

# Commands: debug m through debug z

---

## COMMAND DESCRIPTION

## **Copyright**

© Ericsson AB 2009–2011. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

## **Disclaimer**

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

## **Trademark List**

**SmartEdge** is a registered trademark of Telefonaktiebolaget LM Ericsson.

**NetOp** is a trademark of Telefonaktiebolaget LM Ericsson.



# Contents

<b>1</b>	<b>Command Descriptions</b>	<b>1</b>
1.1	debug mcast	2
1.2	debug mobile-ip	4
1.3	debug mobile-ip agent-common	6
1.4	debug mobile-ip authentication	8
1.5	debug mobile-ip foreign-agent	10
1.6	debug mobile-ip home-agent	12
1.7	debug mobile-ip interaction	14
1.8	debug mobile-ip packet	16
1.9	debug mobile-ip packet-io	19
1.10	debug mpls-static	21
1.11	debug msdp	22
1.12	debug nat	25
1.13	debug nd	27
1.14	debug ntp	30
1.15	debug ospf	33
1.16	debug ospf bfd	35
1.17	debug ospf flooding	37
1.18	debug ospf interface	39
1.19	debug ospf ldp-synchronization	41
1.20	debug ospf lsdb	42
1.21	debug ospf neighbor	45
1.22	debug ospf nsr	47
1.23	debug ospf packet	49
1.24	debug ospf rcm	52
1.25	debug ospf redistribution	54
1.26	debug ospf rib	56
1.27	debug ospf spf	58
1.28	debug packet	61
1.29	debug pedgr	66
1.30	debug pim	68
1.31	debug pim bfd	70



1.32	debug pim mcastmgr	72
1.33	debug pim packet	74
1.34	debug pim ppa	76
1.35	debug pim rp-mapping	78
1.36	debug pm	80
1.37	debug policy access-list	82
1.38	debug policy general	84
1.39	debug ppp	86
1.40	debug ppp multilink	89
1.41	debug pppoe	91
1.42	debug qos	94
1.43	debug radius	96
1.44	debug rcm	98
1.45	debug rip	100
1.46	debug route-map	103
1.47	debug rsvp	105
1.48	debug rsvp bfd	109
1.49	debug shmlib	110
1.50	debug snmp	112
1.51	debug spanning-tree	116
1.52	debug ssh	119
1.53	debug standby ospf nsr	121
1.54	debug static	123
1.55	debug static dvsr	125
1.56	debug subscriber	127
1.57	debug sysmon ftp	129
1.58	debug talk	131
1.59	debug tunnel client	133
1.60	debug vpls	135
1.61	debug vrrp	137
	<b>Glossary</b>	<b>141</b>



# 1 Command Descriptions

Commands starting with “debug m” through commands starting “debug z” are included.

This document applies to both the Ericsson SmartEdge® and SM family routers. However, the software that applies to the SM family of systems is a subset of the SmartEdge OS; some of the functionality described in this document may not apply to SM family routers.

For information specific to the SM family chassis, including line cards, refer to the SM family chassis documentation.

For specific information about the differences between the SmartEdge and SM family routers, refer to the Technical Product Description *SM Family of Systems* (part number 5/221 02-CRA 119 1170/1) in the **Product Overview** folder of this Customer Product Information library.



## 1.1 debug mcast

```
debug mcast {all | group | ppa}
```

```
no debug mcast {all | group | ppa}
```

### 1.1.1 Purpose

Enables the generation of debug messages for IP multicast manager events.

### 1.1.2 Command Mode

exec (10)

### 1.1.3 Syntax Description

<code>all</code>	Enables the generation of debug messages for all IP multicast manager functions
<code>group</code>	Enables the generation of debug messages for IGMP functions.
<code>ppa</code>	Enables the generation of debug messages related to forwarding messages to the Packet Forwarding Engine (PFE).

### 1.1.4 Default

The generation of debug messages for IP multicast manager events is disabled.

### 1.1.5 Usage Guidelines

Use the `debug mcast` command to enable the generation of debug messages for IP multicast manager events.

Use the `no` form of this command to disable the generation of debug messages for IP multicast manager events.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---

---



To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

### 1.1.6 Examples

The following example shows how to enable the generation of debug messages for all IP multicast manager events:

```
[local]Redback#debug mcast all
```



## 1.2 debug mobile-ip

```
debug [{boot {active | standby} | switchover}] mobile-ip [all | event-types]
```

```
no debug [{boot {active | standby} | switchover}] mobile-ip [all | event-types]
```

### 1.2.1 Purpose

Enables the generation of debug messages for Mobile IP events.

### 1.2.2 Command Mode

exec (10)

### 1.2.3 Syntax Description

<b>boot</b>	Optional. Limits the generation of debug messages to system reload events.
<b>active</b>	Enables the generation of debug messages to active controller card events.
<b>standby</b>	Enables the generation of debug messages to standby controller card events.
<b>switchover</b>	Optional. Limits the generation of debug messages to during a switchover from the active to the standby controller.
<b>all</b>	Enables the generation of debug messages for all Mobile IP events.
<b>event types</b>	Optional. Limits the generation of debug messages to the following Mobile IP event types: <ul style="list-style-type: none"> <li>• agent-common—Limits the generation of debug messages to events common to both the HA and FA.</li> <li>• authentication—Limits the generation of debug messages for Mobile IP to authentication events.</li> <li>• foreign-agent—Limits the generation of debug messages for the FA instance events.</li> <li>• home-agent—Limits the generation of debug messages to HA instance events.</li> <li>• interaction—Limits the generation of debug messages for Mobile IP module interaction events, such as Router Configuration Manager (RCM) and Interface and Circuit State Manager (ISM) events.</li> <li>• packet—Limits the generation of debug messages for Mobile IP packets. This is a filtered debugging feature for specific source, destination, circuit, or packet types.</li> <li>• packet-io—Limits the generation of debug messages on I/O packets events on a kernel socket interface for Mobile IP services.</li> </ul>

### 1.2.4 Default

The generation of debug messages for Mobile IP events is disabled.



## 1.2.5 Usage Guidelines

Use the `debug mobile-ip` command to enable the generation of debug messages for Mobile IP events.

Use the `no` form of this command to disable the generation of debug messages for the specified type of Mobile IP events.

**Note:** The SmartEdge 100 router does not support the `standby` and `switchover` keywords.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

## 1.2.6 Examples

The following example show how to enable the generation of debug messages for all Mobile IP events:

```
[local]Redback#debug mobile-ip all
```



## 1.3 debug mobile-ip agent-common

```
debug mobile-ip agent-common [instance | interface | internal |  
shared-memory | tunnel]
```

```
no debug mobile-ip agent-common
```

### 1.3.1 Purpose

Enables the generation of debug messages that are common to both the home-agent (HA) instance and foreign-agent (FA) instances.

### 1.3.2 Command Mode

exec (10)

### 1.3.3 Syntax Description

<code>instance</code>	Optional. Limits the generation of debug messages to Mobile IP instance events common to the HA and FA.
<code>interface</code>	Optional. Limits the generation of debug messages to Mobile IP access interface events.
<code>internal</code>	Optional. Limits the generation of debug messages to Mobile IP internal queue, time, and thread events.
<code>shared-memory</code>	Optional. Limits the generation of debug messages to Mobile IP shared memory events.
<code>tunnel</code>	Optional. Limits the generation of debug messages to Mobile IP tunnel creation and state events.

### 1.3.4 Default

The generation of debug messages for HA instance and FA instance events is disabled.

### 1.3.5 Usage Guidelines

Use the `debug mobile-ip agent-common` command to enable the generation of debug messages for HA and FA instances. When you use this command without specifying an option, the system generates debug messages for all FA and HA events.

Use the `no` form of this command to disable the generation of debug messages for HA instance and FA instance events.



---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

### 1.3.6 Examples

The following example shows how to limit the generation of debug messages to Mobile IP interface events:

```
[local]Redback#debug mobile-ip agent-common interface
```



## 1.4 debug mobile-ip authentication

```
debug mobile-ip authentication [aaa | fa-ha | mn-ha]
```

```
no debug mobile-ip authentication
```

### 1.4.1 Purpose

Enables the generation of debug messages for authentication and security in Mobile IP.

### 1.4.2 Command Mode

exec (10)

### 1.4.3 Syntax Description

<code>aaa</code>	Optional. Limits the generation of debugging messages to authentication, authorization, and accounting (AAA) interface to Mobile IP.
<code>fa-ha</code>	Optional. Limits the generation of debugging messages between foreign-agent (FA) and home-agent (HA) authentication and security events.
<code>mn-ha</code>	Optional. Limits the generation of debugging messages between mobile node (MN) and HA authentication and security events.

### 1.4.4 Default

The generation of debug messages for Mobile IP authentication is disabled.

### 1.4.5 Usage Guidelines

Use the `debug mobile-ip authentication` command to enable the generation of debug messages for Mobile IP authentication. When you use this command without specifying an option, the system generates debug messages for all Mobile IP authentication events.

Use the `no` form of this command to disable the generation of debug messages for Mobile IP authentication.



---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

### 1.4.6 Examples

The following example shows how to limit the generation of debug messages to AAA events:

```
[local]Redback#debug mobile-ip authentication aaa
```



## 1.5 debug mobile-ip foreign-agent

```
debug mobile-ip foreign-agent [visitor | core | home-agent-peer  
| care-of-address]
```

```
no debug mobile-ip foreign-agent
```

### 1.5.1 Purpose

Enables the generation of debug messages for the foreign-agent (FA) instance.

### 1.5.2 Command Mode

exec (10)

### 1.5.3 Syntax Description

<code>visitor</code>	Optional. Limits the generation of debug messages to visitor events on this FA instance.
<code>core</code>	Optional. Limits the generation of debug messages to core FA instance events.
<code>home-agent-peer</code>	Optional. Limits the generation of debug messages to home-agent (HA) peer events of this FA instance.
<code>care-of-address</code>	Optional. Limits the generation of debug messages to care-of-address events of this FA instance.

### 1.5.4 Default

The generation of debug messages for the FA instance is disabled.

### 1.5.5 Usage Guidelines

Use the `debug mobile-ip foreign-agent` command to enable the generation of debug messages for the FA instance. When you use this command without specifying an option, the system generates debug messages for all FA events.

Use the `no` form of this command to disable the generation of debug messages for the FA.



---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

### 1.5.6 Examples

The following example shows how to enable the generation of debug messages for Mobile IP care-of address events:

```
[local]Redback#debug mobile-ip foreign-agent care-of-address
```



## 1.6 debug mobile-ip home-agent

```
debug mobile-ip home-agent [binding | core | foreign-agent-peer  
| local-address]
```

```
no debug mobile-ip home-agent
```

### 1.6.1 Purpose

Enables the generation of debug messages for home-agent (HA) instance.

### 1.6.2 Command Mode

exec (10)

### 1.6.3 Syntax Description

<code>binding</code>	Optional. Limits the generation of debug messages to bindings.
<code>core</code>	Optional. Limits the generation of debug messages to core HA instance events.
<code>foreign-agent-peer</code>	Optional. Limits the generation of debug messages to foreign-agent (FA) peer events of this HA instance.
<code>local-address</code>	Optional. Limits the generation of debug messages to local address events for this HA instance for Mobile IP.

### 1.6.4 Default

The generation of debug messages for Mobile IP HA events is disabled.

### 1.6.5 Usage Guidelines

Use the `debug mobile-ip home-agent` command to enable the generation of debug messages for HA instance events. When you use this command without specifying an option, the system generates debug messages for all HA instance events.

Use the `no` form of this command to disable the generation of debug messages for the HA instance.



---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

### 1.6.6 Examples

The following example shows how to limit the generation of debug messages to the FA peer:

```
[local]Redback#debug mobile-ip home-agent foreign-agent-peer
```



## 1.7 debug mobile-ip interaction

```
debug mobile-ip interaction [configuration | ism | rcm | route]
```

```
no debug mobile-ip interaction
```

### 1.7.1 Purpose

Enables the generation of debug messages for Mobile IP module interaction events with other SmartEdge modules, such as Router Configuration Manager (RCM) and Interface and Circuit State Manager (ISM).

### 1.7.2 Command Mode

exec (10)

### 1.7.3 Syntax Description

<code>configuration</code>	Optional. Limits the generation of debug messages to Mobile IP configuration events.
<code>ism</code>	Optional. Limits the generation of debug messages to Mobile IP interaction with the Interface State (ISM) module.
<code>rcm</code>	Optional. Limits the generation of debug messages to Mobile IP interaction with the Router Configuration Manager (RCM) module.
<code>route</code>	Optional. Limits the generation of debug messages to Mobile IP interaction with the Router Information Base (RIB) module.

### 1.7.4 Default

The generation of debug messages for external Mobile IP events is disabled.

