Commands: d through debug l

COMMAND DESCRIPTION

Copyright

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

1 Command Descriptions

Commands starting with “d” through “debug l” 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 dad-transmits

dad-transmits ***num-dad-transmits***

default dad-transmits

1.1.1 Purpose

Configures the number of Neighbor Solicitation (NS) messages the SmartEdge router sends to its peers for duplicate address detection (DAD) to ensure that the link-local address and the global address of the interface are unique.

1.1.2 Command Mode

ND profile configuration

1.1.3 Syntax Description

num-dad-transmits	Number of DAD NS messages to send. The range of values is 0 to 3. A value of 0 disables NS message transmission. The default value is 1.

1.1.4 Default

The default ***num-dad-transmits*** value is 1.

1.1.5 Usage Guidelines

Use the **`dad-transmit`** command to configure the number of NS messages the SmartEdge router sends to its peers for DAD to ensure that the link-local address of the SmartEdge router and the global unicast address of the interface that is bound to the circuit are unique. If the router does not receive a response after the last NS message is sent, the addresses are considered unique. For more information about DAD, see *Configuring ND*.

Use the **`default`** form of this command to revert to the default setting.

1.1.6 Examples

The following example shows how to configure a DAD transmit value of `3` for the ND profile `ndprofile7`:

```
[local]Redback(config)#context local
[local]Redback(config-ctx)#nd profile ndprofile7
[local]Redback(config-nd-profile)#dad-transmit 3
```

1.2 dampening

`dampening` [*`half-life reuse suppress max-suppress`* | **`route-map`** *`map-name`*] [**`persistent`**]

`no dampening` [*`half-life reuse suppress max-suppress`* | **`route-map`** *`map-name`*] [**`persistent`**]

1.2.1 Purpose

Enables external Border Gateway Protocol (eBGP) route dampening for the specified address family.

1.2.2 Command Mode

BGP address family configuration

1.2.3 Syntax Description

half-life	Optional. Amount of time, in minutes, after which a penalty is decreased. Once a route has been assigned a penalty, the penalty is decreased by half once the half-life period expires. The range of values is 1 to 45; the default value is 15.
reuse	Optional. Value that determines whether a route is unsuppressed and can be reused. When a penalty for a flapping route decreases to the point that it falls below this value, the route is unsuppressed and can be reused. Routes are scanned for reuse every 10 seconds. The range of values is 1 to 20,000; the default value is 750.
suppress	Optional. Value that determines if a route is suppressed. A route is suppressed when its penalty exceeds this limit. The range of values is 1 to 20,000; the default value is 2,000.
max-suppress	Optional. Maximum penalty, in minutes, that can be applied to a route. The range of values is 1 to 20,000; the default value is 4 times the value of the ***half-life*** argument. When the half life argument is left at its default value of 15 minutes, the ***max-suppress*** value defaults to 60.
route-map *map-name*	Optional. Route map name. Any set or match conditions, or both, in the specified route map are applied to BGP route dampening.
persistent	Optional. Specifies persistent route dampening, which maintains the dampening statistics for a route across peer resets.

1.2.4 Default

Route dampening is disabled. When enabled, the value for the ***half-life*** argument is 15 minutes. The value for the ***reuse*** argument is 750 minutes. The value for the ***suppress*** argument is 2,000 minutes. The value for the ***max-suppress*** argument is 4 times the value of the ***half-life*** argument.

1.2.5 Usage Guidelines

Use the **dampening** command to enable eBGP route dampening for the specified address family.

When a route from a remote peer is withdrawn, the local BGP speaker considers the withdrawn route to be a flap, and assigns a penalty of 1,000 to the route. If the remote peer sends a replacement route, the local BGP speaker assigns a penalty of 500 to the route.

Use the **no** form of this command to disable route dampening for the specified address family.

1.2.6 Examples

The following example shows how to enable route dampening:

```
[local]Redback(config)#router bgp 64000
[local]Redback(config-bgp)#address-family ipv4 multicast
[local]Redback(config-bgp-af)#dampening
```

1.3 deadtime

`deadtime` ***`minutes`***

`default deadtime`

1.3.1 Purpose

Sets the minimum amount of time during which any “dead” peer in the group is ignored when computing the Layer 2 Tunneling Protocol (L2TP) group algorithm.

1.3.2 Command Mode

L2TP group configuration

1.3.3 Syntax Description

`minutes`	Minimum number of minutes that a peer is marked as `dead`. The range of values is 1 to 300; the default value is 5.

1.3.4 Default

The deadtime is set to five minutes.

1.3.5 Usage Guidelines

Use the **`deadtime`** command to set the minimum amount of time during which any “dead” peer in the group is ignored when computing the L2TP group algorithm.

A peer is labeled “dead” after it is determined that a new tunnel cannot be established to the peer. This feature prevents an L2TP peer that is not functioning fully from being inundated with connection attempts without disconnecting the peer altogether. It also allows you to identify peers that are not fully functional.

A peer remains labeled as “dead” until a new session is established to it as follows:

- After the deadtime is expired and a connection request arrives, the peer is again considered as a destination, according to the group algorithm.
- If a connection attempt is not made to the peer (the peer is not selected as the destination), the “dead” label is not removed.
- If a connection attempt is made and is successful, the “dead” label is removed from the peer; if the attempt is not successful, the deadtime is again applied to the peer.

  Note: Current sessions to the peer are not brought down if the peer is labeled “dead”. Only attempts to add new tunnels are affected.

A “dead” peer is labeled as “dead” in the output of the **show l2tp peer** command in any mode for at least the length of time indicated in the ***minutes*** argument.

Use the **no** or **default** form of this command to set the deadtime to two minutes.

1.3.6 Examples

The following example shows how to select (or create) an L2TP group and set the number of deadtime minutes to two:

```
[local]Redback(config-ctx)#l2tp-group name group1

[local]Redback(config-l2tp-group)#default deadtime
```

The following example shows how to select (or create) an L2TP group and set the number of deadtime minutes to `10`:

```
[local]Redback(config-ctx)#l2tp-group name group1

[local]Redback(config-l2tp-group)#deadtime 10
```

1.4 debug aaa

debug [{**boot** {**active** | **standby**} | **switchover**}] **aaa** {**acct** | **all** | **authen** | **author** | **config** | **db** | **general** | **ip-pool** | **rad-packet** | **radius** | **tacacs+** | **users**}

no debug [{**boot** {**active** | **standby**} | **switchover**}] **aaa** {**acct** | **all** | **authen** | **author** | **config** | **db** | **general** | **ip-pool** | **rad-packet** | **radius** | **tacacs+** | **users**}

1.4.1 Purpose

Enables the generation of accounting, authorization, and authentication (AAA) debug messages.

