CONFIGURATION GUIDE

Brocade Vyatta Network OS RIP Configuration Guide, 5.2R1

Supporting Brocade vRouter, VNF Platform, and Distributed Services Platform Deployments

53-1004738-01
24 October 2016

Notice: This document is for informational purposes only and does not set forth any warranty, expressed or implied, concerning any equipment, equipment feature, or service offered or to be offered by Brocade. Brocade reserves the right to make changes to this document at any time, without notice, and assumes no responsibility for its use. This informational document describes features that may not be currently available. Contact a Brocade sales office for information on feature and product availability. Export of technical data contained in this document may require an export license from the United States government.

The authors and Brocade Communications Systems, Inc. assume no liability or responsibility to any person or entity with respect to the accuracy of this document or any loss, cost, liability, or damages arising from the information contained herein or the computer programs that accompany it.

Contents

Preface

Document conventions

The document conventions describe text formatting conventions, command syntax conventions, and important notice formats used in Brocade technical documentation.

Notes, cautions, and warnings

Notes, cautions, and warning statements may be used in this document. They are listed in the order of increasing severity of potential hazards.

NOTE
A Note provides a tip, guidance, or advice, emphasizes important information, or provides a reference to related information.

ATTENTION
An Attention statement indicates a stronger note, for example, to alert you when traffic might be interrupted or the device might reboot.

CAUTION
A Caution statement alerts you to situations that can be potentially hazardous to you or cause damage to hardware, firmware, software, or data.

DANGER
A Danger statement indicates conditions or situations that can be potentially lethal or extremely hazardous to you. Safety labels are also attached directly to products to warn of these conditions or situations.

Text formatting conventions

Text formatting conventions such as boldface, italic, or Courier font may be used to highlight specific words or phrases.

Format	Description
bold text	Identifies command names. Identifies keywords and operands. Identifies the names of GUI elements. Identifies text to enter in the GUI.
italic text	Identifies emphasis. Identifies variables. Identifies document titles.
`Courier font`	Identifies CLI output.

Format	Description
	Identifies command syntax examples.

Command syntax conventions

Bold and italic text identify command syntax components. Delimiters and operators define groupings of parameters and their logical relationships.

Convention	Description
bold text	Identifies command names, keywords, and command options.
italic text	Identifies a variable.
value	In Fibre Channel products, a fixed value provided as input to a command option is printed in plain text, for example, **--show** WWN.
[ ]	Syntax components displayed within square brackets are optional. Default responses to system prompts are enclosed in square brackets.
{ **x** | **y** | **z** }	A choice of required parameters is enclosed in curly brackets separated by vertical bars. You must select one of the options. In Fibre Channel products, square brackets may be used instead for this purpose.
x | **y**	A vertical bar separates mutually exclusive elements.
< >	Nonprinting characters, for example, passwords, are enclosed in angle brackets.
...	Repeat the previous element, for example, *member*[*member*...].
\	Indicates a “soft” line break in command examples. If a backslash separates two lines of a command input, enter the entire command at the prompt without the backslash.

Brocade resources

Visit the Brocade website to locate related documentation for your product and additional Brocade resources.

White papers, data sheets, and the most recent versions of Brocade software and hardware manuals are available at www.brocade.com. Product documentation for all supported releases is available to registered users at MyBrocade.
Click the **Support** tab and select **Document Library** to access documentation on MyBrocade or www.brocade.com You can locate documentation by product or by operating system.

Release notes are bundled with software downloads on MyBrocade. Links to software downloads are available on the MyBrocade landing page and in the Document Library.

Document feedback

Quality is our first concern at Brocade, and we have made every effort to ensure the accuracy and completeness of this document. However, if you find an error or an omission, or you think that a topic needs further development, we want to hear from you. You can provide feedback in two ways:

- Through the online feedback form in the HTML documents posted on www.brocade.com
- By sending your feedback to documentation@brocade.com

Provide the publication title, part number, and as much detail as possible, including the topic heading and page number if applicable, as well as your suggestions for improvement.

Contacting Brocade Technical Support

As a Brocade customer, you can contact Brocade Technical Support 24x7 online, by telephone, or by e-mail. Brocade OEM customers should contact their OEM/solution provider.

Brocade customers

For product support information and the latest information on contacting the Technical Assistance Center, go to www.brocade.com and select **Support**.

If you have purchased Brocade product support directly from Brocade, use one of the following methods to contact the Brocade Technical Assistance Center 24x7.

Online	Telephone	E-mail
Preferred method of contact for non-urgent issues: • Case management through the MyBrocade portal. • Quick Access links to Knowledge Base, Community, Document Library, Software Downloads and Licensing tools	Required for Sev 1-Critical and Sev 2-High issues: • Continental US: 1-800-752-8061 • Europe, Middle East, Africa, and Asia Pacific: +800-AT FIBREE (+800 28 34 27 33) • Toll-free numbers are available in many countries. • For areas unable to access a toll-free number: +1-408-333-6061	support@brocade.com Please include: • Problem summary • Serial number • Installation details • Environment description

Brocade OEM customers

If you have purchased Brocade product support from a Brocade OEM/solution provider, contact your OEM/solution provider for all of your product support needs.

- OEM/solution providers are trained and certified by Brocade to support Brocade® products.
- Brocade provides backline support for issues that cannot be resolved by the OEM/solution provider.
- Brocade Supplemental Support augments your existing OEM support contract, providing direct access to Brocade expertise. For more information, contact Brocade or your OEM.
- For questions regarding service levels and response times, contact your OEM/solution provider.

About This Guide

This guide describes how to configure Routing Information Protocol (RIP) on the Brocade 5600 vRouter (referred to as a virtual router, vRouter, or router in the guide).

RIP Configuration

RIP overview

RIP is a dynamic routing protocol suitable for small, homogeneous networks. It is classified as an interior gateway protocol and employs the distance-vector routing algorithm. RIP determines the best path by counting the hops to the destination. The maximum hop count is 15 (16 is considered an infinite distance), making RIP less suitable for large networks. RIP is considered obsoleted by Open Shortest Path First (OSPF).