### 1.7.5 Usage Guidelines

Use the `debug mobile-ip interaction` command to enable the generation of debug messages for Mobile IP events relating to interactions with modules, such as RCM and ISM. When you use this command without specifying an option, the system generates debug messages for all Mobile IP module interaction events.

Use the `no` form of this command to disable the generation of debug messages for external Mobile IP events.



---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

### 1.7.6 Examples

The following example shows how to limit the generation of debug messages to RCM events:

```
[local]Redback#debug mobile-ip interaction rcm
```



## 1.8 debug mobile-ip packet

```
debug [{boot {active | standby} | switchover}] mobile-ip packet
[errors] [hex-dump] [send] [receive] [type packet-type] [source
src-ip-addr] [destination dest-ip-addr] [circuit circuit-handle]
```

```
no debug [{boot {active | standby} | switchover}] mobile-ip packet
[errors] [hex-dump] [send] [receive] [type packet-type] [source
src-ip-addr] [destination dest-ip-addr] [circuit circuit-handle]
```

### 1.8.1 Purpose

Enables the generation of debug messages for Mobile IP packets. This is a filtered debugging feature for specific source, destination, circuit, or packet types.

### 1.8.2 Command Mode

exec (10)

### 1.8.3 Syntax Description

<b>boot</b>	Optional. Limits the generation of debug messages to system reload events.
<b>active</b>	Enables the generation of debug messages to active controller card events.
<b>standby</b>	Enables the generation of debug messages to standby controller card events.
<b>switchover</b>	Optional. Limits the generation of debug messages to events during a switchover from the active to the standby controller.
<b>errors</b>	Optional. Limits the generation of debug messages to Mobile IP packets with errors events.
<b>send</b>	Optional. Limits the generation of debug messages to transmitted Mobile IP packet events.
<b>receive</b>	Optional. Limits the generation of debug messages to received Mobile IP packet events.



<code>type packet-type</code>	<p>Optional. Type of packet for which the generation of debug messages is limited to, according to one of the following keywords:</p> <ul style="list-style-type: none"> <li>• <b>advertisement</b>—Limits the generation of debug messages to Mobile IP advertisement packets.</li> <li>• <b>reg-request</b>—Limits the generation of debug messages to Mobile IP registration-request packets.</li> <li>• <b>reg-reply</b>—Limits the generation of debug messages to Mobile IP registration-reply packets.</li> <li>• <b>reg-revocation</b>—Limits the generation of debug messages to Mobile IP registration-revocation packets.</li> <li>• <b>reg-revocation-ack</b>—Limits the generation of debug messages to Mobile IP registration-revocation acknowledgement packets.</li> <li>• <b>xid</b>—Limits the generation of debug messages to logical link control (LLC) exchange ID (XID) packets.</li> </ul>
<code>hex-dump</code>	Optional. Limits the generation of debug messages to a hexadecimal dump to the standard output.
<code>source src-ip-addr</code>	Optional. Limits the generation of debug messages to the source IP address.
<code>destination dest-ip-addr</code>	Optional. Limits the generation of debug messages to the destination IP address.
<code>circuit circuit-handle</code>	Optional. Limits the generation of debug messages to the circuit.

#### 1.8.4 Default

The generation of debug messages for Mobile IP packets is disabled.

#### 1.8.5 Usage Guidelines

Use the `debug mobile-ip packet` command to enable the generation of debug messages for Mobile IP packets. This is a filtered debugging feature for specific source, destination, circuit, or packet types. When you use this command without specifying an option, the system generates debug messages for all Mobile IP packet events.

Use the `no` form of this command to disable the generation of debug messages for the specified type of Mobile IP packets.



**Note:** The SmartEdge router does not support the `standby` and `switchover` keywords.

You can determine the value of the `circuit-handle` argument for the mobile node (MN) access interface or the care-of-address (CoA) interface using the `show mobile-ip` command or the `show mobile-ip home-agent-peer` command with the `detail` keyword.

---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

### 1.8.6 Examples

The following example shows how to limit the generation of debug messages to registration request (`reg-request`) packets in both directions:

```
[local]Redback#debug mobile-ip packet type reg-request
```



## 1.9 debug mobile-ip packet-io

```
debug mobile-ip packet-io [receive | send]
```

```
no debug mobile-ip packet-io
```

### 1.9.1 Purpose

Enables the generation of debug messages on I/O events on the kernel socket interface for Mobile IP services.

### 1.9.2 Command Mode

exec (10)

### 1.9.3 Syntax Description

<code>receive</code>	Optional. Limits the generation of debugging messages to input packet events on a kernel socket interface for Mobile IP services.
<code>send</code>	Optional. Limits the generation of debugging messages to output packets events on a kernel socket interface for Mobile IP services.

### 1.9.4 Default

The generation of debug messages for I/O packets events on a kernel socket interface for Mobile IP services is disabled.

### 1.9.5 Usage Guidelines

Use the `debug mobile-ip packet-io` command to enable the generation of debug messages for I/O packets on a kernel socket interface for Mobile IP services. When you use this command without specifying an option, the system generates debug messages for all I/O packet events on a kernel socket interface for Mobile IP services.

Use the `no` form of this command to disable the generation of debug messages on I/O packet events on a kernel interface for Mobile IP services.



---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

### 1.9.6 Examples

The following example shows how to limit the generation of debug messages to output packet events on a kernel socket interface:

```
[local]Redback#debug mobile-ip packet-io send
```



## 1.10 debug mpls-static

```
debug mpls-static [all | config | event | ism | label-action | lm |
lsp | next-hop]
```

```
no debug mpls-static [all | config | event | ism | label-action |
lm | lsp | next-hop]
```

### 1.10.1 Command Mode

exec (10)

### 1.10.2 Syntax Description

<code>all</code>	Enables the output of all static MPLS debug messages.
<code>config</code>	Limits the output to static MPLS configuration messages.
<code>event</code>	Limits the output to static MPLS general event messages.
<code>ism</code>	Limits the output to interaction between MPLS-STATIC manager and ISM.
<code>label-action</code>	Limits the output to label-mapping messages.
<code>lm</code>	Limits the output to label-management messages.
<code>lsp</code>	Limits the output to LSP messages.
<code>next-hop</code>	Limits the output to next-hop messages.

### 1.10.3 Default

None

### 1.10.4 Usage Guidelines

Use the `debug mpls-static` command to enable the generation of static MPLS debugging messages.

Disable the generation of debug messages with the `no` form of the command.

---



---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---



---



To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secured Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

### 1.10.5 Example

The following example shows how to enable the generation of static MPLS label-mapping messages:

```
[local]Redback#debug mpls-static label-mapping
```

## 1.11 debug msdp

```
debug [boot {active | standby} | switchover] msdp {all | peer  
[peer-addr] | sa-cache}
```

```
no debug [boot {active | standby} | switchover] msdp {all | peer  
[peer-addr] | sa-cache}
```

### 1.11.1 Purpose

Enables the generation of Multicast Source Discovery Protocol (MSDP) debug messages.

### 1.11.2 Command Mode

exec (10)

### 1.11.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.



<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<code>all</code>	Enables all MSDP debug messages.
<code>peer</code>	Enables MSDP peer debug messages.
<code>peer-addr</code>	Optional. IP address of the MSDP peer.
<code>sa-cache</code>	Enables MSDP source active (SA) cache debug messages.

#### 1.11.4 Default

The generation of debug messages is disabled.

#### 1.11.5 Usage Guidelines

Use the `debug msdp` command to enable the generation for all MSDP debug messages.

Use the `peer-addr` argument to enable the generation of MSDP debug messages for a specific MSDP peer.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---



---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---



---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secured Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.



Use the `no` form of this command to disable the generation of MSDP debug messages.

### 1.11.6 Examples

The following example shows how to enable the generation of MSDP debug messages for the peer address, `192.168.5.5`:

```
[local]Redback#debug msdp peer 192.168.5.5
```



## 1.12 debug nat

```
debug [{boot {active | standby} | switchover}] nat {aaa | all | cli |
config | ppa | rcm | work}
```

```
no debug [{boot {active | standby} | switchover}] nat {aaa | all | cli
| config | ppa | rcm | work}
```

### 1.12.1 Purpose

Enables the generation of Network Address Translation (NAT) debug messages.

### 1.12.2 Command Mode

exec (10)

### 1.12.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>aaa</b>	Enables the generation of debug messages for NAT authentication, authorization, and accounting (AAA) events.
<b>all</b>	Enables the generation of debug messages for all NAT events.
<b>cli</b>	Enables the generation of debug messages for NAT parser events.
<b>config</b>	Enables the generation of debug messages for NAT configuration events.
<b>ppa</b>	Enables the generation of debug messages for NAT Packet Processing ASIC (PPA) events.
<b>rcm</b>	Enables the generation of debug messages for NAT Router Configuration Manager (RCM) events.
<b>work</b>	Enables the generation of debug messages for NAT work orders.

### 1.12.4 Default

The generation of debug messages is disabled.



## 1.12.5 Usage Guidelines

Use the `debug nat` command to enable the generation of NAT debug messages.

**Note:** The SmartEdge 100 router does not support the `standby` and `switchover` keywords.

To store messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command in exec mode to display these stored messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of any debug messages on a production system.

---

---

Use the `no` form of this command to disable the generation of debug messages.

## 1.12.6 Examples

The following example shows how to enable the generation of NAT debug messages:

```
[local]Redback#debug nat
```



## 1.13 debug nd

```
debug [boot {active | standby} | switchover] nd {all | circuit |
config | dns | filter interface if-name | global | interface | ism |
lc | packet [packet-type] [detail] | rib | subscriber | timer}
```

```
no debug [boot {active | standby} | switchover] nd {all | circuit |
config | dns | filter interface if-name | global | interface | ism |
lc | packet [packet-type] [detail] | rib | subscriber | timer}
```

### 1.13.1 Purpose

Enables the generation of debug messages for Neighbor Discovery (ND) events.

### 1.13.2 Command Mode

exec (10)

### 1.13.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>all</b>	Optional. Enables the generation of debug messages for all ND events.
<b>circuit</b>	Enables the generation of debug messages for ND circuit events.
<b>config</b>	Enables the generation of debug messages for ND configuration events.
<b>dns</b>	Enables the generation of debug messages for ND Domain Name System (DNS) events.
<b>filter interface <i>if-name</i></b>	Name of the interface for which the generation of debug messages for ND interface events is enabled.



<b>global</b>	Optional. Enables the generation of debug messages for all ND packet, ISM, DNS, RIB, and timer events in all contexts.
<b>interface</b>	Enables the generation of debug messages for ND interface events.
<b>ism</b>	Enables the generation of debug messages for ND Interface and Circuit State Manager (ISM) events.
<b>lc</b>	Enables the generation of debug messages for ND traffic card events.
<b>packet</b>	Enables the generation of debug messages for ND packet events.
<b>packet-type</b>	Optional. ND packet type, according to one of the keywords listed in Table 1. If this argument is not specified, the <b>packet</b> keyword enables the generation of debug messages for all packet types.
<b>detail</b>	Optional. Includes an ND packet dump for ND packet events.
<b>rib</b>	Enables the generation of debug messages for ND Routing Information Base (RIB) events.
<b>subscriber</b>	Enables the generation of debug messages for events related to ND subscriber circuits and ND profiles.
<b>timer</b>	Enables the generation of debug messages for ND timer events.

#### 1.13.4 Default

The generation of debug messages for ND events is disabled.

#### 1.13.5 Usage Guidelines

Use the `debug nd` command to enable the generation of debug messages for ND events.

**Note:** The SmartEdge router does not support the `standby` and `switchover` keywords.




---



---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---



---

Table 1 lists the keywords supported by the `packet-type` argument.

*Table 1 ND Packet Types*

Keyword	Packet Type
<code>na</code>	ND neighbor advertisement packets
<code>ns</code>	ND neighbor solicitation packets
<code>ra</code>	ND neighbor router advertisement packets
<code>redirect</code>	ND neighbor redirect packets
<code>rs</code>	ND neighbor router solicitation packets

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command in exec mode to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages for ND events.

### 1.13.6 Examples

The following example shows how to enable the generation of debug messages for interface events for the `int1` interface:

```
[local]Redback#debug nd filter interface int1
```



## 1.14 debug ntp

```
debug [boot {active | standby} | switchover] ntp {all | general | packet | update}
```

```
no debug [boot {active | standby} | switchover] ntp {all | general | packet | update}
```

### 1.14.1 Purpose

Enables the generation of debug messages for the Network Time Protocol (NTP).

### 1.14.2 Command Mode

exec (10)

### 1.14.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>all</b>	Enables the generation of debug messages for all NTP messages.
<b>general</b>	Enables the generation of debug messages for NTP general messages.
<b>packet</b>	Enables the generation of debug messages for NTP packet messages.
<b>update</b>	Enables the generation of debug messages for NTP time update messages.