1.4.2 Command Mode

exec (10)

1.4.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
`acct`	Enables the generation of AAA accounting debug messages.
`all`	Enables the generation of all AAA debug messages.
`authen`	Enables the generation of AAA authentication debug messages.
`author`	Enables the generation of AAA authorization debug messages.
`config`	Enables the generation of AAA configuration debug messages.
`db`	Enables the generation of AAA database debug messages.
`general`	Enables the generation of general AAA debug messages.
`ip-pool`	Enables the generation of AAA IP pool debug messages.
`rad-packet`	Enables the generation of RADIUS packet debug messages.
`radius`	Enables the generation of general RADIUS debug messages.
`tacacs+`	Enables the generation of Terminal Access Controller Access Control System Plus (TACACS+) debug messages.
`users`	Enables the generation of AAA user authentication debug messages.

1.4.4 Default

The generation of debug messages is disabled.

1.4.5 Usage Guidelines

Use the **debug aaa** command to enable the generation of AAA debug messages on an SmartEdge router.

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

Note: You can enter this command multiple times to enable the generation of various types of AAA debug messages.

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 debug messages on a production system.

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

1.4.6 Examples

The following example shows how to enable generation of all AAA debug messages:

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

1.5 debug aaa tacacs+

debug [{**boot** {**active** | **standby**} | **switchover**}] **aaa tacacs+** {**accounting** | **all** | **attributes** | **authentication** | **authorization** | **general** | **packet** | **server** | **session**}

no debug [{**boot** {**active** | **standby**} | **switchover**}] **aaa tacacs+** {**accounting** | **all** | **attributes** | **authentication** | **authorization** | **general** | **packet** | **server** | **session**}

1.5.1 Purpose

Enables the generation of Terminal Access Controller Access Control System Plus (TACACS+) debug messages.

1.5.2 Command Mode

exec (10)

1.5.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
`accounting`	Enables the generation of debug messages regarding the progress of TACACS+ accounting sessions.
`all`	Enables the generation of all TACACS+ debug log messages.
`attributes`	Enables the generation of debug messages displaying the values of any TACACS+ authorization attributes received.
`authentication`	Enables the generation of debug messages regarding the progress of TACACS+ authentication sessions.
`authorization`	Enables the generation of debug messages regarding the progress of TACACS+ authorization sessions.
`general`	Enables the generation of debug messages regarding the progress of general configuration operations.
`packet`	Enables the generation of debug messages for TACACS+ packets sent and received.
`server`	Enables the generation of debug messages for changes in TACACS+ server state and server connection status.
`session`	Enables the generation of debug messages for TACACS+ sessions and their states.

1.5.4 Default

The generation of TACACS+ debug messages is disabled.

1.5.5 Usage Guidelines

Use the **debug aaa tacacs+** command to enable the generation of TACACS+ debug messages. You can enter this command multiple times, specifying different types of debugging 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 TACACS+ debug messages.

1.5.6 Examples

The following example shows how to enable generation of TACACS+ debug messages for authorization attributes:

```
[local]Redback#debug aaa tacacs+ attributes
```

1.6 debug ancp

debug [{**boot** {**active** | **standby**} | **switchover**}] **ancp packet** ***event-type***

no debug [{**boot** {**active** | **standby**} | **switchover**}] **ancp packet**

1.6.1 Purpose

Enables the generation of debug messages for Access Node Control Protocol (ANCP) packets.

1.6.2 Command Mode

exec (10)

1.6.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

standby Enables the generation of debug messages for the standby controller card.

switchover Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

ancp Enables the generation of debug messages for ANCP packets.

event-type Type of ANCP events or data for which the generation of debug messages is enabled. See Table 1 for a list of event types and data.

1.6.4 Default

The generation of debug messages for ANCP packets is disabled.

1.6.5 Usage Guidelines

Use the **debug ancp** command to enable the generation of debug messages for ANCP packets.

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.

Table 1 lists the keywords for the types of ANCP events or data for which you can enable the generation of debug messages.

Table 1 ANCP Event Types and Data

Event or Data Type	Description
`aaa`	Authentication, authorization, and accounting (AAA) interaction
`adj`	Adjacency protocol
`all`	Management class packet
`cli`	CLI interaction related to ANCP
`event`	Event class packet
`general`	General debugging
`header`	Packet header
`io`	Packet input and output
`ism`	Interface and Circuit State Manager (ISM) interaction
`mgmt`	Management class packet
`neighbor`	Neighbor event
`packet`	Received packet dump
`stat`	Packet statistics
`timer`	Timer event
`tlv`	Packet Tag-Length-Value (TLV)

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 ANCP packets.

1.6.6 Examples

The following example shows how to enable generation of debug messages for ANCP packets:

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

1.7 debug arp

debug [**boot** {**active** | **standby**} | **switchover**] **arp** [**all** | **config** | **event** [**prefix-list** *pl-name*] | **lc** [**prefix-list** *pl-name*] | **rib** [**prefix-list** *pl-name*] | **vrrp**]

no debug [**boot** {**active** | **standby**} | **switchover**] **debug arp** [**all** | **config** | **event** [**prefix-list** *pl-name*] | **lc** [**prefix-list** *pl-name*] | **rib** [**prefix-list** *pl-name*] | **vrrp**]

1.7.1 Purpose

Enables the generation of Address Resolution Protocol (ARP) debug messages for the current context.

1.7.2 Command Mode

exec (10)

1.7.3 Syntax Description

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

`all`	Optional. Enables the generation of all types of ARP debug messages.
`config`	Optional. Enables the generation of ARP configuration debug messages.
`event`	Optional. Enables the generation of ARP event debug messages.
`prefix-list` ***`pl-name`***	Optional. Prefix list name. Used in conjunction with the **`event`**, **`lc`**, and **`rib`** keywords.
`lc`	Optional. Enables the generation of ARP traffic card event debug messages.
`rib`	Optional. Enables the generation of ARP Routing Information Base (RIB) debug messages.
`vrrp`	Optional. Enables the generation of ARP Virtual Router Redundancy Protocol (VRRP) event debug messages.

1.7.4 Default

The generation of ARP debug messages is disabled. If you use this command without any optional syntax, only the generation of ARP event debug messages is enabled.

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

1.7.5 Usage Guidelines

Use the **`debug arp`** command to enable the generation of ARP debug messages for the current context.

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 debug messages on a production system.

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

1.7.6 Examples

The following example shows how to enable generation of ARP VRRP event debug messages:

```
[local]Redback#debug arp vrrp
```

1.8 debug as-path-list

debug [**boot** {**active** | **standby**} | **switchover**] **as-path-list**

no debug [**boot** {**active** | **standby**} | **switchover**] **as-path-list**

1.8.1 Purpose

Enables the generation of debug messages for the maintenance of autonomous system (AS) path lists and the comparison of Border Gateway Protocol (BGP) AS path attributes with AS path lists.

1.8.2 Command Mode

exec (10)

1.8.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

`standby` Enables the generation of debug messages for the standby controller card.