Supported standards

The Brocade vRouter implementation of RIP complies with the following standards:

- RFC 1058: Routing Information Protocol
- RFC 2453: RIP Version 2

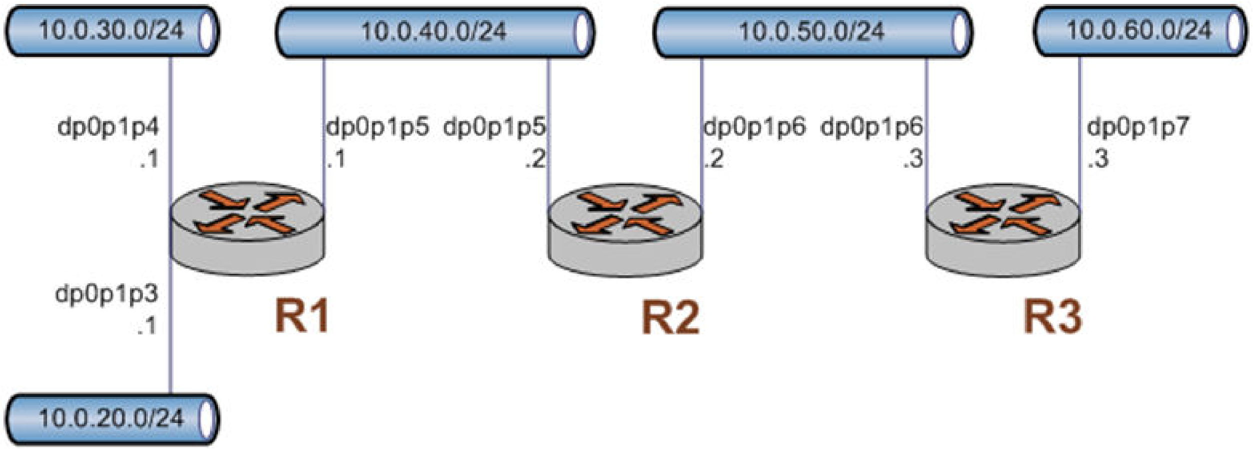
Configuring RIP

This section presents the following topics:

- Basic RIP configuration
- Verifying the RIP configuration

This section presents a sample configuration of RIP. The RIP configuration in Basic RIP configuration on page 12 is based on the diagram in the following figure.

FIGURE 1 Sample RIP configuration



Basic RIP configuration

In this section, you configure RIP on the routers that are labeled R1, R2, and R3 in the sample configuration in Configuring RIP on page 11. The routers are advertising their routes on the 10.0.40.0/24 and 10.0.50.0/24 networks.

It is assumed for this example that you have already configured the router interfaces; only the steps required to implement RIP are shown.

To create a basic RIP configuration, perform the following steps in configuration mode:

TABLE 1 Basic RIP configuration

Router	Step	Command
R1	Advertise to the 10.0.40.0/24 network.	`vyatta@R1# set protocols rip network 10.0.40.0/24`
R1	Redistribute connected routes to RIP.	`vyatta@R1# set protocols rip redistribute connected`
R1	Commit the configuration.	`vyatta@R1# commit`

R1	Display the configuration.	`vyatta@R1# show protocols`
		` rip {`
		`    network 10.0.40.0/24`
		`    redistribute {`
		`        connected {`
		`        }`
		`    }`
		` }`

R2	Advertise to the 10.0.40.0/24 network.	`vyatta@R2# set protocols rip network 10.0.40.0/24`
R2	Advertise to the 10.0.50.0/24 network.	`vyatta@R2# set protocols rip network 10.0.50.0/24`
R2	Redistribute connected routes to RIP.	`vyatta@R2# set protocols rip redistribute connected`
R2	Commit the configuration.	`vyatta@R2# commit`

R2	Display the configuration.	`vyatta@R2# show protocols`
		` rip {`
		`    network 10.0.40.0/24`
		`    network 10.0.50.0/24`
		`    redistribute {`
		`        connected {`
		`        }`
		`    }`
		` }`

R3	Advertise to the 10.0.50.0/24 network.	`vyatta@R3# set protocols rip network 10.0.50.0/24`
R3	Redistribute connected routes to RIP.	`vyatta@R3# set protocols rip redistribute connected`
R3	Commit the configuration.	`vyatta@R3# commit`

TABLE 1 Basic RIP configuration (continued)

Router	Step	Command

R3	Display the configuration.	vyatta@R3# show protocols
		 rip {
		    network 10.0.50.0/24
		    redistribute {
		        connected {
		        }
		    }
		 }

Verifying the RIP configuration

The following operational mode commands verify the RIP configuration.

show ip route

The **show ip route** command shows how to verify RIP on the R3 router.

```
vyatta@R3:~$ show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route
R>* 10.0.20.0/24 [120/3] via 10.0.50.2, dp0p1p6, 00:20:16
R>* 10.0.30.0/24 [120/3] via 10.0.50.2, dp0p1p6, 00:34:04
R>* 10.0.40.0/24 [120/2] via 10.0.50.2, dp0p1p6, 02:15:26
C>* 10.0.50.0/24 is directly connected, dp0p1p6
C>* 10.0.60.0/24 is directly connected, dp0p1p7
C>* 127.0.0.0/8 is directly connected, lo
vyatta@R3:~$
```

The output shows that routes to the 10.0.20.0/24, 10.0.30.0/24, and 10.0.40.0/24 networks have been learned through RIP and that packets to those networks are forwarded out dp0p1p6 to 10.0.50.2. The 10.0.50.0/24 and 10.0.60.0/24 networks are directly connected.

show ip rip

show ip rip on page 39 for R3 displays similar RI verification information in a different format.

```
vyatta@R3:~$ show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface
     Network            Next Hop         Metric From            Tag Time
R(n) 10.0.20.0/24       10.0.50.2             3 10.0.50.2         0 00:23
R(n) 10.0.30.0/24       10.0.50.2             3 10.0.50.2         0 00:23
R(n) 10.0.40.0/24       10.0.50.2             2 10.0.50.2         0 00:23
C(i) 10.0.50.0/24       0.0.0.0               1 self              0
C(r) 10.0.60.0/24       0.0.0.0               1 self (connected:1)  0
vyatta@R3:~$
```

Again, the output shows that routes to 10.0.20.0/24, 10.0.30.0/24, and 10.0.40.0/24 have been learned through RIP and that packets to those networks are forwarded to 10.0.50.2. The 10.0.50.0/24 and 10.0.60.0/24 networks are directly connected.

ping 10.0.20.1

Using the **ping** command from the R3 router, you can confirm that hosts on remote networks can be reached. In this case we ping an IP address on R1. `ping 10.0.20.1` shows how to ping an IP address on the R1 router.

```
vyatta@R3:~$ ping 10.0.20.1
PING 10.0.20.1 (10.0.20.1) 56(84) bytes of data.
64 bytes from 10.0.20.1: icmp_seq=1 ttl=63 time=7.39 ms
64 bytes from 10.0.20.1: icmp_seq=2 ttl=63 time=1.56 ms
64 bytes from 10.0.20.1: icmp_seq=3 ttl=63 time=1.49 ms
^C
--- 10.0.20.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 1.497/3.482/7.390/2.763 ms
vyatta@R3:~$
```

This output confirms that the RIP configuration is working and that a remote network can be reached.

Router-level Configuration Commands

monitor protocol rip disable events

Disables the generation of debug messages that are related to RIP events.

Syntax

monitor protocol rip disable events

Modes

Operational mode

Usage Guidelines

Use this command to disable the generation of debug (trace-level) messages that are related to RIP events.

monitor protocol rip disable packet

Disables the generation of debug messages that are related to all types of RIP packets.