### 1.14.4 Default

The generation of debug messages for NTP is disabled.



### 1.14.5 Usage Guidelines

Use the `debug ntp` command to enable the various types of debug messages for NTP.

**Note:** The SmartEdge 100 router does not support the `standby` and `switchover` keywords.

To store messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command in exec mode to display these stored messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---

---

Use the `no` form of this command to disable the generation of debug messages for NTP.

### 1.14.6 Examples

The following example show how to enable the generation of debug messages for NTP packet and update messages. The commands are followed by sample debug message output:

```
[local]Redback#debug ntp packet
```

```
[local]Redback#debug ntp update
```

```
Feb 23 23:36:37: %NTP-7-PKT: Send packet to 155.53.12.12
```

```
Feb 23 23:36:37: %NTP-7-PKT: Receive packet from 155.53.12.12
```

```
Feb 23 23:36:39: %NTP-7-PKT: Send packet to 155.53.32.75
```



```
Feb 23 23:36:39: %NTP-7-PKT: Receive packet from 155.53.12.12
Feb 23 23:36:39: %NTP-7-PKT: Receive packet from 155.53.32.75
Feb 23 23:36:43: %NTP-7-PKT: Send packet to 155.53.32.75
Feb 23 23:36:43: %NTP-7-PKT: Receive packet from 155.53.32.75
Feb 23 23:36:44: %NTP-7-PKT: Send packet to 155.53.12.12
Feb 23 23:36:44: %NTP-7-PKT: Receive packet from 155.53.12.12
Feb 23 23:36:47: %NTP-7-PKT: Send packet to 155.53.32.75
Feb 23 23:36:47: %NTP-7-PKT: Receive packet from 155.53.32.75
Feb 23 23:36:48: %NTP-7-PKT: Send packet to 155.53.12.12
Feb 23 23:36:48: %NTP-7-PKT: Receive packet from 155.53.12.12
Feb 23 23:36:49: %NTP-7-PKT: Send packet to 155.53.32.75
Feb 23 23:36:49: %NTP-7-PKT: Receive packet from 155.53.32.75
Feb 23 23:36:50: %NTP-7-PKT: Send packet to 155.53.12.12
Feb 23 23:36:50: %NTP-7-PKT: Receive packet from 155.53.12.12
Feb 23 23:36:50: %NTP-7-UPD: adjust clock: current offset at 4287 usecs
Feb 23 23:36:51: %NTP-7-PKT: Send packet to 155.53.32.75
Feb 23 23:36:51: %NTP-7-PKT: Receive packet from 155.53.32.75
Feb 23 23:37:06: %NTP-7-PKT: Send packet to 155.53.32.75
Feb 23 23:37:06: %NTP-7-PKT: Receive packet from 155.53.32.75
Feb 23 23:37:07: %NTP-7-PKT: Send packet to 155.53.12.12
Feb 23 23:37:07: %NTP-7-PKT: Receive packet from 155.53.12.12
Feb 23 23:37:07: %NTP-7-UPD: adjust clock: current offset at 4200 usecs
```



## 1.15 debug ospf

```
debug [boot {active | standby} | switchover] ospf
```

```
no debug [boot {active | standby} | switchover] ospf
```

### 1.15.1 Purpose

Enables the generation of debug messages for all Open Shortest Path First (OSPF) events.

### 1.15.2 Command Mode

exec (10)

### 1.15.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

### 1.15.4 Default

The generation of debug messages is disabled.

### 1.15.5 Usage Guidelines

Use the `debug ospf` command to enable the generation of debug messages for all OSPF events.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.



---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

### 1.15.6 Examples

The following example shows how to enable the generation of debug messages for OSPF events:

```
[local]Redback#debug ospf
```



## 1.16 debug ospf bfd

```
debug ospf bfd
```

```
no debug ospf bfd
```

### 1.16.1 Purpose

Enables the generation of Bidirectional Forwarding Detection (BFD) debug messages for all Open Shortest Path First (OSPF) instances.

### 1.16.2 Command Mode

exec (10)

### 1.16.3 Syntax Description

This command has no keywords or arguments.

### 1.16.4 Default

The generation of debug messages is disabled.

### 1.16.5 Usage Guidelines

Use the `debug ospf bfd` command to enable the generation of BFD debug messages for all OSPF instances.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.



**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

### 1.16.6 Examples

The following example shows how to enable the generation of BFD debug messages for all OSPF instances:

```
[local]Redback#debug opsf bfd
```



## 1.17 debug ospf flooding

```
debug [boot {active | standby} | switchover] ospf flooding
```

```
no debug [boot {active | standby} | switchover] ospf flooding
```

### 1.17.1 Purpose

Enables the generation of debug messages for Open Shortest Path First (OSPF) flooding events.

### 1.17.2 Command Mode

exec (10)

### 1.17.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

### 1.17.4 Default

The generation of debug messages is disabled.

### 1.17.5 Usage Guidelines

Use the `debug ospf flooding` command to enable the generation of debug messages for OSPF flooding events.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.



---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

### 1.17.6 Examples

The following example shows how to enable the generation of debug messages for OSPF flooding events:

```
[local]Redback#debug ospf flooding
```



## 1.18 debug ospf interface

```
debug [boot {active | standby} | switchover] ospf interface
address ip-addr
```

```
no debug [boot {active | standby} | switchover] ospf interface
address ip-addr
```

### 1.18.1 Purpose

Enables the generation of debug messages for a specific Open Shortest Path First (OSPF) interface.

### 1.18.2 Command Mode

exec (10)

### 1.18.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<code>address ip-addr</code>	Interface IP address. Enables the generation of debug messages for the interface using the specified IP address.

### 1.18.4 Default

The generation of debug messages is disabled.

### 1.18.5 Usage Guidelines

Use the `debug ospf interface` command to enable the generation of debug messages for a specific interface.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.



Use the **switchover** keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the **logging debug** command (in global configuration mode). Use the **show log** command (in exec mode) to display these stored debug messages.

To display messages in real time, use the **logging console** command (in context configuration mode) if you are connected to the system through the console port. Or, use the **terminal monitor** command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about **logging** commands and the **terminal monitor** command, see the *Command List*.

Use the **no** form of this command to disable the generation of debug messages.

#### 1.18.6 Examples

The following example shows how to enable the generation of debug messages for events for the OSPF interface at IP address 10.10.1.1:

```
[local]Redback#debug ospf interface 10.10.1.1
```



## 1.19 debug ospf ldp-synchronization

```
debug ospf ldp-synchronization
```

```
no debug ospf ldp-synchronization
```

### 1.19.1 Purpose

Enable the generation of Open Shortest Path First (OSPF) Label Distribution Protocol-Interior Gateway Protocol (LDP-IGP) debug messages.

### 1.19.2 Command Mode

exec (10)

### 1.19.3 Syntax Description

This command has no keywords or arguments.

### 1.19.4 Default

The generation of debug messages is disabled.

### 1.19.5 Usage Guidelines

Use the `debug ospf ldp-synchronization` command to enable the generation of OSPF LDP-IGP debug messages.

Use the `no` form of this command to disable the generation of OSPF LDP-IGP debug messages.

### 1.19.6 Examples

The following example shows how to enable the generation of OSPF LDP-IGP debug message on your system:

```
[local]Redback#debug ospf ldp-synchronization
```



## 1.20 debug ospf lsdb

```
debug [boot {active | standby} | switchover] ospf lsdb
[area-scope-opaque | as-scope-opaque | external |
link-scope-opaque | network | nssa | router | summary-asbr |
summary-network] [advertising-router ip-addr] [link-state-id
ip-addr]
```

```
no debug [boot {active | standby} | switchover] ospf
lsdb [area-scope-opaque | as-scope-opaque | external |
link-scope-opaque | network | nssa | router | summary-asbr |
summary-network] [advertising router ip-addr] [link-state-id
ip-addr]
```

### 1.20.1 Purpose

Enables the generation of debug messages for Open Shortest Path First (OSPF) link-state database (LSDB) events.

### 1.20.2 Command Mode

exec (10)

### 1.20.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>area-scope-opaque</b>	Optional. Enables the generation of debug messages for area scoped opaque LSDB events only.
<b>as-scope-opaque</b>	Optional. Enables the generation of debug messages for AS scoped opaque LSDB events only.
<b>external</b>	Optional. Enables the generation of debug messages for external LSDB events only.
<b>link-scope-opaque</b>	Optional. Enables the generation of debug messages for link scoped opaque LSDB events only.
<b>network</b>	Optional. Enables the generation of debug messages for network LSDB event only.



<code>nssa</code>	Optional. Enables the generation of debug messages for not-so-stubby-area (NSSA) events only.
<code>router</code>	Optional. Enables the generation of debug messages for router events only.
<code>summary-asbr</code>	Optional. Enables the generation of debug messages for summaries of autonomous system boundary router (ASBR) events only.
<code>summary-network</code>	Optional. Enables the generation of debug messages for summaries of area border router (ABR) events only.
<code>advertising-router ip-addr</code>	Optional. Advertising router IP address. Enables the generation of debug messages for the specified IP address only.
<code>link-state-id ip-addr</code>	Optional. Link-state ID address. Enables the generation of debug messages for the specified IP address only.

#### 1.20.4 Default

The generation of debug messages is disabled.

#### 1.20.5 Usage Guidelines

Use the `debug ospf lsdb` command to enable the generation of debug messages for OSPF LSDB events.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---



---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---



---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the



console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

## 1.20.6 Examples

The following example shows how to enable the generation of debug messages for all OSPF LSDB events:

```
[local]Redback#debug ospf lsdb
```



## 1.21 debug ospf neighbor

```
debug [boot {active | standby} | switchover] ospf neighbor
[neighbor-id ip-addr]
```

```
no debug [boot {active | standby} | switchover] ospf neighbor
[neighbor-id ip-addr]
```

### 1.21.1 Purpose

Enables the generation of debug messages for Open Shortest Path First (OSPF) neighbor events.

### 1.21.2 Command Mode

exec (10)

### 1.21.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<code>neighbor-id ip-addr</code>	Optional. Neighbor ID. Enables the generation of debug messages for the specified neighbor only.

### 1.21.4 Default

The generation of debug messages is disabled.

### 1.21.5 Usage Guidelines

Use the `debug ospf neighbor` command to enable the generation of debug messages for OSPF neighbor events.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.



Use the **switchover** keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---

---

### Caution!

Risk of performance loss. Enabling debug messages can severely affect system performance. Use caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the **logging debug** command in global configuration mode. Use the **show log** command(in exec mode to display these stored debug messages.

To display messages in real time if you are connected to the system through the console port, use the **logging console** command in context configuration mode. If you are connected to the system through a Telnet or Secure Shell (SSH) session, use the **terminal monitor** command in exec mode. For more information about these commands, see the *Command List*.

Use the **no** form of this command to disable debug messages.

#### 1.21.6 Examples

The following example enables the generation of debug messages for only the OSPF neighbor at IP address 43.23.45.8:

```
[local]Redback#debug ospf neighbor neighbor-id 43.23.45.8
```



## 1.22 debug ospf nsr

### 1.22.1 Purpose

Enables the generation of debug messages for (OSPF non-stop routing (NSR) events.

### 1.22.2 Command Mode

exec (10)

### 1.22.3 Syntax Description

This command has no keywords or arguments.

### 1.22.4 Default

The generation of debug messages is disabled.

### 1.22.5 Usage Guidelines

Use the `debug ospf nsr` command to enable the generation of debug messages for OSPF NSR events.

To enable debug messages on the standby XCRP, use the `debug standby ospf nsr` command on the active XCRP Controller card.

---

---

### Caution!

Risk of performance loss. Enabling debug messages can severely affect system performance. Use caution when enabling debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command in global configuration mode. To display these stored messages, use the `show log` command in exec mode.

To display messages in real time if you are connected to the system through the console port, use the `logging console` command in context configuration mode. If you are connected to the system through a Telnet or Secure Shell (SSH) session, use the `terminal monitor` command in exec mode. For more information about these commands, see the *Command List*.



Use the `no` form of this command to disable debug messages.

## 1.22.6 Examples

The following example enables the generation of debug messages for OSPF NSR:

```
[local]Redback#debug ospf nsr
```

Debug output is displayed on the active console or management port, tagged with [ACTIVE].

```
Jun 30 17:11:56.764: %OSPF-7-NSR: [ACTIVE] NSR Peer  
Standby EP is available
```

```
Jun 30 17:11:56.790: %OSPF-7-NSR: [ACTIVE] Standby OSPF  
is born
```



## 1.23 debug ospf packet

```
debug [boot {active | standby} | switchover] ospf packet [ack
| database | errors | hello | request | update] [send | rcv]
[destination ip-addr | source ip-addr]
```

```
no debug [boot {active | standby} | switchover] ospf packet
[ack | database | errors | hello | request | update] [send | rcv]
[destination ip-addr | source ip-addr]
```

### 1.23.1 Purpose

Enables the generation of debug messages for Open Shortest Path First (OSPF) packet events.