`switchover` Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

1.8.4 Default

None

1.8.5 Usage Guidelines

Use the **`debug as-path-list`** command to enable the generation of debug messages for the maintenance of AS path lists and the comparison of BGP AS path attributes with AS path lists.

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.

1.8.6 Examples

The following example shows how to enable generation of debug messages for AS path lists:

```
[local]Redback#debug as-path-list
```

1.9 debug atm oam

debug [**boot** {**active** | **standby**} | **switchover**] **atm oam** ***slot***/***port*** **vpi** ***vpi*** **vci** ***vci***

no debug [**boot** {**active** | **standby**} | **switchover**] **atm oam** ***slot***/***port*** **vpi** ***vpi*** **vci** ***vci***

1.9.1 Purpose

Enables the display of operations, administration, and maintenance (OAM) cells for a specific Asynchronous Transfer Mode (ATM) permanent virtual circuit (PVC).

1.9.2 Command Mode

exec (10)

1.9.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

standby Enables the generation of debug messages for the standby controller card.

switchover Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

slot Chassis slot number of the ATM traffic card with the port to be tested.

port Port number of the ATM port to be tested.

vpi ***vpi*** Virtual path identifier (VPI). The range of values is 0 to 255.

vci ***vci*** Virtual channel identifier (VCI). The range of values is 1 to 65535. By convention, values 1 to 30 are reserved for system use; however, VCIs 3 and 4 are solely for system use, and you must not assign them to carry user data.

1.9.4 Default

The generation of debug messages is disabled for all ATM PVCs.

1.9.5 Usage Guidelines

Use the **debug atm oam** command to enable the display of OAM cells for a specific ATM PVC. The first 40 bytes of each generated OAM cell displays.

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 **standby** and **switchover** keywords are not available for the SmartEdge 100 router.

Note: The SmartEdge 100 router limits the value of the *slot* argument to 2.

Note: The value for the *port* argument on the SmartEdge 100 router depends on the MIC slot in which the ATM OC MIC is installed.

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.

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 a specific ATM PVC.

1.9.6 Examples

The following example shows how to enable the display of OAM cells for ATM PVC with VPI `0` and VCI `100` on port `1` in slot `3`:

```
[local]Redback#debug atm oam 3/1 vpi 0 vci 100
```

1.10 debug bfd

debug bfd {**ipv4** | **ipv6**} {**all** | **client** | **config** | **session**}

[no] debug bfd {**ipv4** | **ipv6**} {**all** | **client** | **config** | **session**}

1.10.1 Purpose

Enables the generation of Bidirectional Forwarding Detection (BFD) debug messages.

1.10.2 Command Mode

exec (10)

1.10.3 Syntax Description

all	Enables the generation of BFD client, configuration, and session debug messages for both IPv4 and IPv6 address families.
client	Enables the generation of BFD client debug messages.
config	Enables the generation of BFD configuration debug messages.
session	Enables the generation of BFD session debug messages.
ipv4	Enables the generation of BFD debug messages for IPv4 addresses.
ipv6	Enables the generation of BFD debug messages for IPv6 addresses.

1.10.4 Default

The generation of debug messages is disabled.

1.10.5 Usage Guidelines

Use the **debug bfd** command to generate BFD debug messages. If neither the **ipv4** or **ipv6** keyword is present, debug messages are generated for both address families. If the **all** keyword is used, client, configuration, and session debug messages are generated for all address families.

Caution!

Risk of performance loss. Enabling debug messages can severely affect system performance. Use 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. To display the messages, use the **show log** command in exec mode.

To display messages in real time, 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 the generation of debug messages.

1.10.6 Examples

The following example enables the generation of BFD session debug messages:

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

```
[local]Redback#debug bfd ipv6 session
```

1.11 debug bgp bestpath

debug bgp bestpath [*ip-prefix*]

no debug bgp bestpath [*ip-prefix*]

1.11.1 Purpose

Enables the generation of messages for BGP best path events.

1.11.2 Command Mode

exec (10)

1.11.3 Syntax Description

ip-prefix	Optional. Enables the generation of BGP best path event messages for a specific IPv4 or IPv6 prefix.

1.11.4 Default

The generation of debug messages is disabled.

1.11.5 Usage Guidelines

Use the **debug bgp bestpath** command to enable the generation of messages for BGP best path events for the current context. The output of the **debug bgp bestpath** command includes information about how and why a BGP was selected and the length of time the BGP routing process took to perform best path calculation. This information can be useful when troubleshooting BGP. Enter this command without the optional *ip-prefix* keyword to enable the generation of BGP best-path debug messages for all IP prefixes configuring on this router.

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.

1.11.6 Examples

The following example shows how to enable the generation of messages for all BGP best path events:

```
[local]Redback#debug bgp bestpath
```

1.12 debug bgp event

debug [**boot** {**active** | **standby**} | **switchover**] **bgp event**

no debug [**boot** {**active** | **standby**} | **switchover**] **bgp event**

1.12.1 Purpose

Enables the generation of messages for all general Border Gateway Protocol (BGP) events.

1.12.2 Command Mode

exec (10)

1.12.3 Syntax Description

`boot` Optional. Enables the generation of debug messages during a system reload.

`active` Enables the generation of debug messages for the active controller card.

`standby` Enables the generation of debug messages for the standby controller card.

`switchover` Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

1.12.4 Default

The generation of debug messages is disabled.

1.12.5 Usage Guidelines

Use the **debug bgp event** command to enable the generation of messages for all general BGP 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.

1.12.6 Examples

The following example shows how to enable the generation of messages for all general BGP events:

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

1.13 debug bgp listen

debug [**boot** {**active** | **standby**} | **switchover**] **bgp listen**

no debug [**boot** {**active** | **standby**} | **switchover**] **bgp listen**

1.13.1 Purpose

Enables the generation of debug messages for Border Gateway Protocol (BGP) passive open connections.

1.13.2 Command Mode

exec (10)

1.13.3 Syntax Description

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

1.13.4 Default

The generation of debug messages is disabled.

1.13.5 Usage Guidelines

Use the **`debug bgp listen`** command to enable the generation of debug messages for BGP passive open connections, where the remote peer connects to port 179 on the local 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.

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.

1.13.6 Examples

The following example enables the generation of debug messages for BGP passive open connections:

```
[local]Redback#debug bgp listen
```

1.14 debug bgp message

debug [**boot** {**active** | **standby**} | **switchover**] **bgp** [*ip-addr*] **message** [**keepalive** | **notification** | **open** | **refresh**]

no debug [**boot** {**active** | **standby**} | **switchover**] **bgp** [*ip-addr*] **message** [**keepalive** | **notification** | **open** | **refresh**]

1.14.1 Purpose

Enables the generation of debug messages for Border Gateway Protocol (BGP) non-update events.

