff 2
system {
-   secure-management-status false;
+   secure-management-status true;
}
supervisor@rtbrick>LEAF01: cfg>
```

1.14.3. Rollback to a previously committed configuration

To rollback to a specific configuration prior to the most recently committed one,

enter the following command:

```
rollback <number>
```

<number> - Specifies the rollback ID. Range: 1 through 49. 0 refers to the active configuration, 1 to the most recent previous configuration. Default: 1

For example, to rollback to rollback ID 2, enter following command:

```
supervisor@rtbrick>LEAF01: cfg> rollback 2
```

1.14.4. Rollback to a specific version of the software configuration

To rollback to a specific version of the software configuration, enter the following command:

```
rollback commit-id <commit-hash>
```

Example:

```
supervisor@rtbrick>LEAF01: cfg> rollback commit-id  
29d5db038c1920fdsdsdsdsdsd323232
```

1.15. Displaying Configurations in a Specific Format

The 'show config' command displays the current committed configurations of the system. By default, this command displays the configurations in a 'json' format.

```
show config <format>
```

You can also specify the format explicitly, if needed. The available display formats are:

- **json**: Display configurations in JSON format
- **set**: Display configurations in CLI format (same as commands executed)
- **netconf**: Display configurations in XML format
- **text**: Display configurations in textual format (similar to YANG definition)

The following example shows how configurations are displayed in text format.

```
supervisor@rtbrick>LEAF01: op> show config text
daemon-options {
  instance-name *;
  afi *;
  safi *;
  bd-type bgp.appd;
  bd-name bgp.appd.1;
}
daemon-options {
  instance-name *;
  afi *;
  safi *;
  bd-type bgp.iod;
  bd-name bgp.iod.1;
}
interface {
  name lo-0/0/0;
  unit {
    unit-id 0;
    address {
      ipv4 {
        prefix4 192.0.0.3/32;
      }
      ipv6 {
        prefix6 192:0:0::3/128;
      }
    }
  }
}
interface {
  name lo-0/0/1;
  unit {
    unit-id 0;
    instance ip2vrf;
    address {
      ipv4 {
        prefix4 192.0.1.3/32;
      }
      ipv6 {
        prefix6 192:0:1::3/128;
      }
    }
  }
}
<...>
```

1.16. Displaying Core Files

You can use the `show cores` command to show a set of system core files created when the device service has been crashed. This command is used for diagnostic purposes.

Example:

```
supervisor@rtbrick>LEAF01: op> show cores
Date      Time      Filename
May 10    09:02    core.lldpd.20220510-090237.1317.zst
May 10    09:02    core.igmp.iod.1.20220510-090228.1282.zst
May 10    09:02    core.pim.iod.1.20220510-090228.1280.zst
May 10    09:01    core.lldpd.20220510-090145.984.zst
May 10    09:01    core.igmp.iod.1.20220510-090140.991.zst
May 10    09:01    core.pim.iod.1.20220510-090140.989.zst
```

1.17. Displaying Date And Time

To display system date and time, use the **date** command.

date

Example:

```
supervisor@rtbrick>LEAF01: op> date
Thu Apr 28 09:56:32 UTC 2022
```

1.18. Rebooting Containers and Hosts

The **reboot** command allows you to restart containers and hosts.

reboot <option>

Option	Description
-	Without any option, this command allows you to reboot a container (default). You are prompted to confirm rebooting the container when you enter this command. You must answer yes or no.
container	This command allows you to reboot a container. You are prompted to confirm rebooting the container when you enter this command. You must answer yes or no.
container-and-confirm	This command reboots the container without prompting yes/no.
device	This command allows you to reboot a device. You are prompted to confirm rebooting the device when you enter this command. You must answer yes or no.
device-and-confirm	This command reboots the device without prompting yes/no.

1.19. Displaying Command History

The `history` command enables you to view the previously executed commands. You can execute the command in any of the CLI modes.

history

Example:

```
supervisor@rtbrick>LEAF01: op> history
show config set
exit
show config set
load config test.json
load config obj.json
show config set
exit
show config set
load config test.json
switch-mode config
load config test.json
load config obj.json
exit
switch-mode config
show config set
load config test.josn
load config obj
load config obj.json
exit
show config set
load config obj.json
load config test.json
exit
show bd running-status
load config test.json
show config set
exit
show bd running-status
show co
show cores
exi
show datastore confd table test index index2
exit
show bd running-status
exi
switch-mode config
exit
set forwarding-options prefix-list test ipv4-prefix 1.1.1.1/24
commit
show config
exit
```

1.20. 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.