### 1.23.2 Command Mode

exec (10)

### 1.23.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>ack</b>	Optional. Enables the generation of debug messages for acknowledgement (ack) packets only.
<b>database</b>	Optional. Enables the generation of debug messages for database description packets only.
<b>errors</b>	Optional. Enables the generation of debug messages for packet errors only.
<b>hello</b>	Optional. Enables the generation of debug messages for Hello packets only.
<b>request</b>	Optional. Enables the generation of debug messages for link-state request packets only.
<b>update</b>	Optional. Enables the generation of debug messages for link-state update packets only.
<b>send</b>	Optional. Enables the generation of debug messages for sent packets only.



<code>recv</code>	Optional. Enables the generation of debug messages for received packets only.
<code>destination ip-addr</code>	Optional. Destination IP address. Enables the generation of debug messages for packets sent to the specified IP address only.
<code>source ip-addr</code>	Optional. Source IP address. Enables the generation of debug messages for packets coming from the specified IP address only.

### 1.23.4 Default

The generation of debug messages is disabled.

### 1.23.5 Usage Guidelines

Use the `debug ospf packet` command to enable the generation of debug messages for OSPF packet events.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.



### 1.23.6 Examples

The following example shows how to enable the generation of debug messages for OSPF Hello packet events:

```
[local]Redback#debug ospf packet hello
```



## 1.24 debug ospf rcm

```
debug [boot {active | standby} | switchover] ospf rcm
```

```
no debug [boot {active | standby} | switchover] ospf rcm
```

### 1.24.1 Purpose

Enables the generation of debug messages for interactions between Open Shortest Path First (OSPF) and the Router Configuration Manager (RCM).

### 1.24.2 Command Mode

exec (10)

### 1.24.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

### 1.24.4 Default

The generation of debug messages is disabled.

### 1.24.5 Usage Guidelines

Use the `debug ospf rcm` command to enable the generation of debug messages for interactions between OSPF and the RCM. The RCM manages the configuration database.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.



---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

### 1.24.6 Examples

The following example shows how to enable the generation of debug messages for OSPF RCM events:

```
[local]Redback#debug ospf rcm
```



## 1.25 debug ospf redistribution

```
debug [boot {active | standby} | switchover] ospf redistribution  
[instance instance-id]
```

```
no debug [boot {active | standby} | switchover] ospf  
redistribution [instance instance-id]
```

### 1.25.1 Purpose

Enables the generation of debug messages for Open Shortest Path First (OSPF) redistribution events.

### 1.25.2 Command Mode

exec (10)

### 1.25.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<code>instance <i>instance-id</i></code>	Optional. OSPF instance ID. The range of values is 1 to 65,535. Enables the generation of debug messages for the specified OSPF instance only.

### 1.25.4 Default

The generation of debug messages is disabled.

### 1.25.5 Usage Guidelines

Use the `debug ospf redistribution` command to enable the generation of debug messages for OSPF redistribution events.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.



Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

#### 1.25.6 Examples

The following example shows how to enable the generation of debug messages for redistribution events pertaining to OSPF instance 64001:

```
[local]Redback#debug ospf redistribution instance 64001
```



## 1.26 debug ospf rib

```
debug [boot {active | standby} | switchover] ospf rib [area
area-id] [instance instance-id] [route-type {external |
inter-area | intra-area}]
```

```
no debug [boot {active | standby} | switchover] ospf rib
[area area-id] [instance instance-id] [route-type {external |
inter-area | intra-area}]
```

### 1.26.1 Purpose

Enables the generation of debug messages for interactions between Open Shortest Path First (OSPF) and the Routing Information Base (RIB).

### 1.26.2 Command Mode

exec (10)

### 1.26.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<code>area area-id</code>	Optional. Area ID. Enables the generation of debug messages for only the specified area.
<code>instance instance-id</code>	Optional. OSPF instance ID. The range of values is 1 to 65,535. Enables the generation of debug messages for only the specified instance.
<code>route-type</code>	Optional. Enables the generation of debug messages for the specified route type.
<code>external</code>	Enables the generation of debug messages for external OSPF routes.
<code>inter-area</code>	Enables the generation of debug messages for interarea OSPF routes.
<code>intra-area</code>	Enables the generation of debug messages for intra-area OSPF routes.



#### 1.26.4 Default

The generation of debug messages is disabled.

#### 1.26.5 Usage Guidelines

Use the `debug ospf rib` command to enable the generation of debug messages for interactions between OSPF and the RIB.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

#### 1.26.6 Examples

The following example shows how to enable debug messages for OSPF RIB events for external routes:

```
[local]Redback#debug ospf rib route-type external
```



## 1.27 debug ospf spf

```
debug [boot {active | standby} | switchover] ospf spf [externals  
| incremental | intra-area | nssa | scheduling | summaries |  
virtual-links] [area area-id] [instance instance-id]
```

```
no debug [boot {active | standby} | switchover] ospf spf [externals  
| incremental | intra-area | nssa | scheduling | summaries |  
virtual-links] [area area-id] [instance instance-id]
```

### 1.27.1 Purpose

Enables the generation of debug messages for Open Shortest Path First (OSPF) Shortest Path First (SPF) calculations.

### 1.27.2 Command Mode

exec (10)

### 1.27.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>externals</b>	Optional. Enables the generation of debug messages for external computations.
<b>incremental</b>	Optional. Enables the generation of debug messages for incremental computations.
<b>intra-area</b>	Optional. Enables the generation of debug messages for intra-area path computations.
<b>nssa</b>	Optional. Enables the generation of debug messages for not-so-stubby-area (NSSA) computations.
<b>scheduling</b>	Optional. Enables the generation of debug messages for SPF scheduling and triggering.
<b>summaries</b>	Optional. Enables the generation of debug messages for summary computations.
<b>virtual-links</b>	Optional. Enables the generation of debug messages for virtual links.



<code>area area-id</code>	Optional. Enables the generation of debug messages for all areas or for the specified area.
<code>instance instance-id</code>	Optional. Enables the generation of debug messages for all instances or for the specified instance.

#### 1.27.4 Default

The generation of debug messages is disabled.

#### 1.27.5 Usage Guidelines

Use the `debug ospf spf` command to enable the generation of debug messages for OSPF SPF calculations.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---



---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---



---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.



## 1.27.6 Examples

The following example shows how to enable the generation of debug messages for OSPF SPF calculations:

```
[local]Redback#debug ospf spf
```



## 1.28 debug packet

For circuits specified by link groups, the syntax is:

```
debug packet circuit lg {lg-name | id lg-id} [circuit-id] {in |
out} [count number of packets] [sample-interval sample-interval-i
n-msec] [size packet-size-in-bytes | header-only] [ip acl acl-name
[context acl-context]]
```

```
no debug packet circuit lg {lg-name | id lg-id} [circuit-id]
{in | out} [count number of packets] [sample-interval
sample-interval-in-msec] [size packet-size-in-bytes | header-only]
[ip acl acl-name [context acl-context]]
```

```
no debug packet all
```

For circuits specified by slot and port, the syntax is:

```
debug packet circuit {slot/port} [circuit-id] {in | out} [count
number of packets] [sample-interval sample-interval-in-msec]
[size packet-size-in-bytes | header-only] [ip acl acl-name [context
acl-context]]
```

```
no debug packet circuit {slot/port} [circuit-id] {in | out} [count
number of packets] [sample-interval sample-interval-in-msec]
[size packet-size-in-bytes | header-only] [ip acl acl-name [context
acl-context]]
```

```
no debug packet all
```

### 1.28.1 Purpose

Enables the generation of debug messages that include the header and data of the packet on a specified circuit for a specified traffic direction.

### 1.28.2 Command Mode

exec (10)

### 1.28.3 Syntax Description

<code>circuit</code>	Specifies the circuit to monitor.
<code>lg lg-name</code>	Specifies the name of the link group for which the generation of debug messages is enabled.
<code>id lg-id</code>	Specifies the ID of the link group for which the generation of debug messages is enabled.
<code>slot</code>	Chassis slot number of the traffic card for which the generation of debug messages is enabled.



<code>port</code>	Required if you enter the <code>slot</code> argument. Port number for which the generation of debug messages is enabled.
<code>circuit-id</code>	Circuit identifier according to one of the constructs listed in Table 2 for which the generation of debug messages is enabled.
<code>in</code>	Specifies that the debugging of packets is to apply only to incoming traffic on the circuit.
<code>out</code>	Specifies that the debugging of packets is to apply only to outgoing traffic on the circuit.
<code>count number of packets</code>	Specifies the number of consecutive packets to display. The range of values is from 1 to 1000.
<code>sample-interval sample-interval-in-msec</code>	Specifies a minimum interval between packet samples in milliseconds. The range of values is from 1 to 20000.
<code>size packet-size-in-bytes</code>	Specifies the number of bytes from the beginning of a packet to display. The range of values is from 1 to 160.
<code>header-only</code>	Specifies that only packet headers are to be displayed.
<code>ip acl acl-name</code>	Specifies the name of the IP ACL to be used to filter packets for debugging.
<code>context acl-context</code>	Specifies the name of the context in which the specified IP ACL resides.
<code>all</code>	Applies only to the <code>no</code> form of the <code>debug packet</code> command. Cancels the generation of debug messages for all circuits in both directions of the current CLI session.

## 1.28.4 Default

The generation of debug messages for packets on a circuit is disabled.

## 1.28.5 Usage Guidelines

Use the `debug packet circuit` command to enable the generation of debug messages that include the header and data of the packet on a specified circuit for a specified traffic direction. This command supports the following circuit types:

- Cross-connect circuit
- 802.1Q PVC
- 802.1Q tunnel
- PPPoE circuit
- Cross-connect access circuits

The `debug packet circuit` command also supports these listed circuit types over the following link groups:

- Access link group (economical and noneconomical)
- 802.1Q link group (applies to constituent circuits only)
- Ethernet link group (applies to constituent circuits only)



Enabling packet sampling on more than one circuit is permitted. Specify whether the debugging of packets is to apply to incoming or outgoing traffic on the circuit by using either the `in` or `out` keyword.

To enable the generation of debug messages for packets on a circuit belonging to a specified link group, use the `lg lg-name` or `lg lg-id` construct with the `debug packet circuit` command.

For 802.1Q and Ethernet link groups, you can apply the `debug packet circuit` command to only each constituent circuit within the link group instead of the entire link group. The constituent circuits are the physical ports or channels in the link group bundle. Before applying this command to a constituent circuit, obtain the slot and port, or slot, port, and circuit ID of the constituent circuit by using the `show link-group lg-name detail` command. This information is displayed in the Constituent Circuits field within the command output. Use the `debug packet circuit slot/port [circuit-id] {in | out}` command to sample traffic on the given constituent circuit within the link group.

**Note:** Applying the `debug packet circuit` command to a link group is functionally equivalent to applying this command to each constituent circuit of the link group. The application implies that the debug request remains active on the constituent circuits that have not received or transmitted the requested number of packets defined by the `count` parameter until the request is canceled by the `no debug packet circuit` command. We recommend that after you issue the `debug packet circuit` command for a link group and the requested number of packets is displayed, issue the `no debug packet circuit` command for the link group.

To enable the generation of debug messages for packets on a circuit within a specified slot and port number of a traffic card, use the `slot` and `port` keywords with the `debug packet circuit` command. After you have selected either the link group or slot and port in which the circuit resides, specify the `circuit-id` argument.

The `debug packet circuit` command is not persistent. After closing the vty session, this command is disabled.

Table 2 lists the values for the `circuit-id` argument.

Table 2 Values for the *circuit-id* Argument

Construct	Description
<code>pppoe</code> <code>sess-id</code>	Specifies a PPPoE circuit. The range of values for the <i>sess-id</i> argument is from 1 to 65534.
<code>vlan-id</code> <code>vlan-id</code>	Specifies the VLAN tag value for an 802.1Q tunnel or PVC. The <i>vlan-id</i> argument is one of the following arguments: <ul style="list-style-type: none"><li>• <i>pvc-vlan-id</i>—VLAN tag value of a PVC that is not within an 802.1Q tunnel.</li><li>• <i>tunl-vlan-id</i>—VLAN tag value of a tunnel.</li><li>• <i>tunl-vlan-id;pvc-vlan-id</i>—VLAN tag value for the tunnel followed by the VLAN tag value for the PVC within the tunnel.</li></ul> The range of values for either VLAN tag value is 1 to 4095.

The syntax for the *circuit-id* argument is [`pppoe sess-id`] [`vlan-id vlan-id`].