1.14.2 Command Mode

exec (10)

1.14.3 Syntax Description

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

switchover Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

ip-addr Optional. IP address of the neighbor for which debug messages are generated.

message Enables the generation of debug messages for BGP non-update events.

keepalive Optional. Enables the generation of only BGP keepalive messages.

notification Optional. Enables the generation of only BGP notification messages.

open Optional. Enables the generation of only BGP open messages.

refresh Optional. Enables the generation of only BGP refresh messages.

1.14.4 Default

The generation of debug messages is disabled.

1.14.5 Usage Guidelines

Use the **debug bgp message** command to enable the generation of debug messages for BGP non-update 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.

1.14.6 Examples

The following example shows how to enable generation of debug messages for BGP non-update messages for the BGP neighbor, `10.1.1.1:`

```
[local]Redback#debug bgp 10.1.1.1 message
```

1.15 debug bgp policy

debug [**boot** {**active** | **standby**} | **switchover**] **bgp policy**

no debug [**boot** {**active** | **standby**} | **switchover**] **bgp policy**

1.15.1 Purpose

Enables the generation of debug messages for Border Gateway Protocol (BGP) routing policies.

1.15.2 Command Mode

exec (10)

1.15.3 Syntax Description

`boot` Optional. Enables the generation of debug messages during a system reload.

`active` Enables the generation of debug messages for the active controller card.

`standby` Enables the generation of debug messages for the standby controller card.

`switchover` 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 bgp policy`** command to enable the generation of debug messages for BGP routing policies.

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 BGP routing policies:

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

1.16 debug bgp rib

debug [**boot** {**active** | **standby**} | **switchover**] **bgp rib** {**download** | **redistribution**} [**prefix-list** *pl-name*]

no debug [**boot** {**active** | **standby**} | **switchover**] **bgp rib** {**download** | **redistribution**} [**prefix-list** *pl-name*]

1.16.1 Purpose

Enables the generation of debug messages for interaction between the Border Gateway Protocol (BGP) and the Routing Information Base (RIB).

1.16.2 Command Mode

exec (10)

1.16.3 Syntax Description

boot	Optional. Enables the generation of debug messages during a system reload.
active	Enables the generation of debug messages for the active controller card.
standby	Enables the generation of debug messages for the standby controller card.
switchover	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
download	Enables the logging only RIB download debug messages.
redistribution	Enables the generation of only RIB redistribution debug messages.
prefix-list *pl-name*	Optional. RIB debug messages that are allowed by the specified prefix list.

1.16.4 Default

The generation of debug messages is disabled.

1.16.5 Usage Guidelines

Use the **debug bgp rib** command to enable the generation of debug messages for interactions between the BGP routing process 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.

1.16.6 Examples

The following example shows how to enable generation of debug messages for BGP RIB events:

```
[local]Redback#debug bgp rib
```

1.17 debug bgp session-state

debug [**boot** {**active** | **standby**} | **switchover**] **bgp** [*ip-addr*] **session-state**

no debug [**boot** {**active** | **standby**} | **switchover**] **bgp** [*ip-addr*] **session-state**

1.17.1 Purpose

Enables the generation of debug messages for Border Gateway Protocol (BGP) session states and timers.

1.17.2 Command Mode

exec (10)

1.17.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
`ip-addr`	Optional. BGP neighbor IP address. Enables the generation of debug messages for only the specified neighbor.

1.17.4 Default

Need default information

1.17.5 Usage Guidelines

Use the **`debug bgp session-state`** command to enable the generation of debug messages for BGP session states and timers.

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 generation of debug messages for session states and timers for the BGP neighbor, `10.1.1.1:`

```
[local]Redback#debug bgp 10.1.1.1 session-state
```

1.18 debug bgp update

`debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`bgp`** [*`ip-addr`*] **`update`** [**`prefix-list`** *`pl-name`* [**`in`** | **`out`**] | **`in`** | **`out`** [**`summary`**]]

`no debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`bgp`** [*`ip-addr`*] **`update`** [**`prefix-list`** *`pl-name`* [**`in`** | **`out`**] | **`in`** | **`out`** [**`summary`**]]

1.18.1 Purpose

Enables the generation of debug messages for Border Gateway Protocol (BGP) updates.

1.18.2 Command Mode

exec (10)

1.18.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
`ip-addr`	Optional. IP address in the form *`A.B.C.D`*. Enables the generation of BGP updates for only the specified neighbor.
`prefix-list` *`pl-name`*	Optional. Prefix list name. Enables the generation of only BGP updates that match those defined in the specified prefix list.
`in`	Optional. Enables the generation of only incoming BGP updates. When used with the **`prefix-list`** *`pl-name`* construct, enables the generation of only incoming BGP updates that match those defined in the specified prefix list.
`out`	Optional. Enables the generation of only outgoing BGP updates. When used with the **`prefix-list`** *`pl-name`* construct, enables the generation of only outgoing BGP updates that match those defined in the specified prefix list.
`summary`	Optional. Enables the generation of summary information for outbound updates.

1.18.4 Default

The generation of debug messages is disabled.

1.18.5 Usage Guidelines

Use the **`debug bgp update`** command to enable the generation of messages for BGP update events.

We recommend the use of the **`prefix-list`** *`pl-name`* construct when there is a large number of routes in the BGP routing table.

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 generation of debug messages for BGP updates for the BGP neighbor, `10.1.1.1:`

```
[local]Redback#debug bgp 10.1.1.1 update
```

1.19 debug bridge

`debug [boot {active | standby} | switchover] bridge {ageing | config | loop-detection | table}`

1.19.1 Purpose

Enables the generation of debug messages for bridge-related events and entities.

1.19.2 Command Mode

exec (10)

1.19.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
`ageing`	Enables the generation of debug messages for the aging of the bridge forwarding entries (BFEs).
`config`	Enables the generation of debug messages for bridging configuration events.
`loop-detection`	Enables the generation of debug messages for loop-detection events.
`table`	Enables the generation of debug messages for learned medium access control (MAC) addresses.

1.19.4 Default

None

1.19.5 Usage Guidelines

Use the **`debug bridge`** command to enable the generation of debug messages for bridge-related events and entities.

The output for loop detections indicate the following information. Output can be extensive and impact the performance of a production system. Be cautious when enabling bridge debugging messages on large configurations.

- `%s LD Disabled CCT %s in bridge %s\n`
- State of sending circuit (cct)
- Upticks/retry timeouts
- Whether a circuit is blocked and the retry timeout
- The LD state sent
- LD state change log with from and to state
- Nonblockable circuits

- The circuit chosen for blocking, with its priority/move count and if the threshold is exceeded

- MAC move counts

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

1.19.6 Examples

The following example shows how to enable the generation of debug messages for the aging of the BFEs:

```
[local]Redback#debug bridge ageing
```

1.20 debug cfm

`debug cfm` {**`all`** | **`config`** | **`ipc`** | **`ism`** | **`misc`** | **`packet`** | **`ppa`** | **`rcm`**}

1.20.1 Purpose

Enables the generation of debug logs for connectivity fault management (CFM) related events.