Syntax

monitor protocol rip disable packet [**recv** | **send**]

Parameters

recv

Optional. Disables debugging on all received packets.

send

Optional. Disables debugging on all sent packets.

Modes

Operational mode

Usage Guidelines

Use this command to disable the generation of debug (trace-level) messages that are related to all types of RIP packets.

monitor protocol rip disable rib

Disables the generation of debug messages that are related to the RIP Routing Information Base (RIB).

Syntax

monitor protocol rip disable rib

Command Default

Debug messages are disabled for actions that are related to the RIP RIB.

Modes

Operational mode

Usage Guidelines

Use this command to disable the generation of debug (trace-level) messages that are related to the RIP RIB.

monitor protocol rip enable events

Enables the generation of debug messages that are related to RIP events.

Syntax

monitor protocol rip enable events

Modes

Operational mode

Usage Guidelines

Use this command to enable the generation of debug (trace-level) messages that are related to RIP events.

monitor protocol rip enable packet

Enables the generation of debug messages that are related to all types of RIP packets.

Syntax

monitor protocol rip enable packet [**recv** | **send**]

Parameters

recv

Optional. Enables debugging on all received packets.

send

Optional. Enables debugging on all sent packets.

Modes

Operational mode

Usage Guidelines

Use this command to enable the generation of debug (trace-level) messages that are related to all types of RIP packets.

monitor protocol rip enable rib

Enables the generation of debug messages that are related to the RIP Routing Information Base (RIB).

Syntax

monitor protocol rip enable rib

Command Default

Debug messages are generated for actions related to the RIP RIB.

Modes

Operational mode

Usage Guidelines

Use this command to enable the generation of debug (trace-level) messages that are related to the RIP RIB.

protocols rip default-distance <distance>

Sets the default administrative distance for RIP.

Syntax

set protocols rip default-distance *distance*

delete protocols rip default-distance

show protocols rip default-distance

Command Default

The default administrative distance is 120.

Parameters

distance

Mandatory. The default administrative distance. The distance ranges from 1 through 255. The default distance is 120.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        default-distance distance
    }
}
```

Usage Guidelines

Use the **set** form of this command to set the default administrative distance for RIP.

Use the **delete** form of this command to restore the default administrative distance for RIP, which is 120.

Use the **show** form of this command to display the default administrative distance for RIP.

protocols rip default-information originate

Generates a default route to the RIP routing domain.

Syntax

set protocols rip default-information originate

delete protocols rip default-information originate

show protocols rip default-information originate

Command Default

By default, the system does not generate a default route.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        default-information {
            originate
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to generate a default route to the RIP routing domain.

Use the **delete** form of this command to restore the default behavior for default route generation to the RIP routing domain, that is, the system does not generate a route.

Use the **show** form of this command to display the default route generation to the RIP routing domain.

protocols rip default-metric <metric>

Changes the default metric for routes that are redistributed to RIP.

Syntax

set protocols rip default-metric *metric*

delete protocols rip default-metric

show protocols rip default-metric

Command Default

Routes that are redistributed to RIP are assigned a metric of 1.

Parameters

metric

Mandatory. A metric that is assigned to routes. The metric ranges from 1 through 16. The default metric is 1.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        default-metric metric
    }
}
```

Usage Guidelines

Use the **set** form of this command to change the metric for routes that are redistributed to RIP.

Use the **delete** form of this command to restore the default metric to 1 for routes that are redistributed to RIP.

Use the **show** form of this command to display the default metric for routes that are redistributed to RIP.

protocols rip interface <interface>

Enables RIP on an interface.

Syntax

set protocols rip interface *interface*

delete protocols rip interface *interface*

show protocols rip interface *interface*

Parameters

interface

The identifier of an interface. Supported interface types are:

- Data plane
- Loopback

For more information about these interface types, refer to Supported Interface Types on page 61.

You can enable RIP on more than one interface by creating multiple **protocols rip interface** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        interface interface
    }
}
```

Usage Guidelines

Use the **set** form of this command to enable RIP on an interface. The interface must be enabled for RIP before you can use it for RIP routing.

Use the **delete** form of this command to disable RIP on an interface.

Use the **show** form of this command to display RIP configuration on an interface.

protocols rip log

Enables logging for RIP.

Syntax

set protocols rip log { **all** | **events**| **nsm** | **packet**| **rib**}

delete protocols rip log { **all** | **events**| **nsm**| **packet** | **rib**}

show protocols rip log { **all** | **events**| **nsm** | **packet**| **rib**}

Command Default

None

Parameters

all

Enables all RIP logs.

events

Enables only RIP events logs.

nsm

Enables only RIP NSM logs.

packet

Enables only RIP packet logs.

rib

Enables only RIP RIB logs.

Modes

Configuration mode

Configuration Statement

```
protocols {
        rip {
                log {
                        all
                        events
                        nsm
                        packet
                        rib
                }
        }
}
```

Usage Guidelines

Use the **set** form of this command to enable routing information protocol (RIP) logs.

Use the **delete** form of this command to remove RIP logs.

Use the **show** form of this command to view RIP logs.

protocols rip log packet

Enables logging for RIP packets.

Syntax

set protocols rip log packet { **all** | **detail**| **rcv** | **send** }

delete protocols rip log packet { **all** | **detail**| **rcv** | **send** }

show protocols rip log packet { **all** | **detail**| **rcv** | **send** }

Command Default

None

Parameters

all

Enables all RIP packet logs.

detail

Enables only RIP packet detail logs.

rcv

Enables only RIP packet receive logs.

send

Enables only RIP packet send logs.

Modes

Configuration mode

Configuration Statement

```
protocols {
       rip {
              log {
                     packet {
                            all
                            detail
                            rcv
                            send
                     }
              }
       }
}
```

Usage Guidelines

Use the **set** form of this command to enable routing information protocol (RIP) packet logs.

Use the **delete** form of this command to remove RIP packet logs.

Use the **show** form of this command to view RIP packet logs.

protocols rip neighbor <ipv4>

Defines a RIP neighbor router.

Syntax

set protocols rip neighbor *ipv4*

delete protocols rip neighbor *ipv4*

show protocols rip neighbor

Parameters

ipv4

The IP address of a neighbor router.
You can define more than one RIP neighbor router by creating multiple **protocols rip neighbor** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        neighbor ipv4
    }
}
```

Usage Guidelines

Use the **set** form of this command to define a RIP neighbor router.

Use the **delete** form of this command to remove a RIP neighbor router.

Use the **show** form of this command to display the configuration of RIP neighbor routers.

protocols rip network <ipv4net>

Specifies a network for RIP.

Syntax

set protocols rip network *ipv4net*

delete protocols rip network *ipv4net*

show protocols rip network

Parameters

ipv4net

Mandatory. Multi-node. The IP network address of a RIP network.