Use the `no debug packet all` command to cancel the generation of debug messages for all circuits in both directions of the current CLI session. Use the `no debug packet circuit {lg {lg-name | id lg-id} | slot/port [circuit-id]}` command with either the `in` or `out` keyword to cancel the generation of debug messages for incoming or outgoing traffic on the specified circuit.

## 1.28.6 Examples

The following example shows how to enable the generation of debug messages for incoming packets on a circuit in slot 9 and port 1. The debug messages are to display 1000 consecutive packets with each packet displaying the first 128 bytes:



```
[local]Redback#debug packet circuit 9/1 in count 1000 size 128
```

```
741474a9/0000000144/559600000:09/IPPA/EU00: 9/1:1023:63/1/1/5
```

```
debug packet 1/1 timestamp 144.847569955 length 128/252 eu 5
```

```
ETH  DST_MAC 00:30:88:02:7D:74 SRC_MAC 00:10:94:00:00:01 ETHER_TYPE ipv4(0800)
```

```
IP  VER 4 HDR_LEN 20 LEN 238 ID 0 FRAG_OFFS 0 TTL 10 PROTO udp
```

```
  CHKSUM 9DF1 SRC 9.1.0.2 DST 9.10.0.2
```

```
UDP  SRC_PORT 1024 DST_PORT 0 LEN 218 CHK_SUM FFFF
```

```
DATA 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
```

```
  10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F
```

```
  20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F
```

```
  30 31 32 33 34 35 36 37 38 39 3A 3B 3C 3D 3E 3F
```

```
  40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F
```

```
  50 51 52 53 54 55
```



## 1.29 debug pedgr

```
debug [{boot {active | standby} | switchover}] pedgr
```

```
no debug [{boot {active | standby} | switchover}] pedgr
```

### 1.29.1 Purpose

Enables the generation of debug messages for process execution descriptor graph (PEDGR) events.

### 1.29.2 Command Mode

exec (10)

### 1.29.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

### 1.29.4 Default

The generation of debug messages for PEDGR events is disabled.

### 1.29.5 Usage Guidelines

Use the `debug pedgr` command to enable the generation of debug messages for PEDGR events.

---



---

## Caution!

Risk of performance loss. Enabling debug messages can severely affect system performance. To reduce the risk, use `debug` commands only when necessary and for as short a time as possible.

---



---



Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

**Note:** The SmartEdge router does not support standby controller cards; therefore, the `standby` and `switchover` keywords are not applicable.

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

## 1.29.6 Examples

The following example shows how to enable the generation of debug messages for PEDGR events:

```
[local]Redback#debug pedgr
```



## 1.30 debug pim

```
debug [boot {active | standby} | switchover] pim [group-addr]
```

```
no debug [boot {active | standby} | switchover] pim [group-addr]
```

### 1.30.1 Purpose

Enables the generation of Protocol Independent Multicast (PIM) debug messages.

### 1.30.2 Command Mode

exec (10)

### 1.30.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<code>group-addr</code>	Optional. IP address of the Internet Group Management Protocol (IGMP) group.

### 1.30.4 Default

The generation of debug messages is disabled.

### 1.30.5 Usage Guidelines

Use the `debug pim` command to enable the generation of PIM debug messages.

Use the `group-addr` argument to enable the generation of PIM debug messages for a specific IGMP group.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.



Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secured Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of PIM debug messages.

#### 1.30.6 Examples

The following example shows how to enable the generation of PIM debug messages for the group, 224.1.1.1:

```
[local]Redback#debug pim 224.1.1.1
```



## 1.31 debug pim bfd

`debug pim bfd`

`no debug pim bfd`

### 1.31.1 Purpose

Enables generation of Protocol Independent Multicast (PIM) Bidirectional Forwarding Detection (BFD) debug messages.

### 1.31.2 Command Mode

exec (10)

### 1.31.3 Syntax Description

This command has no keywords or arguments.

### 1.31.4 Default

Generation of debug messages is disabled.

### 1.31.5 Usage Guidelines

Use the `debug pim bfd` command to enable generation of PIM BFD debug messages.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the



console port. Use the `terminal monitor` command (in exec mode) if you are connected through a Telnet or Secured Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable generation of PIM BFD debug messages.

### 1.31.6 Examples

The following example shows how to enable generation of PIM BFD debug messages.

```
[local]router1#debug pim bfd
```



## 1.32 debug pim mcastmgr

```
debug pim mcastmgr
```

```
no debug pim mcastmgr
```

### 1.32.1 Purpose

Enables generation of Protocol Independent Multicast (PIM) multicast manager debug messages.

### 1.32.2 Command Mode

exec (10)

### 1.32.3 Syntax Description

This command has no keywords or arguments.

### 1.32.4 Default

Generation of debug messages is disabled.

### 1.32.5 Usage Guidelines

Use the `debug pim mcastmgr` command to enable generation of PIM multicast manager debug messages.

---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Use the `terminal monitor` command (in exec mode) if you are connected through a Telnet or Secured Shell (SSH) session.



**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable generation of PIM multicast manager debug messages.

### 1.32.6 Examples

The following example shows how to enable generation of PIM multicast manager debug messages.

```
[local] [local] Redback#debug pim mcastmgr
```



## 1.33 debug pim packet

```
debug [boot {active | standby} | switchover] pim packet  
[group-addr]
```

```
no debug [boot {active | standby} | switchover] pim packet  
[group-addr]
```

### 1.33.1 Purpose

Enables the generation of Protocol Independent Multicast (PIM) multicast data packet debug messages.

### 1.33.2 Command Mode

exec (10)

### 1.33.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<code>group-addr</code>	Optional. IP address of the Internet Group Management Protocol (IGMP) group.

### 1.33.4 Default

The generation of debug messages is disabled.

### 1.33.5 Usage Guidelines

Use the `debug pim packet` command to enable the generation of PIM multicast data packet debug messages.

Use the `group-addr` argument to enable the generation of PIM multicast data packet debug messages for a specific IGMP group.



Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secured Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of PIM debug multicast data packet messages.

#### 1.33.6 Examples

The following example shows how to enable the generation of all PIM multicast data packet debug messages:

```
[local]Redback#debug pim packet
```

The following example shows how to enable the generation of PIM multicast data packet debug messages for the IGMP group, 224.1.1.1:

```
[local]Redback#debug packet 224.1.1.1
```



## 1.34 debug pim ppa

```
debug [boot {active | standby} | switchover] pim ppa [slot-num  
[group-addr] | group-addr]
```

```
no debug [boot {active | standby} | switchover] pim ppa
```

### 1.34.1 Purpose

Enables the generation of debug messages for Protocol Independent Multicast (PIM) routes downloaded to Packet Processing ASICs (PPAs).

### 1.34.2 Command Mode

exec (10)

### 1.34.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<code>slot-num</code>	Optional. Slot number.
<code>group-addr</code>	Optional. IP address of the Internet Group Management Protocol (IGMP) group.

### 1.34.4 Default

The generation of debug messages is disabled.

### 1.34.5 Usage Guidelines

Use the `debug pim ppa` command to enable the generation of debug messages for PIM routes downloaded to PPAs.

Use the `slot-num` argument or `group-addr` argument to enable the generation of PIM multicast data packet debug messages only for a specific slot or IGMP group, respectively. Use the `slot-num group-addr` construct to enable the generation of PIM multicast data packet debug messages only for a specific IGMP group on a specific slot.



Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secured Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages for PIM routes downloaded to PPAs.

#### 1.34.6 Examples

The following example shows how to enable the generation of debug messages for PIM routes downloaded to PPAs:

```
[local]Redback#debug pim ppa
```



## 1.35 debug pim rp-mapping

```
debug [boot {active | standby} | switchover] pim rp-mapping
```

```
no debug [boot {active | standby} | switchover] pim rp-mapping
```

### 1.35.1 Purpose

Enables the generation of bootstrap router (BSR) and rendezvous point (RP) debug messages.

### 1.35.2 Command Mode

exec (10)

### 1.35.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

### 1.35.4 Default

The generation of debug messages is disabled.

### 1.35.5 Usage Guidelines

Use the `debug pim rp-mapping` command to enable the generation of BSR and RP debug messages.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.



---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secured Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of BSR and RP debug messages.

### 1.35.6 Examples

The following example shows how to enable the generation of BSR and RP debug messages:

```
[local]Redback#debug pim rp-mapping
```



## 1.36 debug pm

```
debug [{boot {active | standby} | switchover}] pm [{all | config |  
general | ipc | memory-error | process}]
```

```
no debug [{boot {active | standby} | switchover}] pm [{all | config |  
general | ipc | memory-error | process}]
```

### 1.36.1 Purpose

Enables the generation of debug messages for the Process Manager (PM).

### 1.36.2 Command Mode

exec (10)

### 1.36.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<code>all</code>	Optional. Enables debug messages for all PM events.
<code>config</code>	Optional. Enables debug messages for PM configuration events.
<code>general</code>	Optional. Enables debug messages for general PM events.
<code>ipc</code>	Optional. Enables debug messages for interprocess communication (IPC) PM events.
<code>memory-error</code>	Optional. Enables debug messages for PM memory-error events.
<code>process</code>	Optional. Enables debug messages for PM process events.

### 1.36.4 Default

The generation of debug messages for the PM is disabled.



### 1.36.5 Usage Guidelines

Use the `debug pm` command to enable the generation of debug messages for the PM.

---

---

#### Caution!

Risk of performance loss. Enabling debug messages can severely affect system performance. To reduce the risk, use debug commands only when necessary and for as short a time as possible.

---

---

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

**Note:** The SmartEdge router does not support standby controller cards; therefore, the `standby` and `switchover` keywords are not applicable.

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

### 1.36.6 Examples

The following example shows how to enable PM memory-error events:

```
[local]Redback#debug pm memory-error
```



## 1.37 debug policy access-list

```
debug [{boot {active | standby} | switchover}] policy access-list  
no debug [{boot {active | standby} | switchover}] policy  
access-list
```

### 1.37.1 Purpose

Enables the generation of debug messages for policy access control lists (ACLs).

### 1.37.2 Command Mode

exec (10)

### 1.37.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

### 1.37.4 Default

The generation of debug messages is disabled.

### 1.37.5 Usage Guidelines

Use the `debug policy access-list` command to enable the generation of debug messages for policy ACLs.

**Note:** The SmartEdge router does not support the `standby` and `switchover` keywords.

To store messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command in exec mode to display these stored messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the



console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about logging commands and the `terminal monitor` command, see the *Command List*.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of any debug messages on a production system.

---

---

Use the `no` form of this command to disable the generation of debug messages for policy ACLs.

#### 1.37.6 Examples

The following example shows how to enable the generation of debug messages for policy ACLs:

```
[local]Redback#debug policy access-list
```



## 1.38 debug policy general

```
debug [boot {active | standby} | switchover] policy general
```

```
no debug [boot {active | standby} | switchover] policy general
```

### 1.38.1 Purpose

Enables the generation of debug messages for all configured routing policies, including autonomous system (AS) path lists, community lists, IP access control lists (ACLs), IP prefix lists, key chains, and route maps.

### 1.38.2 Command Mode

exec (10)

### 1.38.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

### 1.38.4 Default

None

### 1.38.5 Usage Guidelines

Use the `debug policy general` command to enable the generation of debug messages for all configured routing policies, including AS path lists, community lists, IP ACLs, IP prefix lists, key chains, and route maps.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.



---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

### 1.38.6 Examples

The following example shows how to enable the generation of debug messages for all configured routing policies:

```
[local]Redback#debug policy general
```



## 1.39 debug ppp

`debug [boot {active | standby} | switchover] ppp {all | event-type}`

`no debug [boot {active | standby} | switchover] ppp {all | event-type}`

### 1.39.1 Purpose

Enables the generation of debug messages for various types of Point-to-Point Protocol (PPP) events on the system.

### 1.39.2 Command Mode

exec (10)

### 1.39.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.  Use the <code>boot active</code> or <code>boot standby</code> construct to enable debug messages during a system reload for the active or standby controller card, respectively.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card and enable debug messages while the system is switching from the active to the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<code>all</code>	Enables the generation of debug messages for all PPP event types.
<code>event-type</code>	Type of event, according to one of the keywords listed in Table 3.

### 1.39.4 Default

The generation of debug messages for all PPP events on the system is disabled.



### 1.39.5 Usage Guidelines

Use the `debug ppp` command to enable the generation of debug messages for various types of PPP events on the system.

Table 3 lists the types of PPP events for which you can enable debug messages.