1.20.2 Command Mode

exec (10)

1.20.3 Syntax Description

`all`	Enables logging of all of the following debugging information.
`config`	Enables logging of changes in CFM configuration.
`ipc`	Enables logging of interprocess communication (IPC) messages.
`ism`	Enables logging of Interface and Circuit State Manager (ISM) received and sent messages.
`misc`	Enables logging of changes in other CFM- related events.
`packet`	Enables logging of CFM protocol data unit (PDU) processing.
`ppa`	Enables logging of messages to and from the Packet Processing ASIC (PPA).
`rcm`	Enables logging of Router Configuration Manager (RCM) level events; namely, database add, delete, and update events, configuration changes, and other similar configuration management events.

1.20.4 Default

None

1.20.5 Usage Guidelines

Use the **debug cfm** command to enable the generation of debug logs for CFM related events.

1.20.6 Examples

The following example shows how to enable logging of CFM PDU processing:

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

1.21 debug circuit

For Asynchronous Transfer Mode (ATM) OC, Ethernet, or Packet over SONET/SDH (POS) traffic cards the syntax is:

debug circuit{*slot/port* [*circuit-id*]} | {*facility category*}

no debug circuit {*slot/port* [*circuit-id*]} | {*facility category*}

For all circuits, to disable circuit-level debugging, the syntax is:

no debug circuit **all**

For all circuits, to disable circuit-level debugging, the syntax is:

no debug circuit all

1.21.1 Purpose

Enables the generation of debug messages for one or more circuits in the system.

1.21.2 Command Mode

exec (10)

1.21.3 Syntax Description

slot	Chassis slot number of a traffic card with the port for which debug messages are enabled.
port	Port number of the port for which debug messages are enabled.
chan-num	Optional. Channel number for which debug messages are enabled. If omitted, enables debug messages for all channels on the specified port. The range of values depends on the type of port.
sub-chan-num	Optional. Subchannel number on the channel for which debug messages are enabled. If omitted, enables debug messages for all subchannels on the specified channel. The range of values depends on the type of port.
circuit-id	Optional. Circuit identifier, according to one of the constructs listed in Table 2. If omitted, enables debug messages for all circuits on the specified port or channel.
facility	Facility for which debug messages are enabled; supported facilities listed in Table 3. Facilities can be further divided into categories; one facility may include many categories.
category	Second level of organization inside facilities.

1.21.4 Default

The generation of debug messages for any circuit is disabled.

1.21.5 Usage Guidelines

Use the **debug circuit** command to enable the generation of debug messages for one or more circuits in the system.

Note: The SmartEdge 100 router limits the value of the *slot* argument to 2.

Note:

The value for the *port* argument on the SmartEdge 100 router is either of the following:

- For a native port, it is either 1 or 2.
- For a MIC port, it depends on the type of MIC and the slot in which it is installed.

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.

To enable the generation of debug messages for specific circuits, follow this two-step process:

1. Decide which type of debug messages to log for the circuits. You can use the ***facility*** and ***category*** arguments as a construct. For instance, to log all incoming and outgoing PPP packets, use **ppp** as the value for ***facility*** argument and **packet** as the value for the ***category*** argument. Not all facilities have attendant categories.

2. Decide which circuits to debug.For example, to enable debugging on only the circuit on slot 1, port 2, channel 1, subchannel 1, use **1/2** as the value of the ***slot/port*** argument, **:1** as the value of the ***channel*** argument, and **:1** as the value of the ***subchannel*** argument.

General debugging provides wide coverage. By enabling debug messages for specific circuits, you receive only debug messages related to the circuits of interest.

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*.

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

Table 2 Values for the circuit-id Argument

Construct	Description
dlci ***dlci***	Data-link connection identifier (DLCI) for the Frame Relay PVC. The range of values is 16 to 991.

Table 2 Values for the circuit-id Argument

Construct	Description
vlan-id ***vlan-id***	Virtual LAN (VLAN) tag value for an 802.1Q tunnel or PVC. The ***vlan-id*** argument is one of the following constructs: • ***pvc-vlan-id***—VLAN tag value of a PVC that is not within an 802.1Q tunnel. • ***tunl-vlan-id***—VLAN tag value of a tunnel. • ***tunl-vlan-id*:*pvc-vlan-id***—VLAN tag value for the tunnel followed by the VLAN tag value for the PVC within the tunnel. The range of values for any VLAN tag value is 1 to 4095.
vpi-vci ***vpi vci***	Virtual path identifier (VPI) and virtual circuit identifier (VCI) for an ATM permanent virtual circuit (PVC). The range of values is 0 to 255 and 1 to 65535, respectively.

If you specify the VLAN tag value for an 802.1Q tunnel, the command generates debug messages for the tunnel, but not the PVCs within the tunnel.

Table 3 lists the supported facilities for the ***facility*** argument. Not all facilities are available for all debug commands, and not all facilities have an “all” option.

Note: The SmartEdge 100 router does not support the **chdlc** and **fr** keywords.

Table 3 Supported Facilities

Keyword	Description
aaa	Limits the debug output to Authentication, Authorization, and Accounting (AAA) messages.
arp	Limits the debug output to Address Resolution Protocol (ARP) messages.
as-path-limit	Limits the debug output to Autonomous System (AS) path limit messages.
atm	Limits the debug output to ATM messages.
bfd	Limits the debug output to BFD messages.
bgp	Limits the debug output to Border Gateway Patrol (BGP) messages.
bridge	Limits the debug output to bridge messages.
clips	Limits the debug output to clientless IP service selection (CLIPS) messages.
cls	Limits the debug output to community list messages.
context	Limits the debug output to context messages.

Table 3 Supported Facilities

Keyword	Description
`dhcp-relay`	Limits the debug output to Dynamic Host Configuration Protocol (DHCP) relay messages.
`dhcp-server`	Limits the debug output to DHCP server messages.
`dot1q`	Limits the debug output to 802.1Q messages.
`ether`	Limits the debug output to Ethernet messages.
`ext-community-list`	Limits the debug output to extended community list messages.
`fm` or `frame-relay`	Limits the debug output to Frame Relay messages.
`gre`	Limits the debug output to Generic Routing Encapsulation (GRE) tunnel messages.
`hr`	Limits the debug output to HTTP redirect messages.
`if`	Limits the debug output to interface messages.
`igmp`	Limits the debug output to Internet Group Management Protocol (IGMP) messages.
`ip`	Limits the debug output to IP messages.
`iprwlock`	Limits the debug output to Interprocess locks messages.
`ipv6`	Limits the debug output to IP version 6 (IPv6) messages.
`isis`	Limits the debug output to Intermediate System-to-Intermediate System (ISIS) messages.
`key-chain`	Limits the debug output to Key chain messages.
`l2vpn`	Limits the debug output to Layer 2 Virtual Private Network (L2VPN) messages.
`ldp`	Limits the debug output to Label Distribution Protocol (LDP) messages.
`lg`	Limits the debug output to LG messages.
`lm`	Limits the debug output to label-manager messages.
`logger-rcm`	Limits the debug output to logger's RCM messages.
`mpls-static`	Limits the debug output to Multiprotocol Label Switching (MPLS) Static messages.
`msdp`	Limits the debug output to Multicast Source Discovery Protocol (MSDP) messages.
`nat`	Limits the debug output to Network Address Translation (NAT) messages.
`nd`	Limits the debug output to ND messages.
`ntp`	Limits the debug output to Network Time Protocol (NTP) messages.