You can identify more than one RIP network by creating multiple **protocols rip network** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        network ipv4net
    }
}
```

Usage Guidelines

Use the **set** form of this command to specify a RIP network.

Use the **delete** form of this command to remove a RIP network.

Use the **show** form of this command to display RIP network configuration.

protocols rip network-distance <ipv4net>

Establishes the administrative distance for or applies an access list to a RIP network.

Syntax

set protocols rip network-distance *ipv4net* { **access-list** *list-name* | **distance** *distance* }

delete protocols rip network-distance *ipv4net* [**access-list** *list-name* | **distance** *distance*]

show protocols rip network-distance *ipv4net* [**access-list** | **distance**]

Parameters

ipv4net

Mandatory. The IP address of a network.

access-list

A defined access list for the specified network.

distance

An administrative distance for the network. The distance ranges from 1 through 255. The default distance is 120.

Modes

Configuration mode

Configuration Statement

```
protocols {
   rip {
      network-distance ipv4net {
         access-list list-name
         distance distance
      }
   }
}
```

Usage Guidelines

Use the **set** form of this command to establish the administrative distance for or apply an access list to a RIP network.

The administrative distance indicates the trustworthiness of a router or group of routers as a source of routing information. In general, the higher the value, the less trusted the entity. An administrative distance of 1 usually represents a directly connected network, and an administrative distance of 255 means the routing source is unreliable or unknown. The administrative distance conventionally applied to RIP is 120.

Use the **delete** form of this command to restore the default administrative distance, which is 120, to a RIP network or remove an access list.

Use the **show** form of this command to display the administrative distance of a RIP network or the application of an access list.

protocols rip passive-interface <interface>

Suppresses RIP routing updates on an interface.

Syntax

set protocols rip passive-interface *interface*

delete protocols rip passive-interface *interface*

show protocols rip passive-interface

Command Default

RIP routing updates are not suppressed.

Parameters

interface

The identifier of an interface. Supported interface types are:

- Data plane
- Loopback

For more information about these interface types, refer to Supported Interface Types on page 61.

You can suppress routing updates on more than one RIP interface by creating multiple **protocols rip passive-interface** configuration nodes.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        passive-interface interface
    }
}
```

Usage Guidelines

Use the **set** form of this command to suppress RIP routing updates on an interface.

Use the **delete** form of this command to disable the suppression of RIP routing updates on an interface.

Use the **show** form of this command to display RIP route suppression configuration for an interface.

protocols rip route <ipv4net>

Defines a RIP static route.

Syntax

set protocols rip route *ipv4net*

delete protocols rip route *ipv4net*

show protocols rip route

Parameters

ipv4net

Mandatory. The network address of a RIP static route.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        route ipv4net
    }
}
```

Usage Guidelines

Use the **set** form of this command to define a RIP static route.

Use the **delete** form of this command to remove a RIP static route.

Use the **show** form of this command to display the configuration of RIP static routes.

protocols rip timers garbage-collection <seconds>

Sets a timer for RIP garbage collection.

Syntax

set protocols rip timers garbage-collection *seconds*

delete protocols rip timers garbage-collection [*seconds*]

show protocols rip timers garbage-collection

Command Default

RIP garbage collection occurs at 120 seconds.

Parameters

seconds

Mandatory. An interval in seconds. The number of seconds ranges from 5 through 2147483647.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        timers {
            garbage-collection seconds
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to set a timer for RIP garbage collection. When the timer expires, the system scans for stale RIP resources and releases them for use.

Use the **delete** form of this command to restore the default interval, which is 120 seconds, for the RIP garbage collection timer.

Use the **show** form of this command to display the RIP garbage collection timer.

protocols rip timers timeout <seconds>

Sets an interval for RIP time-outs.

Syntax

set protocols rip timers timeout *seconds*

delete protocols rip timers timeout [*seconds*]

show protocols rip timers timeout

Command Default

RIP time-outs occur at 180 seconds.

Parameters

seconds

Mandatory. An interval in seconds. The number of seconds ranges from 5 through 2147483647. The default number of seconds is 180.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        timers {
            timeout seconds
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to set an interval for RIP time-outs.

Use the **delete** form of this command to restore the default interval, which is 180 seconds, for RIP time-outs.

Use the **show** form of this command to display the RIP time-out interval.

protocols rip timers update <seconds>

Sets a timer for updates to the RIP routing table.

Syntax

set protocols rip timers update *seconds*

delete protocols rip timers update [*seconds*]

show protocols rip timers update

Command Default

The RIP routing table is updated every 30 seconds.

Parameters

seconds

Mandatory. An interval in seconds. The number of seconds ranges from 5 through 2147483647. The default number of seconds is 30.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        timers {
            update seconds
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to set a timer for updates to the RIP routing table. A shorter interval means more accurate routing information in the table; however, more protocol network traffic occurs.

Use the **delete** form of this command to restore the default interval, which is 30 seconds, for updates to the RIP routing table.

Use the **show** form of this command to display the interval for updates to the RIP routing table.

reset ip rip route

Resets data in the RIP routing table.

Syntax

reset ip rip [**statistics** | **route** [**all** | **bgp** | **connected** | **kernel** | **ospf** | **rip** | **static** | *ip-address*]]

Parameters

all

Removes all entries from the RIP routing table.

bgp

Removes only BGP routes from the RIP routing table.

connected

Removes entries for connected routes from the RIP routing table.

kernel

Removes kernel entries from the RIP routing table.

ospf

Removes only OSPF routes from the RIP routing table.

rip

Removes only RIP routes from the RIP routing table.

static

Removes static entries from the RIP routing table.

ip-address

Removes entries that match *ip-address (x.x.x.x/x)*, a destination IP address, from the RIP routing table.

statistics

Resets the RIP statistics.

Modes

Operational mode.

Usage Guidelines

Use the **reset ip rip route all** command to clear the RIP routing table.

show ip rip

Displays information for the Routing Information Protocol (RIP).

Syntax

show ip rip [**status**]

Command Default

Displays all RIP protocol information.

Parameters

status

Optional. Displays only protocol status.

Modes

Operational mode

Usage Guidelines

Use this command to display protocol information for RIP.

Examples

The following example shows how to display protocol information for RIP.

```
vyatta@vyatta:~$ show ip rip
Codes: R - RIP, C - connected, S - Static, O - OSPF, B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r) - redistribute,
      (i) - interface
     Network            Next Hop         Metric From            Tag Time
C(i) 192.168.1.0/24     0.0.0.0               1 self              0
vyatta@vyatta:~$
```

show ip route rip

Displays all IP RIP routes that are contained in the Routing Information Base (RIB).

Syntax

show ip route rip

Modes

Operational mode

Usage Guidelines

Use this command to display all RIP routes that are contained in the RIB.

Examples

The following example shows how to display all RIP routes that are contained in the RIB.