*Table 3 Event Types*

Keyword	Description
<code>authentication</code>	Password Authentication Protocol (PAP)/Challenge Handshake Authentication Protocol (CHAP) authentication events
<code>circuit</code>	Circuit-related events
<code>config</code>	Configuration-related events
<code>down</code>	PPP session down-related events
<code>exception</code>	Exception events, such as when a timer expires
<code>fsm</code>	State-change events for the Finite State Machine (FSM)
<code>ipc</code>	Interprocess communication (IPC)-related events
<code>ipcp</code>	Internet Protocol Control Protocol (IPCP) events
<code>ism</code>	Interface and Circuit State Manager (ISM)-related events
<code>lcp</code>	Link Control Protocol (LCP) events
<code>multilink</code>	PPP multilink events
<code>negotiation</code>	Negotiation events
<code>nlcp</code>	Network Link Control Protocol (NLCP) events
<code>packet</code>	PPP packet events
<code>phase</code>	PPP phase events
<code>ppa</code>	Packet Processing ASIC (PPA)-related events
<code>rcm</code>	Router Configuration Manager (RCM)-related events
<code>session</code>	PPP session-related events
<code>timer</code>	PPP timer-related events

---



---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---



---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.



To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information on logging, see *Logging*.

Use the `no` form of this command to disable the generation of debug messages for PPP events.

### 1.39.6 Examples

The following example shows how to enable the generation of debug messages for PPP packet events:

```
[local]Redback#debug ppp packet
```



## 1.40 debug ppp multilink

```
debug [boot {active | standby} | switchover] ppp multilink
```

```
no debug [boot {active | standby} | switchover] ppp multilink
```

### 1.40.1 Purpose

Enables the generation of debug messages for Multilink Point-to-Point Protocol (MLPPP) events.

### 1.40.2 Command Mode

exec (10)

### 1.40.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

### 1.40.4 Default

The generation of debug messages for MLPPP events is disabled.

### 1.40.5 Usage Guidelines

Use the `debug ppp multilink` command to enable the generation of debug messages for MLPPP events.

---



---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of any debug messages on a production system.

---



---



Use the `boot active` or `boot standby` construct to enable debug messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debug messages while the system is switching from the active to the standby controller card.

**Note:** The SmartEdge 100 router does not support the `standby` and `switchover` keywords.

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages for MLPPP events.

## 1.40.6 Examples

The following example shows how to enable the generation of debug messages for MLPPP events:

```
[local]Redback#debug ppp multilink
```



## 1.41 debug pppoe

```
debug [boot {active | standby} | switchover] pppoe {all | cct |
discovery | exception | info | packet | timer}
```

```
no debug [boot {active | standby} | switchover] pppoe {all | cct |
discovery | exception | info | packet | timer}
```

### 1.41.1 Purpose

Enables the generation of debug messages for various types of Point-to-Point Protocol over Ethernet (PPPoE) events.

### 1.41.2 Command Mode

exec (10)

### 1.41.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>all</b>	Enables the generation of PPPoE debug messages for all types of events.
<b>cct</b>	Enables the generation of PPPoE debug messages for circuit-related events.
<b>discovery</b>	Enables the generation of PPPoE debug messages for discovery protocol-related events.
<b>exception</b>	Enables the generation of PPPoE exception debug messages.
<b>info</b>	Enables the generation of PPPoE debug messages for PPPoE information.
<b>packet</b>	Enables the generation of PPPoE debug messages for packet input and output events.
<b>timer</b>	Displays timer-related debug messages.



#### 1.41.4 Default

Debugging is disabled.

#### 1.41.5 Usage Guidelines

Use the `debug pppoe` command to enable the generation of debug messages for various types of PPPoE events.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of any debug messages on a production system.

---

---

Use the `boot active` or `boot standby` construct to enable debug messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debug messages while the system is switching from the active to the standby controller card.

**Note:** The SmartEdge 100 router does not support the `standby` and `switchover` keywords.

Use the `exception` keyword to enable the display of PPPoE exception debug messages.

Use the `timer` keyword to display timer-related debug messages.

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.



### 1.41.6 Examples

The following example shows how to enable the generation of all types of PPPoE debug messages and displays sample debug messages:

```
[local]Redback#debug pppoe all
```



## 1.42 debug qos

```
debug [{boot {active | standby} | switchover}] qos {all | cli | config
| internal | ism | ppa | rcm}
```

```
no debug [{boot {active | standby} | switchover}] qos {all | cli |
config | internal | ism | ppa | rcm}
```

### 1.42.1 Purpose

Enables the generation of quality of service (QoS) debug messages.

### 1.42.2 Command Mode

exec (10)

### 1.42.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>all</b>	Enables the generation of debug messages for all QoS events.
<b>cli</b>	Enables the generation of debug messages for QoS parser events.
<b>config</b>	Enables the generation of debug messages for QoS daemon configuration events.
<b>internal</b>	Enables the generation of debug messages for QoS internal events.
<b>ism</b>	Enables the generation of debug messages for QoS Interface and Circuit State Manager (ISM) events.
<b>ppa</b>	Enables the generation of debug messages for QoS Packet Processing ASIC (PPA) events.
<b>rcm</b>	Enables the generation of debug messages for QoS Router Configuration Manager (RCM) events.

### 1.42.4 Default

The generation of debug messages is disabled.



## 1.42.5 Usage Guidelines

Use the `debug qos` command to enable the generation of QoS debug messages.

**Note:** The SmartEdge 100 router does not support the `standby` and `switchover` keywords.

To store messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command in exec mode to display these stored messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of any debug messages on a production system.

---

---

Use the `no` form of this command to disable the generation of debug messages.

## 1.42.6 Examples

The following example shows how to enable the generation of QoS PPA event debug messages:

```
[local]Redback#debug qos ppa
```



## 1.43 debug radius

```
debug [{boot {active | standby} | switchover}] radius {accounting |
attributes | authentication | authorization | packet}
```

```
no debug [{boot {active | standby} | switchover}] radius
{accounting | attributes | authentication | authorization |
packet}
```

### 1.43.1 Purpose

Enables the generation of Remote Authentication Dial-In User Service (RADIUS) debug messages.

### 1.43.2 Command Mode

exec (10)

### 1.43.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<code>accounting</code>	Enables the generation of RADIUS accounting debug messages.
<code>attributes</code>	Enables the generation of RADIUS attribute debug messages.
<code>authentication</code>	Enables the generation of RADIUS authentication debug messages.
<code>authorization</code>	Enables the generation of RADIUS authorization debug messages.
<code>packet</code>	Enables the generation of RADIUS packet-level debug messages.

### 1.43.4 Default

The generation of debug messages is disabled.

### 1.43.5 Usage Guidelines

Use the `debug radius` command to enable the generation of RADIUS debug messages.



**Note:** The SmartEdge 100 router does not support the `standby` and `switchover` keywords.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command in global configuration mode. Use the `show log` command in exec mode to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of RADIUS debug messages.

#### 1.43.6 Examples

The following example shows how to enable the generation of packet-level debug messages for RADIUS:

```
[local]Redback#debug radius packet
```

The following example shows how to enable the generation of RADIUS authentication debug messages:

```
[local]Redback#debug radius authentication
```

The following example shows how to disable the generation of RADIUS authentication debug messages:

```
[local]Redback#no debug radius authentication
```



## 1.44 debug rcm

```
debug [{boot {active | standby} | switchover}] rcm[event]
```

```
no debug [{boot {active | standby} | switchover}] rcm [event]
```

### 1.44.1 Purpose

Enables the generation of debug messages for the Router Configuration Manager (RCM).

### 1.44.2 Command Mode

exec (10)

### 1.44.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<code>event</code>	Optional. Enables debugging messages for only RCM events.

### 1.44.4 Default

The generation of debug messages for the RCM is disabled.

### 1.44.5 Usage Guidelines

Use the `debug rcm` command to enable the generation of debug messages for the RCM.

---

---

### Caution!

Risk of performance loss. Enabling debug messages can severely affect system performance. To reduce the risk, use debug commands only when necessary and for as short a time as possible.

---

---



Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

**Note:** The SmartEdge 100 router does not support standby controller cards; therefore, the `standby` and `switchover` keywords are not applicable.

Use the optional `event` keyword to enable the debug messages for only RCM events.

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

### 1.44.6 Examples

The following example shows how to enable debugging messages for only RCM events:

```
[local]Redback#debug rcm event
```



## 1.45 debug rip

```
debug [boot {active | standby} | switchover] rip {all | config |  
global-rib | interface | packet [recv | request | response | send] |  
policy | route | timer} [detail]
```

```
no debug [boot {active | standby} | switchover] rip {all | config |  
global-rib | interface | packet [recv | request | response | send] |  
policy | route | timer} [detail]
```

### 1.45.1 Purpose

Enables the generation of Routing Information Protocol (RIP) debug messages.

### 1.45.2 Command Mode

exec (10)

### 1.45.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>all</b>	Enables the generation of messages for all RIP debug options.
<b>config</b>	Enables the generation of debug messages for RIP configuration events.
<b>global-rib</b>	Enables the generation of debug messages for global Routing Information Base (RIB) events.
<b>interface</b>	Enables the generation of debug messages for RIP interface events.
<b>packet</b>	Enables the generation of debug messages for all RIP packet events.
<b>recv</b>	Optional. Enables the generation of debug messages for only received RIP packets.
<b>request</b>	Optional. Enables the generation of debug messages for only request packets.



<b>response</b>	Optional. Enables the generation of debug messages for only response packets.
<b>send</b>	Optional. Enables the generation of debug messages for only send packets.
<b>policy</b>	Enables the generation of debug messages for RIP policy events.
<b>route</b>	Enables the generation of debug messages for RIP routing table events.
<b>timer</b>	Enables the generation of debug messages for RIP timer events.
<b>detail</b>	Optional. Provides detailed information in debug messages.

#### 1.45.4 Default

The generation of debug messages is disabled.

#### 1.45.5 Usage Guidelines

Use the `debug rip` command to enable the generation of RIP debug messages.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---



---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---



---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.



**Note:** For more information about **logging** commands and the **terminal monitor** command, see the *Command List*.

Use the **no** form of this command to disable the generation of RIP debug messages.

### 1.45.6 Examples

The following example shows how to enable the generation of debug messages for all RIP packet events:

```
[local]Redback#debug rip packet all
```



## 1.46 debug route-map

```
debug [boot {active | standby} | switchover] route-map
```

```
no debug [boot {active | standby} | switchover] route-map
```

### 1.46.1 Purpose

Enables the generation of debug messages for the maintenance of route maps and for the comparison of routes to route maps.

### 1.46.2 Command Mode

exec (10)

### 1.46.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

### 1.46.4 Default

None

### 1.46.5 Usage Guidelines

Use the `debug route-map` command to enable the generation of debug messages for the maintenance of route maps and for the comparison of routes to route maps.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.



---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

### 1.46.6 Examples

The following example shows how to enable the generation of debug messages for route maps:

```
[local]Redback#debug route-map
```



## 1.47 debug rsvp

```
debug [boot {active | standby} | switchover] rsvp [auth | cfg |
errors | frr | interface [address ip-addr] | io | ism | label | lm | nbr |
nhop | packet [recv [packet-type] | send [packet-type]] | path | ping |
psb | qos | rib | rsb | session | state-changes | timers]
```

```
no debug [boot {active | standby} | switchover] rsvp [auth | cfg |
frr | errors | interface [address ip-addr] | io | ism | label | lm | nbr |
nhop | packet [recv [packet-type] | send [packet-type]] | path | ping |
psb | qos | rib | rsb | session | state-changes | timers]
```

### 1.47.1 Purpose

Enables the generation of debug messages for Resource Reservation Protocol (RSVP) activities.

### 1.47.2 Command Mode

exec (10)

### 1.47.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>auth</b>	Optional. Enables the generation of authentication debug messages.
<b>cfg</b>	Optional. Enables the generation of configuration event debug messages.
<b>errors</b>	Optional. Enables the generation of protocol error debug messages.
<b>frr</b>	Optional. Enables the generation of fast reroute event debug messages.
<b>interface</b>	Optional. Enables the generation of interface event debug messages.
<b>address ip-addr</b>	Optional. IP address for the specific interface to be debugged.



<b>io</b>	Optional. Enables the generation of packet I/O debug messages.
<b>ism</b>	Optional. Enables the generation of Interface And Circuit State Manager (ISM) event debug messages.
<b>label</b>	Optional. Enables the generation of label operation debug messages.
<b>lm</b>	Optional. Enables the generation of label manager debug messages.
<b>nbr</b>	Optional. Enables the generation of neighbor event debug messages.
<b>nhop</b>	Optional. Enables the generation of next-hop event debug messages.
<b>packet</b>	Optional. Enables the generation of packet debug messages.
<b>recv</b>	Optional. Enables the generation of packet reception debug messages.
<b>packet-type</b>	<ul style="list-style-type: none"><li>• <b>path-err</b>—Specifies path-error packets.</li><li>• <b>path-tear</b>—Specifies path-tear packets.</li><li>• <b>resv</b>—Specifies reservation packets.</li><li>• <b>resv-err</b>—Specifies reservation error packets.</li><li>• <b>resv-tear</b>—Specifies reservation tear packet.</li></ul>
<b>send</b>	Optional. Enables the generation of packet transmission debug messages.
<b>path</b>	Optional. Enables the generation of label-switched path (LSP) origination debug messages.
<b>ping</b>	Optional. Enables the generation of RSVP LSP ping event debug messages.
<b>psb</b>	Optional. Enables the generation of path state block (PSB) event debug messages.
<b>qos</b>	Optional. Enables the generation of quality of service (QoS) event debug messages.
<b>rib</b>	Optional. Enables the generation of Routing Information Base (RIB) event debug messages.
<b>rsb</b>	Optional. Enables the generation of Reservation State Block (RSB) event debug messages.
<b>session</b>	Optional. Enables the generation of session event debug messages.