Table 3 Supported Facilities

Keyword	Description
`ospf`	Limits the debug output to open shortest path first (OSPF) messages.
`ospf3`	Limits the debug output to OSPF version 3 messages.
`pedgr`	Limits the debug output to provider edge router (PEDGR) messages.
`pim`	Limits the debug output to Protocol Independent Multicast (PIM) messages.
`pm`	Limits the debug output to performance-monitoring messages.
`policy`	Limits the debug output to policy messages.
`ppp`	Limits the debug output to Point-to-Point Protocol (PPP) messages.
`pppoe`	Limits the debug output to PPP over Ethernet (PPPoE) messages.
`qos`	Limits the debug output to quality of service (QoS) messages.
`rcm`	Limits the debug output to router configuration manager (RCM) messages.
`rip`	Limits the debug output to Routing Information Protocol (RIP) messages.
`route-map`	Limits the debug output to route map messages.
`rsvp`	Limits the debug output to RSVP messages.
`shmlib`	Limits the debug output to shared memory library messages.
`snmp`	Limits the debug output to Simple Network Management Protocol (SNMP) messages.
`sysmon`	Limits the debug output to system monitoring messages.
`talk`	Limits the debug output to talk messages.
`tunnel`	Limits the debug output to tunnel messages.
`vpls`	Limits the debug output to Virtual Private LAN Services (VPLS) messages.
`vrrp`	Limits the debug output to Virtual Router Redundancy Protocol (VRRP) messages.

Use the `no` forms of this command to disable the generation of debug messages for one or more circuits:

- **`no debug circuit`** ***`slot/port circuit-id`*** disables circuit-level debugging only for the specified circuit. All other circuit-level debugging remains in place.

- **`no debug circuit`** ***`facility category`*** disables circuit-level debugging for the specified category within the specified facility.

- **`no debug circuit`** `all` disables circuit-level debugging for all circuits, including all facilities and categories. This command restores the system to the default settings.

1.21.6 Examples

The following example show how to enable the generation of all PPP debug messages for all circuits on port `1` on the traffic card in slot `3`:

```
[local]Redback#debug circuit 3/1
[local]Redback#debug circuit ppp all
```

The following example shows how to enable the generation of AAA debug messages for only subchannel `1` on channel `2` of slot `1`, port `1`:

```
[local]Redback#debug circuit 1/1 :2 :1
[local]Redback#debug circuit aaa all
```

1.22 debug circuit mobile-ip

`debug circuit mobile-ip` [**`all`** | **`agent-common`** | **`authentication`** | **`foreign-agent`** | **`home-agent`** | **`interaction`** | **`packet`** | **`packet-io`**]

`no debug circuit mobile-ip`

1.22.1 Purpose

Enables the generation of debug messages for Mobile IP services on a circuit.

1.22.2 Command Mode

exec

1.22.3 Syntax Description

`all`	Optional. Enables the generation of all debugging messages to Mobile IP events on a circuit.
`agent-common`	Optional. Limits the generation of debug messages to events common to both the HA and FA.
`authentication`	Optional. Limits the generation of debugging messages to Mobile IP authentication events on a circuit.

`foreign-agent` Optional. Limits the generation of debugging messages to Mobile IP FA instance events on a circuit.

`home-agent` Optional. Limits the generation of debugging messages to Mobile IP HA instance events on a circuit.

`interaction` Optional. Limits the generation of debugging messages to Mobile IP module interaction events, such as Router Configuration Manager (RCM) events and Interface and Circuit State Manager (ISM) events on a circuit.

`packet` Optional. Limits the generation of debugging messages to Mobile IP send and receive packets events on a circuit.

`packet-io` Optional. Limits the generation of debugging messages to Mobile IP I/O events on a circuit.

1.22.4 Default

The generation of debug messages for Mobile IP services on a circuit is disabled.

1.22.5 Usage Guidelines

Use the **`debug circuit mobile-ip`** command to enable the generation of debug messages for Mobile IP services on a circuit. To use this command, do the following:

1. Issue the **`debug circuit handle`** command.

2. Issue the **`debug mobile-ip`** command.

When you use this command without specifying an option, the system generates debug messages for all Mobile IP events on a circuit.

Use the **`no`** form of this command to disable the generation of debug messages for HAs and FAs on a circuit.

For information about the **`debug circuit`** 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.

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*.