```
vyatta@vyatta:~$ show ip route rip
R       19.1.1.0/24 [120/1] is directly connected, dp0p192p1, 00:01:04
vyatta@vyatta:~$
```

show monitoring protocols rip

Displays RIP protocol debugging flags.

Syntax

show monitoring protocols rip

Modes

Operational mode

Usage Guidelines

Use this command to see how debugging is set for RIP.

Route Redistribution Commands

protocols rip redistribute bgp

Redistributes Border Gateway Protocol (BGP) routes into RIP routing tables.

Syntax

set protocols rip redistribute bgp [**metric** *metric* | **route-map** *map-name*]

delete protocols rip redistribute bgp [**metric** | **route-map**]

show protocols rip redistribute bgp [**metric** | **route-map**]

Command Default

BGP routes that are redistributed into RIP routing tables are assigned a routing metric of 1. By default, no route map is applied to redistributed BGP routes.

Parameters

metric

A routing metric. The metric ranges from 1 through 16. The default metric is 1.

map-name

Optional. A route map.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        redistribute {
            bgp {
                metric metric
                route-map map-name
            }
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to set the routing metric for BGP routes being redistributed into RIP, or to specify a route map to be applied to redistributed BGP routes.

Use the **delete** form of this command to remove BGP route redistribution configuration.

Use the **show** form of this command to display BGP route redistribution configuration.

protocols rip redistribute connected

Redistributes directly connected routes into RIP routing tables.

Syntax

set protocols rip redistribute connected [**metric** *metric* | **route-map** *map-name*]

delete protocols rip redistribute connected [**metric** | **route-map**]

show protocols rip redistribute connected [**metric** | **route-map**]

Command Default

Connected routes that are redistributed into RIP are assigned a routing metric of 1. By default, no route map is applied to redistributed connected routes.

Parameters

metric

Optional. A routing metric. The metric ranges from 1 through 16. The default metric is 1.

map-name

Optional. A route map.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        redistribute {
            connected {
                metric metric
                route-map map-name
            }
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to set the routing metric for connected routes being redistributed into RIP, or to specify a route map to be applied to redistributed connected routes.

Use the **delete** form of this command to remove connected route redistribution configuration.

Use the **show** form of this command to display connected route redistribution configuration.

protocols rip redistribute kernel

Redistributes kernel routes into RIP routing tables.

Syntax

set protocols rip redistribute kernel [**metric** *metric* | **route-map** *map-name*]

delete protocols rip redistribute kernel [**metric** | **route-map**]

show protocols rip redistribute kernel [**metric** | **route-map**]

Command Default

Kernel routes that are redistributed into RIP are assigned a routing metric of 1. By default, no route map is applied to redistributed kernel routes.

Parameters

metric

Optional. A routing metric. The metric ranges from 1 through 16. The default metric is 1.

map-name

Optional. A route map.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        redistribute {
            kernel {
                metric metric
                route-map map-name
            }
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to set the routing metric for kernel routes being redistributed into RIP, or to specify a route map to be applied to redistributed kernel routes.

Use the **delete** form of this command to remove kernel route redistribution configuration.

Use the **show** form of this command to display kernel route redistribution configuration.

protocols rip redistribute ospf

Redistributes (OSPF) routes into RIP routing tables.

Syntax

set protocols rip redistribute ospf [**metric** *metric* | **route-map** *map-name*]

delete protocols rip redistribute ospf [**metric** | **route-map**]

show protocols rip redistribute ospf [**metric** | **route-map**]

Command Default

OSPF routes that are redistributed into RIP are assigned a routing metric of 1. By default, no route map is applied to redistributed OSPF routes.

Parameters

metric

Optional. A routing metric. The metric ranges from 1 through 16. The default metric is 1.

map-name

Optional. A route map.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        redistribute {
            ospf {
                metric metric
                route-map map-name
            }
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to set the routing metric for OSPF routes being redistributed into RIP, or to specify a route map to be applied to redistributed OSPF routes.

Use the **delete** form of this command to remove OSPF route redistribution configuration.

Use the **show** form of this command to display OSPF route redistribution configuration.

protocols rip redistribute static

Redistributes static routes into RIP routing tables.

Syntax

set protocols rip redistribute static [**metric** *metric* | **route-map** *map-name*]

delete protocols rip redistribute static [**metric** | **route-map**]

show protocols rip redistribute static [**metric** | **route-map**]

Command Default

Static routes that are redistributed into RIP are assigned a routing metric of 1. By default, no route map is applied to redistributed static routes.

Parameters

metric

Optional. A routing metric. The metric ranges from 1 through 16. The default metric is 1.

map-name

Optional. A route map.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        redistribute {
            static {
                metric metric
                route-map map-name
            }
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to set the routing metric for static routes being redistributed into RIP, or to specify a route map to be applied to redistributed static routes.

Use the **delete** form of this command to remove static route redistribution configuration.

Use the **show** form of this command to display static route redistribution configuration.

Route Filtering Commands

protocols rip distribute-list access-list

Applies an access list to filter inbound or outbound RIP packets.

Syntax

set protocols rip distribute-list access-list { **in** *in-list* | **out** *out-list* }

delete protocols rip distribute-list access-list { **in** | **out** }

show protocols rip distribute-list access-list { **in** | **out** }

Parameters

in *in-list* | **out** *out-list*

in-list : The identifier of a defined access list. The access list is applied to filter inbound RIP packets.

out-list :The identifier of a defined access list. The access list is applied to filter outbound RIP packets.

The number of the access list that is used to filter networks in routing updates. The number ranges are as follows:

1-99: IP standard access list.

100-199: IP extended access list.

1300-1999: IP standard access list (expanded range).

2000-2699: IP extended access list (expanded range).

Modes

Configuration mode

Configuration Statement