<code>state-changes</code>	Optional. Enables the generation of LSP state change debug messages.
<code>timers</code>	Optional. Enables the generation of timer events debug messages.

#### 1.47.4 Default

The generation of debug messages is disabled.

#### 1.47.5 Usage Guidelines

Use the `debug rsvp` command to enable the generation of debug messages for RSVP activities.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---



---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---



---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages for RSVP activities.



## 1.47.6 Examples

The following example shows how to enable the generation of debug messages for all RSVP activities:

```
[local]Redback#debug rsvp
```



## 1.48 debug rsvp bfd

```
debug rsvp bfd
```

```
no debug rsvp bfd
```

### 1.48.1 Purpose

Enables the generation of event debug messages for Resource Reservation Protocol (RSVP) Bidirectional Forwarding Detection (BFD).

### 1.48.2 Command Mode

exec (10)

### 1.48.3 Syntax Description

This command has no keywords or arguments.

### 1.48.4 Default

The generation of debug messages is disabled.

### 1.48.5 Usage Guidelines

Use the `debug rsvp bfd` command to enable the generation of event debug messages for RSVP BFD.

Use the `no` form of this command to disable the generation of event debug messages for RSVP BFD.

### 1.48.6 Examples

The following example shows how to enable the generation of event debug messages for RSVP BFD:

```
[local]Redback#debug rsvp bfd
```



## 1.49 debug shmlib

`debug` [{boot {active | standby} | switchover}] `shmlib`

`no debug` [{boot {active | standby} | switchover}] `shmlib`

### 1.49.1 Purpose

Enables the generation of debug messages for the shared memory library.

### 1.49.2 Command Mode

exec (10)

### 1.49.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

### 1.49.4 Default

The generation of debug messages for the shared memory library is disabled.

### 1.49.5 Usage Guidelines

Use the `debug shmlib` command to enable the generation of debug messages for the shared memory library.

---

---

### Caution!

Risk of performance loss. Enabling debug messages can severely affect system performance. To reduce the risk, use debug commands only when necessary and for as short a time as possible.

---

---



Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

**Note:** The SmartEdge 100 router does not support standby controller cards; therefore, the `standby` and `switchover` keywords are not applicable.

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

## 1.49.6 Examples

The following example shows how to enable the generation of debug messages for shared memory library events:

```
[local]Redback#debug shmlib
```



## 1.50 debug snmp

```
debug [{boot {active | standby} | switchover}] snmp {all | general |
mib mib-description | packet | pdu {notify | request}}
```

```
no debug [{boot {active | standby} | switchover}] snmp {all | general
| mib mib-description | packet | pdu}
```

### 1.50.1 Purpose

Enables the generation of Simple Network Management Protocol (SNMP) debug messages.

### 1.50.2 Command Mode

exec (10)

### 1.50.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>mib <i>mib-description</i></b>	Optional. Enables the generation of debug messages for an SNMP MIB you identify using the <i>mib-description</i> argument. The value of the <i>mib-description</i> argument can be any one of the keywords listed in the Usage Guidelines.
<b>all</b>	Enables the generation of debug messages for all SNMP events.
<b>general</b>	Enables the generation of debug messages for general SNMP events.
<b>packet</b>	Enables the generation of debug messages for SNMP packets.
<b>pdu</b>	Enables the generation of debug messages for the protocol data unit (PDU) field in SNMP packets.



### 1.50.4 Default

The generation of debug messages for SNMP is disabled.

### 1.50.5 Usage Guidelines

Use the `debug snmp` command to enable the generation of SNMP debug messages.

---



---

## Caution!

Risk of performance loss. Enabling debug messages can severely affect system performance. To reduce the risk, use debug commands only when necessary and for as short a time as possible.

---



---

Table 4 lists the keywords to use to enable debugging for specific MIBs supported by this command.

*Table 4 Keywords for MIB Descriptions*

Keyword	MIB Description
<code>alarm</code>	Displays debugging messages for ALARM-MIB
<code>all</code>	Displays debugging messages for all MIBs
<code>aps</code>	Displays debugging messages for APS-MIB
<code>atm</code>	Displays debugging messages for ATM-MIB
<code>bgp</code>	Displays debugging messages for BGP-MIB
<code>bind</code>	Displays debugging messages for RBN-BIND-MIB
<code>bpa</code>	Displays debugging messages for BGP policy-based accounting MIB
<code>bridge</code>	Displays debugging messages for BRIDGE-MIB
<code>bulkstats</code>	Displays debugging messages for RBN-BULKSTATS-MIB
<code>cardmon</code>	Displays debugging messages for RBN-CARDMON-MIB
<code>cellpw</code>	Displays debugging messages for RBN-ATM-CELL-PW-MIB
<code>cfgfile</code>	Displays debugging messages for RBN-CONFIG-FILE-MIB
<code>circuitGroup</code>	Displays debugging messages for RBN-CIRCUIT-GROUP-MIB
<code>cpumeter</code>	Displays debugging messages for RBN-CPU-METER-MIB
<code>dhcp</code>	Display debugging messages for RBN-DHCP-MIB
<code>ds1</code>	Displays debugging messages for DS1-MIB
<code>ds3</code>	Displays debugging messages for RBN-BULKSTATS-MIB
<code>entity</code>	Displays debugging messages for ENTITY-MIB
<code>envmon</code>	Displays debugging messages for RBN-ENVMON-MIB



Table 4 Keywords for MIB Descriptions

Keyword	MIB Description
<code>etherwis</code>	Displays debugging messages for Ether-WIS-MIB
<code>fr</code>	Display debugging messages for FRAME-RELAY-DTE-MIB
<code>if</code>	Displays debugging messages for IF-MIB
<code>ip</code>	Displays debugging messages for IP-MIB
<code>ipbind</code>	Displays debugging messages for RBN-IP-BIND-MIB
<code>ipf</code>	Displays debugging messages for IP-FORWARD-MIB
<code>ippool</code>	Displays debugging messages for RBN-IPPOOL-MIB
<code>l2tp</code>	Displays debugging messages for RBN-L2TP-MIB
<code>mau</code>	Displays debugging messages for MAU-MIB
<code>memory</code>	Displays debugging messages for RBN-MEMORY-MIB
<code>ne</code>	Displays debugging messages for RBN-NOTIFY-ENHANCE-MIB
<code>ospf</code>	Displays debugging messages for OSPF-MIB
<code>pvc</code>	Displays debugging messages for RBN-PVC-MIB
<code>qos</code>	Displays debugging messages for RBN-QOS-MIB
<code>radius</code>	Displays debugging messages for RBN-RADIUS-MIB
<code>rmon</code>	Displays debugging messages for RMON-MIBs
<code>sonet</code>	Displays debugging messages for SONET-MIBs
<code>sr</code>	Displays debugging messages for RBN-SYS-RESOURCES-MIBs
<code>subscriber</code>	Displays debugging messages for RBN-SUBSCRIBER-ACTIVE-MIBs
<code>tacacs</code>	Displays debugging messages for RBN-TACACS-MIBs
<code>tcp</code>	Displays debugging messages for TCP-MIB
<code>udp</code>	Displays debugging messages for UDP-MIB
<code>vrrp</code>	Displays debugging messages for VRRP-MIB and RBN-FAST-VRRP-MIB

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

**Note:** The SmartEdge 100 router does not support standby controller cards; therefore, the `standby` and `switchover` keywords are not applicable.

To store messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the



console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about logging commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of SNMP debug messages.

### 1.50.6 Examples

The following example displays all categories of debug information for SNMP and enables the SNMP server:

```
[local]Redback#debug snmp all
```

```
[local]Redback#config
```

```
[local]Redback(config)#snmp server
```

```
[local]Redback(config)#exit
```

```
[local]Redback#
```

```
Jun 26 11:06:29: %SNMP-7-GEN: snmp process is ALIVE
```

```
Jun 26 11:06:29: %SNMP-7-GEN: snmp ready to receive packets
```



## 1.51 debug spanning-tree

```
debug [{boot {active | standby} | switchover}]
spanning-tree {all | config | packet-arg | packet-io-arg |
rcm | timers}
```

```
no debug [{boot {active | standby} | switchover}]
spanning-tree {all | config | packet-arg | packet-io-arg |
rcm | timers}
```

### 1.51.1 Purpose

Enables the generation of debug messages for the Rapid Spanning Tree Protocol (RSTP).

### 1.51.2 Command Mode

exec (10)

### 1.51.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>all</b>	Generates all debug messages related to RSTP.
<b>config</b>	Generates debug messages related to RSTP configuration events.
<b>packet-arg</b>	Generates debug messages related to the received and transmitted RSTP packets.  The syntax for the <i>packet-arg</i> argument is:  <code>packet [receive   send] [hex-dump   verbose] [type {cfg   mstp   rstp   tcn}] [trace]</code>
<b>packet-io-arg</b>	Generates debug messages related to RSTP packet input/output flow.  The syntax for the <i>packet-io-arg</i> argument is:  <code>packet-io [receive   send] [hex-dump]</code>



<code>rcm</code>	Generates debug messages for RSTP Router Configuration Manager (RCM) events.
<code>timers</code>	Generates debug messages related to RSTP timers.

#### 1.51.4 Default

The generation of debug messages for RSTP is disabled.

#### 1.51.5 Usage Guidelines

Use the `debug spanning-tree` command to enable generation of debug messages for RSTP.

---



---

### Caution!

Risk of performance loss. Enabling debug messages can severely affect system performance. To reduce the risk, use `debug` commands only when necessary and for as short a time as possible.

---



---

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

**Note:** The SmartEdge 100 router does not support standby controller cards; therefore, the `standby` and `switchover` keywords are not applicable.

To store messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or SSH session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages for RSTP.



## 1.51.6 Examples

The following example shows how to enable the generation of detailed debug messages for RSTP:

```
[local]Redback#debug spanning-tree
```



## 1.52 debug ssh

```
debug [{boot {active | standby} | switchover}] ssh {all | ssh-general
| sshd-detail | sshd-general}
```

```
no debug [{boot {active | standby} | switchover}] ssh {all |
ssh-general | sshd-detail | sshd-general}
```

### 1.52.1 Purpose

Enables the generation of debug messages for Secure Shell (SSH).

### 1.52.2 Command Mode

exec (10)

### 1.52.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<code>all</code>	Generates all debug messages related to SSH.
<code>ssh-general</code>	Generates general debug messages related to SSH.
<code>sshd-detail</code>	Generates detailed daemon (server) debug messages related to SSH.
<code>sshd-general</code>	Generates general daemon (server) debug messages related to SSH.

### 1.52.4 Default

The generation of debug messages for SSH is disabled.

### 1.52.5 Usage Guidelines

Use the `debug ssh` command to generate debug messages for the SSH.



---

---

## Caution!

Risk of performance loss. Enabling debug messages can severely affect system performance. To reduce the risk, use `debug` commands only when necessary and for as short a time as possible.

---

---

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

**Note:** The SmartEdge 100 router does not support standby controller cards; therefore, the `standby` and `switchover` keywords are not applicable.

To store messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or SSH session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages for SSH.

### 1.52.6 Examples

The following example shows how to enable the generation of detailed debug messages for SSH:

```
[local]Redback#debug sshd-detail
```



## 1.53 debug standby ospf nsr

### 1.53.1 Purpose

Enables the generation of debug messages for OSPF nonstop-routing events for the standby XCRP.

### 1.53.2 Command Mode

exec (10)

### 1.53.3 Syntax Description

This command has no keywords or arguments.

### 1.53.4 Default

The generation of debug messages is disabled.

### 1.53.5 Usage Guidelines

Use the `debug standby ospf nsr` command on the active XCRP to enable debug messages for OSPF NSR events on the standby XCRP. This command is issued on the active XCRP card, even though the output is displayed on the standby XCRP.

If debugging on the standby XCRP console is desired, console logging may need to be enabled for the standby XCRP card.

```
[local]Redback(config)#logging standby
[local]Redback(config)#context 1
[local]Redback(config-ctx)#logging console
```

---

---

### Caution!

Risk of performance loss. Enabling debug messages can severely affect system performance. Use caution when enabling debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.