1.22.6 Examples

The following example shows how to limit the generation of debug messages to circuit handle `255/4:1023:63/1/1/12` for an FA instance on a circuit `GRE 12:`

```
[local]Redback#debug circuit handle 255/4:1023:63/1/1/12

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

1.23 debug clips

`debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`clips`** {**`all`** | **`authentication`** | **`be-cli`** | **`cct`** | **`cli`** | **`dhcp`** | **`fsm`** | **`ism`** | **`rcm`** | **`timer`**}

`no debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`clips`** {**`all`** | **`authentication`** | **`be-cli`** | **`cct`** | **`cli`** | **`dhcp`** | **`fsm`** | **`ism`** | **`rcm`** | **`timer`**}

1.23.1 Purpose

Enables the generation of clientless IP service selection (CLIPS) debug messages.

1.23.2 Command Mode

exec

1.23.3 Syntax Description

`boot` Optional. Enables the generation of debug messages during a system reload.

`active` Enables the generation of debug messages for the active controller card.

`standby` Enables the generation of debug messages for the standby controller card.

switchover Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

all Enables the generation of debug messages for all CLIPS events.

authentication Enables the generation of debug messages for CLIPS session authentication.

be-cli Enables the generation of debug messages for CLIPS backend command-line interface (CLI) events.

cct Enables the generation of debug messages for CLIPS circuit events.

cli Enables the generation of debug messages CLIPS CLI events.

dhcp Enables the generation of debug messages for CLIPS Dynamic Host Configuration Protocol (DHCP) events.

fsm Enables the generation of debug messages for CLIPS Finite State Machine (FSM) events.

ism Enables the generation of debug messages for CLIPS Interface and Circuit State Manager (ISM) events.

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

timer Enables the generation of debug messages for CLIPS timer events.

1.23.4 Default

The generation of CLIPS debug messages is disabled.

1.23.5 Usage Guidelines

Use the **debug clips** command to enable the generation of CLIPS 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.

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 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 CLIPS debug messages about all CLIPS details:

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

1.24 debug cls

`debug` [{**`boot`** {**`active`** | **`standby`**} | **`switchover`**}] **`cls`** [{**`acl`** | **`all`** | **`condition`** | **`config`** | **`ipc`** | **`mrfc`** | **`ppa`** | **`rpm`** | **`work`**}]

`no debug` [{**`boot`** {**`active`** | **`standby`**} | **`switchover`**}] **`cls`** [{**`acl`** | **`all`** | **`condition`** | **`config`** | **`ipc`** | **`mrfc`** | **`ppa`** | **`rpm`** | **`work`**}]

1.24.1 Purpose

Enables the generation of debug messages for IP classifier processes.

1.24.2 Command Mode

exec (10)

1.24.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.

switchover Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

acl Optional. Enables the generation of debug messages for configured access control lists (ACLs).

all Optional. Enables the generation of debug messages for all classifier processes.

condition Optional. Enables the generation of debug messages for configured ACL conditions.

config Optional. Enables the generation of debug messages for the classifier configuration.

ipc Optional. Enables the generation of debug messages for interprocess communication (IPC) events.

mrfc Optional. Enables the generation of debug messages for the modified recursive flow classification (MRFC) algorithm.

ppa Optional. Enables the generation of debug messages for classifier Packet Processing ASIC (PPA) events.

rpm Optional. Enables the generation of debug messages for classifier router policy manager (RPM) events.

work Optional. Enables the generation of debug messages for classifier work orders.

1.24.4 Default

The generation of debug messages is disabled.

1.24.5 Usage Guidelines

Use the **debug cls** command to enable the generation of debug messages for IP classifier processes. The IP classifier process is responsible for ACL computations and processing, so this command is useful for debugging when ACLs are configured and operating.

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 logging of classifier process debug messages.

1.24.6 Examples

The following example shows how to remove a configured access group, `tc1`, from the `two` interface, resulting in debug messages that are displayed by the **`debug cls`** command with the **`work`** keyword:

```
[local]Redback(config-ctx)#interface two

[local]Redback(config-if)#no ip access-group tc1

[local]Redback(config-if)#end

[local]Redback#debug cls work


Jul 18 15:51:10: %CLS-7-WORK: (WORK) cls_process_work_order@1458:
WORK_ORDER: ACL_UNBIND (type=1 prec=1) ctx=0x40080001, ifGrid=0x10000002, (OUT)

Jul 18 15:51:10: %CLS-7-WORK: (WORK) cls_int_unbind_to_ppa@700:
PPA_UNBIND( ifGrid=0x10000002/OUT, img_grid 400a2000 )

Jul 18 15:51:10: %CLS-7-WORK: (WORK) cls_CLSImageReleaseXX@718: cls_IFCClassifierDelete@349
released image=0x400a2000; refCount=1

Jul 18 15:51:10: %CLS-7-WORK: (WORK) cls_HWReleaseXX@531: cls_IFCClassifierDelete@353
released hw=0x400b2200, slot=12; refCount=1

Jul 18 15:51:10: %CLS-7-WORK: (WORK) cls_IFCGC@764: deleting ifc=0x10000002
```

1.25 debug community-list

`debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`community-list`**

`no debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`community-list`**

1.25.1 Purpose

Enables the generation of debug messages for the maintenance of community lists and for the comparison of Border Gateway Protocol (BGP) community attributes with community lists.

1.25.2 Command Mode

exec (10)

1.25.3 Syntax Description

`boot` Optional. Enables the generation of debug messages during a system reload.

`active` Enables the generation of debug messages for the active controller card.

`standby` Enables the generation of debug messages for the standby controller card.

`switchover` Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

1.25.4 Default

None

1.25.5 Usage Guidelines

Use the **`debug community-list`** command to enable the generation of debug messages for the maintenance of community lists and for the comparison of BGP community attributes with community lists.

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 community lists:

```
[local]Redback#debug community-list
```

1.26 debug context

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

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

1.26.1 Purpose

Enables the generation of debug messages for the current context.

1.26.2 Command Mode

exec (10)

1.26.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.

`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller card.
`general`	Enables the generation of general context debug messages.

1.26.4 Default

The generation of debug messages is disabled.

1.26.5 Usage Guidelines

Use the **`debug context`** command to enable the generation of debug messages for the current context.

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 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 enables the generation of context debug messages:

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

1.27 debug cspf

```
debug cspf {all | cfg | cli | cst | path | pcc | rcm |  spf | ted | tent}

no debug cspf
```

1.27.1 Purpose

Enables the generation of debug messages for Constrained Shortest Path First (CSPF) events.

1.27.2 Command Mode

exec

1.27.3 Syntax Description

`all` Enables the generation debug messages for all CSPF events.

`cfg` Enables the generation of CSPF configuration debug messages for configuration events.

`cli` Enables the generation of CSPF debug messages for CLI events.

`cst` Enables the generation of CSPF debug messages for constrained events.

`path` Enables the generation of CSPF debug messages for path events.

`pcc` Enables the generation of CSPF debug messages for path request information from RSVP events. `pcc` stands for Path Computation Client (PCC).

`rcm` Enables the generation of CSPF debug messages for Router Configuration Manager (RCM) events.

`spf` Enables the generation of CSPF debug messages for Shortest Path First (SPF) events.

`ted` Enables the generation CSPF debug messages for traffic engineering database (TED) events.

`tent` Enables the generation of CSPF debug messages for SPF computation events related to nodes that are tentatively on the final path. `tent` stands for tentative.

1.27.4 Default

The generation of debug messages for CSPF is disabled.

1.27.5 Usage Guidelines

Use the **`debug cspf`** command to enable the generation of debug messages for CSPF 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*.

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

1.27.6 Examples

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

```
[local]Redback#debug cspf ted
```

1.28 debug dhcp-relay

`debug` [{**`boot`** {**`active`** | **`standby`**} | **`switchover`**}] **`dhcp-relay`** {**`aaa`** | **`all`** | **`arp`** | **`clips`** | **`configuration`** | **`exception`** | **`file`** | **`general`** | **`helper`** | **`ipc`** | **`ism`** | **`packet`** | **`rcm`** | **`timer`**}