```
protocols
    rip {
        distribute-list {
            access-list {
                in in-list
                out out-list
            }
        }
    }
```

Usage Guidelines

Use the **set** form of this command to apply an access list to filter inbound or outbound RIP packets.

Use the **delete** form of this command to remove filtering of RIP packets by access list.

Use the **show** form of this command to display the configuration for filtering of RIP packets by access list.

protocols rip distribute-list interface <interface> access-list

Applies an access list to an interface to filter inbound or outbound RIP packets.

Syntax

set protocols rip distribute-list interface *interface* **access-list** { **in** *in-list* | **out** *out-list*]

delete protocols rip distribute-list interface *interface* **access-list** { **in** | **out** }

show protocols rip distribute-list interface *interface* **access-list** { **in** | **out** }

Parameters

interface

The identifier of an interface. Supported interface types are:

- Data plane
- Loopback

For more information about these interface types, refer to Supported Interface Types on page 61.

in-list

The identifier of a defined access list. The access list is applied to the interface to filter inbound RIP packets.

out-list

The identifier of a defined access list. The access list is applied to the interface to filter outbound RIP packets.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        distribute-list {
            interface interface {
                access-list {
                    in in-list
                    out out-list
                }
            }
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to apply an access list to an interface to filter inbound or outbound RIP packets.

Use the **delete** form of this command to remove filtering of RIP packets by access list from an interface.

Use the **show** form of this command to display the configuration for filtering of RIP packets by access list for an interface.

protocols rip distribute-list interface <interface> prefix-list

Applies a prefix list to an interface to filter inbound or outbound RIP packets.

Syntax

set protocols rip distribute-list interface *interface* **prefix-list** { **in** *in-list* | **out** *out-list* }

delete protocols rip distribute-list interface *interface* **prefix-list** { **in** | **out** }

show protocols rip distribute-list interface *interface***prefix-list** { **in** | **out** }

Parameters

interface

The identifier of an interface. Supported interface types are:

- Data plane
- Loopback

For more information about these interface types, refer to Supported Interface Types on page 61.

in-list

The identifier of a defined prefix list. The prefix list is applied to the interface to filter inbound RIP packets.

out-list

The identifier of a defined prefix list. The prefix list is applied to the interface to filter outbound RIP packets.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        distribute-list {
            interface interface {
                prefix-list {
                    in in-list
                    out out-list
                }
            }
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to apply a prefix list to an interface to filter inbound or outbound RIP packets.

Use the **delete** form of this command to remove filtering of RIP packets by prefix list from an interface.

Use the **show** form of this command to display the configuration for filtering of RIP packets by prefix list for an interface.

protocols rip distribute-list prefix-list

Applies a prefix list to filter inbound or outbound RIP packets.

Syntax

set protocols rip distribute-list prefix-list { **in** *in-list* | **out** *out-list* }

delete protocols rip distribute-list prefix-list { **in** | **out** }

show protocols rip distribute-list prefix-list { **in** | **out** }

Parameters

in-list

The identifier of a defined prefix list. The prefix list is applied to filter inbound RIP packets.

out-list

The identifier of a defined prefix list. The prefix list is applied to filter outbound RIP packets.

Modes

Configuration mode

Configuration Statement

```
protocols {
    rip {
        distribute-list {
            prefix-list {
                in in-list
                out out-list
            }
        }
    }
}
```

Usage Guidelines

Use the **set** form of this command to apply a prefix list to filter inbound or outbound RIP packets.

Use the **delete** form of this command to remove filtering of RIP packets by prefix list.

Use the **show** form of this command to display the configuration for filtering of RIP packets by prefix list.

RIP Interface Commands

interfaces <interface> ip rip

Enables RIP on an interface.

Syntax

set interfaces *interface* **ip rip**

delete interfaces *interface* **ip rip**

show interfaces *interface* **ip rip**

Parameters

interface

Mandatory. A type of interface. For detailed keywords and arguments that can be specified as an interface, refer to Supported Interface Types on page 61.

Modes

Configuration mode

Configuration Statement

```
interfaces interface {
    ip {
        rip
    }
}
```

Usage Guidelines

Use this command to enable RIP on an interface.

Use the **set** form of this command to enable RIP on an interface.

Use the **delete** form of this command to remove all RIP configuration and disable RIP on an interface.

Use the **show** form of this command to display RIP configuration on an interface.

interfaces <interface> ip rip authentication

Establishes an authentication method to be used for RIP on an interface.

Syntax

set interfaces *interface* **ip rip authentication** [**md5** *md5-key* **password** *md5-password* | **plaintext-password** *password*]

delete interfaces *interface* **ip rip authentication** [**md5** *md5-key* **password** | **plaintext-password**]

show interfaces *interface* **ip rip authentication** [**md5** *md5-key* **password** | **plaintext-password**]

Parameters

interface

Mandatory. A type of interface. For detailed keywords and arguments that can be specified as an interface, refer to Supported Interface Types on page 61.

md5-key

Optional. An authentication key. This key must be the same on both the sending and receiving systems. The key ranges from 1 through 255.

md5-password

Optional. A password to use in MD5 authentication. This password must be the same on both the sending and receiving systems.

password

Optional. A password to use in simple (plain text) authentication. This password must be the same on both the sending and receiving systems.

Modes

Configuration mode

Configuration Statement

```
interfaces interface {
    ip {
        rip {
            authentication {
                md5 md5-key {
                    password md5-password
                }
                plaintext-password password
            }
        }
    }
}
```

Usage Guidelines

Use this command to establish an authentication method to be used for RIP on an interface. This authentication is independent of the authentication configured for the RIP area.

In plain text authentication, passwords are sent through the network in plain text. In MD5 authentication, the system uses the Message Digest 5 (MD5) algorithm to compute a hash value from the contents of the RIP packet and the password. The hash value and the MD5 key are included in the transmitted packet, and the receiving system (configured with the same password) calculates its own hash function, which must match.

The authentication parameters must be the same for all routers that are to establish two-way communication within a network. If two routers do not agree on these parameters, they do not consider adjacencies, and disregard communication from each other.

Use the **set** form of this command to specify an authentication method to be used for RIP on an interface.

Use the **delete** form of this command to remove an authentication method to be used for RIP from an interface.

Use the **show** form of this command to display an authentication method to be used for RIP on an interface.

interfaces <interface> ip rip split-horizon

Enables split-horizon or split-horizon poison-reverse on an interface that is running RIP.

Syntax

set interfaces *interface* **ip rip split-horizon** [**disable** | **poison-reverse**]

delete interfaces *interface* **ip rip split-horizon** [**disable** | **poison-reverse**]

show interfaces *interface* **ip rip split-horizon**

Command Default

Split-horizon and split-horizon poison-reverse are disabled.

Parameters

interface

Mandatory. A type of interface. For detailed keywords and arguments that can be specified as an interface, refer to Supported Interface Types on page 61.

disable

Disables split-horizon on the interface.

poison-reverse

Enables split-horizon poison-reverse on the interface.

Modes

Configuration mode

Configuration Statement