To display messages in real time if you are connected to the system through the console port, use the `logging console` command in context configuration mode. If you are connected to the system through a Telnet or Secure Shell (SSH) session, use the `terminal monitor` command in exec mode. For more information about these commands, see the *Command List*.

Use the `no` form of this command to disable debug messages.

### 1.53.6 Examples

The following example enables debug messages for OSPF NSR on the standby XCRP:

```
[local]Redback#debug standby ospf nsr
```

Debug output is displayed on the standby console, tagged with [STANDBY].

```
Jun 30 17:11:56.761: %OSPF-7-NSR: [STANDBY] NSR Peer  
Active EP is available
```

```
Jun 30 17:11:56.811: [0002]: %OSPF-7-NSR: [STANDBY]  
OSPF-1: NSR enabled
```



## 1.54 debug static

```
debug [boot {active | standby} | switchover] static {all | config
| ism | rib}
```

```
no debug [boot {active | standby} | switchover] static {all |
config | ism | rib}
```

### 1.54.1 Purpose

Enables the generation of debug messages for static IP routes.

### 1.54.2 Command Mode

exec (10)

### 1.54.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>all</b>	Enables the generation of all debug message options for static IP routes.
<b>config</b>	Enables the generation of debug messages for the IP static route configuration.
<b>ism</b>	Enables the generation of debug messages for interactions between static IP routes and the Interface State and Circuit Manager (ISM).
<b>rib</b>	Enables the generation of debug messages for interactions between static IP routes and the Routing Information Base (RIB).

### 1.54.4 Default

The generation of debug messages is disabled.



## 1.54.5 Usage Guidelines

Use the `debug static` command to enable the generation of debug messages for static IP routes.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

## 1.54.6 Examples

The following example shows how to enable the generation of debug messages for interactions between static IP routes and the ISM:

```
[local]Redback#debug static ism
```



## 1.55 debug static dvsr

```
debug [boot {active | standby} | switchover] static dvsr [detail | packet]
```

```
no debug [boot {active | standby} | switchover] static dvsr [detail | packet]
```

### 1.55.1 Purpose

Enables the generation of dynamically verified static routing (DVSR) debug messages.

### 1.55.2 Command Mode

exec(10)

### 1.55.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>detail</b>	Optional. Enables the generation of detailed debug messages for DVSR activities.
<b>packet</b>	Optional. Enables the generation of debug messages for DVSR verification IP packets.

### 1.55.4 Default

The generation of debug messages is disabled.

### 1.55.5 Usage Guidelines

Use the `debug static dvsr` command to enable the generation of DVSR debug messages.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.



Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.

---

---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

## 1.55.6 Examples

The following example shows how to enable the generation of detail DVSR debug messages:

```
[local]Redback#debug static dvsr detail
```

```
Mar 28 22:45:35: %STATIC-7-DVSR_DET: dvsr node: seq_num 258, verify-addr 192.168.1.5, ping operation
Mar 28 22:45:35: %STATIC-7-DVSR_DET: + send an icmp pkt to 192.168.1.5, src 0.0.0.0, len 56, seq_num 258
Mar 28 22:45:35: %STATIC-7-DVSR_DET: dvsr node: seq_num 259, verify-addr 192.168.1.5, ping operation
Mar 28 22:45:35: %STATIC-7-DVSR_DET: + send an icmp pkt to 192.168.1.5, src 0.0.0.0, len 56, seq_num 259
```



## 1.56 debug subscriber

`debug subscriber username username`

`no debug`

### 1.56.1 Purpose

Enables the generation of subscriber debug messages for Mobile IP user name events on a home-agent (HA) instance and foreign-agent (FA) instance.

### 1.56.2 Command Mode

exec (10)

### 1.56.3 Syntax Description

<code>username</code>	Limits the generation of subscriber debug messages to a single Mobile IP subscribed identified by username or NAI.
<code>username</code>	Network Access Identifier (NAI) that identifies the user.

### 1.56.4 Default

The generation of subscriber debug messages for Mobile IP service username events on an HA instance is disabled.

### 1.56.5 Usage Guidelines

Use the `debug subscriber` command to enable the generation of subscriber debug messages for Mobile IP user name events on an HA and FA instance.

Use the `no` form of this command to disable the generation of subscriber debug messages for Mobile IP services user name events on an HA instance and FA instance.

---



---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---



---



To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

### 1.56.6 Examples

The following example show how to enable the generation of debug messages for this session to `user1@rback`:

```
[local]Redback#debug subscriber username user1@user1@rback
```

The following example shows how to enable the generation of debug messages to the GRE12 Mobile IP tunnel circuit:

```
[local]Redback#debug circuit handle 255/4:1023:63/1/1/12
```

The debug messages are not generated until the required circuit debug commands are enabled. For example, to see all Mobile IP subscriber binding events on an HA instance, you must enable the `debug circuit mobile-ip home-agent` command; for example:

```
[local]Redback#debug circuit mobile-ip home-agent binding
```



## 1.57 debug sysmon ftp

```
debug [{boot {active | standby} | switchover}] sysmon ftp
```

```
no debug [{boot {active | standby} | switchover}] sysmon ftp
```

### 1.57.1 Purpose

Enables the generation of debug messages related to transferring crash files out of the SmartEdge router using File Transfer Protocol (FTP).

### 1.57.2 Command Mode

exec (10)

### 1.57.3 Syntax Description

<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

### 1.57.4 Default

The generation of debug messages related to transferring crash files out of the SmartEdge router using FTP is disabled.

### 1.57.5 Usage Guidelines

Use the `debug sysmon ftp` command to enable the generation of debug messages related to transferring crash files out of the SmartEdge router using FTP.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.



To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

Use the `no` form of this command to disable the generation of debug messages.

### 1.57.6 Examples

The following example shows how to enable all system monitoring debugging messages:

```
[local]Redback#debug sysmon ftp
```



## 1.58 debug talk

```
debug [{boot {active | standby} | switchover}] talk
```

```
no debug [{boot {active | standby} | switchover}] talk
```

### 1.58.1 Purpose

Enables the generation of debug messages for communications with another administrator during active Telnet or Secure Shell (SSH) sessions on the same SmartEdge router.

### 1.58.2 Command Mode

exec (10)

### 1.58.3 Syntax Description

<code>boot</code>	Optional. Enables the generation of debug messages during a system reload.
<code>active</code>	Enables the generation of debug messages for the active controller card.
<code>standby</code>	Enables the generation of debug messages for the standby controller card.
<code>switchover</code>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

### 1.58.4 Default

The generation of debug messages for talk is disabled.

### 1.58.5 Usage Guidelines

Use the `debug talk` command to enable the generation of debug messages for communications with another administrator during active Telnet or SSH sessions on the same SmartEdge router.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.



To store messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or SSH session.

Use the `no` form of this command to disable the generation of debug messages for SSH.

### 1.58.6 Examples

The following example shows how to enable the generation of detailed debug messages for talk:

```
[local]Redback#debug talk
```



## 1.59 debug tunnel client

`debug tunnel client`

`no debug tunnel client`

### 1.59.1 Purpose

Enables the generation of debug messages for dynamic tunnel client events.

### 1.59.2 Command Mode

exec (10)

### 1.59.3 Syntax Description

This command has no keywords or arguments.

### 1.59.4 Default

The generation of debug messages for dynamic tunnel client events is disabled.

### 1.59.5 Usage Guidelines

Use the `debug tunnel client` command to enable the generation of debug messages for dynamic tunnel events.

Use the `no` form of this command to disable the generation of debug messages for dynamic tunnel client events.

---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the



console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

## 1.59.6 Examples

The following example shows how to limit the generation of debug messages for dynamic tunnel clients:

```
[local]Redback#debug tunnel client
```



## 1.60 debug vpls

```
debug vpls {all | cct | cfg | event | peer | sm | state}
```

```
no debug vpls {all | cct | cfg | event | peer | sm | state}
```

### 1.60.1 Purpose

Enables the generation of Virtual Private LAN Services (VPLS) debug messages.

### 1.60.2 Command Mode

exec (10)

### 1.60.3 Syntax Description

<b>all</b>	Enables the generation of all debug messages.
<b>cct</b>	Enables the generation of debug messages related to circuit and pseudo-wire processing.
<b>cfg</b>	Enables the generation of configuration-related debug messages.
<b>event</b>	Enables the generation of event messages.
<b>sm</b>	Enables the generation of debug messages related to shared memory operations.
<b>state</b>	Enables the generation of state debug messages.

### 1.60.4 Default

The generation of debug messages is disabled.

### 1.60.5 Usage Guidelines

Use the `debug vpls` command to enable the generation of VPLS debug messages.

---



---

### Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution before enabling the generation of any debug messages on a production system.

---



---



To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of debug messages.

### 1.60.6 Examples

The following example shows how to enable the generation of all VPLS debug messages:

```
[local]Redback#debug vpls all
```



## 1.61 debug vrrp

```
debug [boot {active | standby} | switchover] vrrp {all | arp [if-name
[vrrp-id]] | general | ism | packet [error | receive | send] [if-name
[vrrp-id]] | pktio [receive | send] | rcm | rpm | state [if-name [vrrp-id]]
| snmp}
```

```
no debug [boot {active | standby} | switchover] vrrp {all | arp
[if-name [vrrp-id]] | general | ism | packet [error | receive | send]
[if-name [vrrp-id]] | pktio [receive | send] | rcm | rpm | state [if-name
[vrrp-id]] | snmp}
```

### 1.61.1 Purpose

Enables the generation of Virtual Router Redundancy Protocol (VRRP) debug messages.

### 1.61.2 Command Mode

exec (10)

### 1.61.3 Syntax Description

<b>active</b>	Enables the generation of debug messages for the active controller card.
<b>all</b>	Enables the generation of all VRRP debug messages.
<b>arp</b>	Enables the generation of debug messages for Address Resolution Protocol (ARP) and VRRP interactions.
<b>boot</b>	Optional. Enables the generation of debug messages during a system reload.
<b>error</b>	Optional. Enables the generation of debug messages for packet errors only.
<b>general</b>	Enables the generation of debug messages for general operations.
<b>if-name</b>	Optional. Interface name. Enables the generation of debug messages for the specified VRRP interaction on only the named interface.
<b>ism</b>	Enables the generation of Interface and Circuit State Manager (ISM) and VRRP interactions.
<b>packet</b>	Enables the generation of debug messages for packet events.
<b>pktio</b>	Enables the generation of debug messages for packet I/O events.



<b>rcm</b>	Enables the generation of debug messages for Router Configuration Manager (RCM) and VRRP interactions.
<b>receive</b>	Optional. Enables the generation of debug messages for received packet I/O events only.
<b>rpm</b>	Enables the generation of debug messages for Router Policy Manager (RPM) and VRRP interactions.
<b>send</b>	Optional. Enables the generation of debug messages for sent packet I/O events only.
<b>snmp</b>	Enables the generation of debug messages for VRRP-MIB and RBN-FAST-VRRP-MIB.
<b>standby</b>	Enables the generation of debug messages for the standby controller card.
<b>state</b>	Enables the generation of debug messages for the virtual router state.
<b>switchover</b>	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
<b>vrrp-id</b>	Optional. Virtual router ID. The range of values is 1 to 255. Used in conjunction with the optional <i>if-name</i> argument. Enables the generation of debug messages for the specified VRRP interaction for only traffic sent to or from the specified VRRP ID over the named interface.

#### 1.61.4 Default

The generation of debug messages is disabled.

#### 1.61.5 Usage Guidelines

Use the `debug vrrp` command to enable the generation of VRRP debug messages.

Use the `boot active` or `boot standby` construct to enable debugging messages during a system reload for the active or standby controller card, respectively.

Use the `switchover` keyword to enable debugging messages while the system is switching from the active to the standby controller card.



---

---

## Caution!

Risk of performance loss. Enabling the generation of debug messages can severely affect system performance. To reduce the risk, exercise caution when enabling the generation of debug messages on a production system.

---

---

To store debug messages in the system log buffer, use the `logging debug` command (in global configuration mode). Use the `show log` command (in exec mode) to display these stored debug messages.

To display messages in real time, use the `logging console` command (in context configuration mode) if you are connected to the system through the console port. Or, use the `terminal monitor` command (in exec mode) if you are connected to the system through a Telnet or Secure Shell (SSH) session.

**Note:** For more information about `logging` commands and the `terminal monitor` command, see the *Command List*.

Use the `no` form of this command to disable the generation of VRRP debug messages.

### 1.61.6 Examples

The following example shows how to enable the generation of debug messages for all VRRP packet events:

```
[local]Redback#debug vrrp packet all
```



Commands: debug m through debug z



# Glossary

**MLPPP**

Multilink PPP.

**MP**

Multilink PPP.

**PWFQ**

Priority weighted fair queuing.