```
interfaces interface {
    ip {
        rip {
            split-horizon {
                disable
                poison-reverse
            }
        }
    }
}
```

Usage Guidelines

Use this command to enable split-horizon or split-horizon poison-reverse on an interface that is running RIP.

Split-horizon is a stability feature that reduces the possibility of network loops, particularly when links become disconnected. It stops an interface from including in its network updates of any routes that it learned from that interface. Split-horizon is effective at preventing loops between routers that are directly connected to each another and speeds convergence when network conditions change; it is the default setting in RIP.

Poison-reverse is a variation of split-horizon. When an interface that has poison-reverse enabled detects that a link is down, it increases the metric for that route to 16 and propagates that information in its next update. Because 15 is the largest number of hops that are considered reachable on a RIP network, increasing the metric to 16 renders the route unreachable as far as downstream RIP routers are concerned. This is called “poisoning” the route. Poison-reverse can be used to propagate information about bad routes to routers that are downstream but not immediate neighbors, where split-horizon is ineffective.

When this option is enabled, the router includes the route in announcements to the neighbor from which it was learned. When this option is disabled, the router omits the route in announcements to the neighbor from which it was learned.

Use the **set** form of this command to configure split-horizon and split-horizon poison-reverse on an interface that is running RIP.

Use the **delete** form of this command to restore the default configuration, that is, split-horizon and split-horizon poison-reverse are disabled.

Use the **show** form of this command to display whether split-horizon and split-horizon poison-reverse are enabled or disabled.

Supported Interface Types

The following table shows the syntax and parameters of supported interface types. Depending on the command, some of these types may not apply.

Interface Type	Syntax	Parameters
Bridge	**bridge** *brx*	*brx*: The name of a bridge group. The name ranges from br0 through br999.
Data plane	**dataplane** *interface-name*	*interface-name*: The name of a data plane interface. Following are the supported formats of the interface name: • **dp***x***p***y***p***z*—The name of a data plane interface, where — **dp***x* specifies the data plane identifier (ID). Currently, only dp0 is supported. — p*y* specifies a physical or virtual PCI slot index (for example, p129). — **p***z* specifies a port index (for example, p1). For example, dp0p1p2, dp0p160p1, and dp0p192p1. • **dp***x***em***y* —The name of a data plane interface on a LAN-on-motherboard (LOM) device that does not have a PCI slot, where **em***y* specifies an embedded network interface number (typically, a small number). For example, dp0em3. • **dp***x***s***y* —The name of a data plane interface on a device that is installed on a virtual PCI slot, where *x***s***y* specifies an embedded network interface number (typically, a small number). For example, dp0s2. • **dp***x***P***n***p***y***p***z* —The name of a data plane interface on a device that is installed on a secondary PCI bus, where **P***n* specifies the bus number. You can use this format to name data plane interfaces on large physical devices with multiple PCI buses. For these devices, it is possible to have network interface cards installed on different buses with these cards having the same slot ID. The value of *n* must be an integer greater than 0. For example, dp0P1p162p1 and dp0P2p162p1.
Data plane vif	**dataplane** *interface-name* **vif** *vif-id* [**vlan** *vlan-id*]	*interface-name*: Refer to the preceding description. *vif-id*: A virtual interface ID. The ID ranges from 1 through 4094. *vlan-id*: The VLAN ID of a virtual interface. The ID ranges from 1 through 4094.
Loopback	**loopback lo** or **loopback lo***n*	*n*: The name of a loopback interface, where *n* ranges from 1 through 99999.
OpenVPN	**openvpn** *vtunx*	*vtunx*: The identifier of an OpenVPN interface. The identifier ranges from vtun0 through vtun*x*, where *x* is a nonnegative integer.
Tunnel	**tunnel** *tunx* or **tunnel** *tunx* **parameters**	*tunx*: The identifier of a tunnel interface you are defining. The identifier ranges from tun0 through tun*x*, where *x* is a nonnegative integer.
Virtual tunnel	**vti** *vtix*	*vtix*: The identifier of a virtual tunnel interface you are defining. The identifier ranges from vti0 through vti*x*, where *x* is a nonnegative integer.

Interface Type	Syntax	Parameters
		Note: This interface does not support IPv6.
VRRP	*parent-interface* **vrrp vrrp-group** *group*	*parent-interface*: The type and identifier of a parent interface; for example, data plane dp0p1p2 or bridge br999. *group*: A VRRP group identifier. The name of a VRRP interface is not specified. The system internally constructs the interface name from the parent interface identifier plus the VRRP group number; for example, dp0p1p2v99. Note that VRRP interfaces support the same feature set as does the parent interface.

VRF Support

VRF support for RIP and RIPng

This section describes VRF support for RIP and RIPng configuration- and operational-mode commands. This section also describes VRF support for monitoring and logging commands.

VRF support for router-mode commands

You can run RIP and RIPng router-mode configuration commands in the context of a routing instance by using the optional **routing routing-instance** *instance-name* keywords and variable. The following examples show how to configure RIP and RIPng in the context of the RED routing instance.

```
routing routing-instance RED protocols rip …
routing routing-instance RED protocols ripng …
```

If you do not specify a routing instance, the vRouter applies the configuration to the default routing instance.

NOTE
An interface belongs to only one routing instance.

VRF support for interface-mode commands

The RIP and RIPng interface-mode configuration commands do not support the **routing routing-instance** *instance-name* keywords and variable because these commands run in the context of the routing instance to which the interfaces belong.

```
interfaces <intf_type> <intf_name> ip rip …
interfaces <intf_type> <intf_name> ipv6 ripng …
```

VRF support for operational commands

You can use the optional **routing-instance** *instance-name* keyword and variable with the RIP and RIPng operational commands. If you do not use this optional keyword and variable, the commands run in the context of the default routing instance.

```
show ip rip [routing-instance <instance_name>] …
reset ip rip [routing-instance <instance_name>] route …
show ipv6 ripng [routing-instance <instance_name>] …
reset ipv6 ripng [routing-instance <interface_name>] route …
```

VRF support for monitoring and logging commands

You can run the RIP and RIPng monitoring and logging commands in the context of a routing instance with the exception of the commands that enable RIB and NSM logging. If you do not use the **routing-instance** *instance-name* keyword and variable, the commands run in the context of the default routing instance.

```
monitor protocol rip [routing-instance <instance_name>]…
[routing routing-instance <instance_name>] protocols rip log …
```

```
monitor protocol ripng [routing-instance <instance_name>] …
[routing routing-instance <instance_name>] protocols ripng log …
```

The **rib** and **nsm** logging options are global options and apply to all routing instances. The **rib** and **nsm** logging options cannot be enabled or disabled on a routing instance basis. The following commands apply to all routing instances.

```
monitor protocol rip … nsm
monitor protocol rip … rib
protocols rip log nsm
protocols rip log rib
monitor protocol ripng … nsm
monitor protocol ripng … rib
protocols ripng log nsm
protocols ripng log rib
```

The output of the following commands displays routing instance information, if relevant.

```
show monitoring protocols rip
show monitoring protocols ripng
```

Command support for VRF routing instances

VRF allows a Brocade 5600 vRouter to support multiple routing tables, one for each VRF routing instance. Some commands in this guide support VRF and can be applied to particular routing instances.

Use the guidelines in this section to determine correct syntax when adding VRF routing instances to commands. For more information about VRF, refer to *Brocade Vyatta Network OS Basic Routing Configuration Guide*. This guide includes an overview of VRF, VRF configuration examples, information about VRF-specific features, and a list of commands that support VRF routing instances.

Adding a VRF routing instance to a Configuration mode command

For most Configuration mode commands, specify the VRF routing instance at the beginning of a command. Add the appropriate VRF keywords and variable to follow the initial action (**set**, **show**, or **delete**) and before the other keywords and variables in the command.

Configuration mode example: syslog

The following command configures the syslog logging level for the specified syslog host. The command does not include a VRF routing instance, so the command applies to the default routing instance.

```
vyatta@R1# set system syslog host 10.10.10.1 facility all level debug
vyatta@R1# show system syslog
syslog {
    host 10.10.10.1 {
            facility all {
                    level debug
            }
    }
}
```

The following example shows the same command with the VRF routing instance (GREEN) added. Notice that **routing routing-instance GREEN** has been inserted between the basic action (**set** in the example) and the rest of the command. Most Configuration mode commands follow this convention.

```
vyatta@R1# set routing routing-instance GREEN system syslog host 10.10.10.1 facility all level debug
vyatta@R1# show routing
routing {
     routing-instance GREEN {
             system {
                     syslog {
                            host 11.12.13.2:514 {
                                     facility all {
                                             level debug
                                     }
                             }
                     }
             }
     }
}
```

Configuration mode example: SNMP

Some features, such as SNMP, are not available on a per-routing instance basis but can be bound to a specific routing instance. For these features, the command syntax is an exception to the convention of specifying the routing instance at the beginning of Configuration mode commands.

The following example shows how to configure the SNMPv1 or SNMPv2c community and context for the RED and BLUE routing instances. The first two commands specify the RED routing instance as the context for community A and BLUE routing instance as the context for community B. The subsequent commands complete the configuration.

For more information about configuring SNMP, refer to *Brocade Vyatta Network OS Remote Management Configuration Guide*.

```
vyatta@R1# set service snmp community commA context RED
vyatta@R1# set service snmp community commB context BLUE
vyatta@R1# set service snmp view all oid 1
vyatta@R1# set service snmp community commA view all
vyatta@R1# set service snmp community commB view all
vyatta@R1# show service snmp community
 community commA {
        context RED
        view all
 }
 community commB {
        context BLUE
        view all
 }
[edit]
vyatta@vyatta#
```

Adding a VRF routing instance to an Operational mode command

The syntax for adding a VRF routing instance to an Operational mode command varies according to the type of command parameters:

- If the command does not have optional parameters, specify the routing instance at the end of the command.
- If the command has optional parameters, specify the routing instance after the required parameters and before the optional parameters.

Operational mode examples without optional parameters

The following command displays dynamic DNS information for the default routing instance.

```
vyatta@vyatta:~$ show dns dynamic status
```

The following command displays the same information for the specified routing instance (GREEN). The command does not have any optional parameters, so the routing instance is specified at the end of the command.

```
vyatta@vyatta:~$ show dns dynamic status routing-instance GREEN
```

Operational mode example with optional parameters

The following command obtains multicast path information for the specified host (10.33.2.5). A routing instance is not specified, so the command applies to the default routing instance.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 detail
```

The following command obtains multicast path information for the specified host (10.33.2.5) and routing instance (GREEN). Notice that the routing instance is specified before the optional **detail** keyword.

```
vyatta@vyatta:~$ mtrace 10.33.2.5 routing-instance GREEN detail
```

Operational mode example output: SNMP

The following SNMP **show** commands display output for routing instances.

```
vyatta@vyatta:~$ show snmp routing-instance
Routing Instance SNMP Agent is Listening on for Incoming Requests:
Routing-Instance            RDID
-----------------           ----
RED                         5

vyatta@vyatta:~$ show snmp community-mapping
SNMPv1/v2c Community/Context Mapping:
Community                   Context
---------                   -------
commA                       'RED'
commB                       'BLUE'
deva                        'default'


vyatta@vyatta:~$ show snmp trap-target
SNMPv1/v2c Trap-targets:
Trap-target                   Port   Routing-Instance Community
-----------                   ----   ---------------- ---------
1.1.1.1                              'RED'           'test'


vyatta@vyatta:~$ show snmp v3 trap-target
SNMPv3 Trap-targets:
Trap-target                   Port   Protocol Auth Priv Type   EngineID              Routing-Instance User
-----------                   ----   -------- ---- ---- ----   --------              ---------------- ----
2.2.2.2                       '162'  'udp'    'md5      'infor                       'BLUE'           'test'
```

List of Acronyms

Acronym	Description
ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
AH	Authentication Header
AMI	Amazon Machine Image
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
AWS	Amazon Web Services
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
CCMP	AES in counter mode with CBC-MAC
CHAP	Challenge Handshake Authentication Protocol
CLI	command-line interface
DDNS	dynamic DNS
DHCP	Dynamic Host Configuration Protocol
DHCPv6	Dynamic Host Configuration Protocol version 6
DLCI	data-link connection identifier
DMI	desktop management interface
DMVPN	dynamic multipoint VPN
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System
DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
EBS	Amazon Elastic Block Storage
EC2	Amazon Elastic Compute Cloud
EGP	Exterior Gateway Protocol
ECMP	equal-cost multipath
ESP	Encapsulating Security Payload
FIB	Forwarding Information Base
FTP	File Transfer Protocol
GRE	Generic Routing Encapsulation
HDLC	High-Level Data Link Control
I/O	Input/Output
ICMP	Internet Control Message Protocol

Acronym	Description
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocol
IPS	Intrusion Protection System
IKE	Internet Key Exchange
IP	Internet Protocol
IPOA	IP over ATM
IPsec	IP Security
IPv4	IP Version 4
IPv6	IP Version 6
ISAKMP	Internet Security Association and Key Management Protocol
ISM	Internet Standard Multicast
ISP	Internet Service Provider
KVM	Kernel-Based Virtual Machine
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol
LLDP	Link Layer Discovery Protocol
MAC	medium access control
mGRE	multipoint GRE
MIB	Management Information Base
MLD	Multicast Listener Discovery
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
NBMA	Non-Broadcast Multi-Access
ND	Neighbor Discovery
NHRP	Next Hop Resolution Protocol
NIC	network interface card
NTP	Network Time Protocol
OSPF	Open Shortest Path First
OSPFv2	OSPF Version 2
OSPFv3	OSPF Version 3
PAM	Pluggable Authentication Module
PAP	Password Authentication Protocol
PAT	Port Address Translation
PCI	peripheral component interconnect
PIM	Protocol Independent Multicast
PIM-DM	PIM Dense Mode

Acronym	Description
PIM-SM	PIM Sparse Mode
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PTMU	Path Maximum Transfer Unit
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RHEL	Red Hat Enterprise Linux
RIB	Routing Information Base
RIP	Routing Information Protocol
RIPng	RIP next generation
RP	Rendezvous Point
RPF	Reverse Path Forwarding
RSA	Rivest, Shamir, and Adleman
Rx	receive
S3	Amazon Simple Storage Service
SLAAC	Stateless Address Auto-Configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SPT	Shortest Path Tree
SSH	Secure Shell
SSID	Service Set Identifier
SSM	Source-Specific Multicast
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TBF	Token Bucket Filter
TCP	Transmission Control Protocol
TKIP	Temporal Key Integrity Protocol
ToS	Type of Service
TSS	TCP Maximum Segment Size
Tx	transmit
UDP	User Datagram Protocol
VHD	virtual hard disk
vif	virtual interface
VLAN	virtual LAN
VPC	Amazon virtual private cloud
VPN	virtual private network
VRRP	Virtual Router Redundancy Protocol

Acronym	Description
WAN	wide area network
WAP	wireless access point
WPA	Wired Protected Access