```
no debug [{boot {active | standby} | switchover}] dhcp-relay {aaa
| all | arp | clips | configuration | exception | file | general |
helper | ipc | ism | packet | rcm | timer}
```

1.28.1 Purpose

Enables the generation of debug messages for external Dynamic Host Configuration Protocol (DHCP) servers.

1.28.2 Command Mode

exec (10)

1.28.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
`aaa`	Enables the generation of debug messages for authentication, authorization, and accounting (AAA) events for external DHCP servers.
`all`	Enables the generation of debug messages for all external DHCP server events.
`arp`	Enables the generation of debug messages for external DHCP server Address Resolution Protocol (ARP) events.
`clips`	Enables the generation of debug messages for external DHCP server clientless IP service selection (CLIPS) events.
`configuration`	Enables the generation of debug messages for the external DHCP server configuration.
`exception`	Enables the generation of debug messages for external DHCP server exception events.
`file`	Enables the generation of debug messages for external DHCP server file events.
`general`	Enables the generation of debug messages for external DHCP server general events.
`helper`	Enables the generation of debug messages for external DHCP server helper events.

`ipc`	Enables the generation of debug messages for external DHCP server interprocess communication (IPC) events.
`ism`	Enables the generation of debug messages for external DHCP server Interface and Circuit State Manager (ISM) events.
`packet`	Enables the generation of debug messages for external DHCP server packet events.
`rcm`	Enables the generation of debug messages for external DHCP server Router Configuration Manager (RCM) events.
`timer`	Enables the generation of debug messages for external DHCP server timer events.

1.28.4 Default

The generation of debug messages for external DHCP server events is disabled.

1.28.5 Usage Guidelines

Use the **`debug dhcp-relay`** command to enable the generation of debug messages for external DHCP server events.

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.

1.28.6 Examples

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

```
[local]Redback#debug dhcp-relay packet
```

1.29 debug dhcp-server

debug [{**boot** {**active** | **standby**} | **switchover**}] **dhcp-server** {**aaa** | **all** | **arp** | **clips** | **configuration** | **exception** | **file** | **general** | **helper** | **ipc** | **ism** | **packet** | **rcm** | **timer**}

no debug [{**boot** {**active** | **standby**} | **switchover**}] **dhcp-server** {**aaa** | **all** | **arp** | **clips** | **configuration** | **exception** | **file** | **general** | **helper** | **ipc** | **ism** | **packet** | **rcm** | **timer**}

1.29.1 Purpose

Enables the generation of debug messages for internal Dynamic Host Configuration Protocol (DHCP) servers.

1.29.2 Command Mode

exec (10)

1.29.3 Syntax Description

`boot` — Optional. Enables the generation of debug messages during a system reload.

`active` — Enables the generation of debug messages for the active controller card.

`standby` — Enables the generation of debug messages for the standby controller card.

`switchover` — Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

`aaa` — Enables the generation of debug messages for authentication, authorization, and accounting (AAA) events for internal DHCP servers.

`all` — Enables the generation of debug messages for all internal DHCP server events.

`arp` — Enables the generation of debug messages for internal DHCP server Address Resolution Protocol (ARP) events.

`clips` Enables the generation of debug messages for internal DHCP server clientless IP service selection (CLIPS) events.

`configuration` Enables the generation of debug messages for the internal DHCP server configuration.

`exception` Enables the generation of debug messages for internal DHCP server exception events.

`file` Enables the generation of debug messages for internal DHCP server file events.

`general` Enables the generation of debug messages for internal DHCP server general events.

`helper` Enables the generation of debug messages for internal DHCP server helper events.

`ipc` Enables the generation of debug messages for internal DHCP server interprocess communication (IPC) events.

`ism` Enables the generation of debug messages for internal DHCP server Interface and Circuit State Manager (ISM) events.

`packet` Enables the generation of debug messages for internal DHCP server packet events.

`rcm` Enables the generation of debug messages for internal DHCP server Router Configuration Manager (RCM) events.

`timer` Enables the generation of debug messages for internal DHCP server timer events.

1.29.4 Default

The generation of debug messages for internal DHCP server events is disabled.

1.29.5 Usage Guidelines

Use the **`debug dhcp-server`** command to enable the generation of debug messages for internal DHCP server events.

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.

1.29.6 Examples

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

```
[local]Redback#debug dhcp-server packet
```

1.30 debug dhcpv6

debug dhcpv6 {aaa | all | be | cct | cli | diag | duid | err | exception | helper | info | ipv6addr | ipv6prefix | ism | packet | rcm | rib | timer | tree}

1.30.1 Command Mode

exec (10)

1.30.2 Syntax Description

aaa	Enables the generation of DHCPv6 AAA debugging messages.
all	Enables the generation of all DHCPv6 debugging messages.
be	Enables the generation of DHCPv6 backend debugging messages.
cct	Enables the generation of DHCPv6 circuit debugging messages.
cli	Enables the generation of CLI-related DHCPv6 debugging messages.
diag	Enables the generation of DHCPv6 diagnostic debugging messages.
duid	Enables the generation of DHCPv6 DUID debugging messages.
err	Enables the generation of DHCPv6 error level debugging messages.

`exception` Enables the generation of DHCPv6 exception debugging messages.

`helper` Enables the generation of DHCPv6 helper debugging messages.

`info` Enables the generation of DHCPv6 informational level debugging messages.

`ipv6addr` Enables the generation of DHCPv6 IPv6 address debugging messages.

`ipv6prefix` Enables the generation of DHCPv6 IPv6 prefix debugging messages.

`ism` Enables the generation of DHCPv6 ISM debugging messages.

`packet` Enables the generation of DHCPv6 packet debugging messages.

`rcm` Enables the generation of DHCPv6 RCM debugging messages.

`rib` Enables the generation of DHCPv6 RIB debugging messages.

`timer` Enables the generation of DHCPv6 timer debugging messages.

`tree` Enables the generation of DHCPv6 tree debugging messages.

1.30.3 Default

None

1.30.4 Usage Guidelines

Use the **`debug dhcpv6`** command to generate DHCPv6-related debugging messages. To generate them all, use the **`all`** keyword. To filter them by one part of the DHCPv6 functionality, use one of the other keywords.

1.30.5 Example

The following example shows how to generate debugging messages related to DHCPv6 exceptions:

```
[local]Redback#debug dhcpv6 exception
```

1.31 debug dot1q

```
debug [boot {active | standby} | switchover] dot1q [aaa | config |
internal | profile | pvc | rcm]
```

```
no debug [boot {active | standby} | switchover] dot1q [aaa | config |
internal | profile | pvc | rcm]
```

1.31.1 Purpose

Enables the generation of debug messages for various types of 802.1Q permanent virtual circuit (PVC)-related events.

1.31.2 Command Mode

exec (10)

1.31.3 Syntax Description

`boot` Optional. Enables the generation of debug messages during a system reload.

`active` Enables the generation of debug messages for the active controller card.

`standby` Enables the generation of debug messages for the standby controller card.

`switchover` Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

`aaa` Optional. Enables the generation of debug messages for Authentication, Authorization, and Accounting events.

`config` Optional. Enables the generation of debug messages for configuration events.

`internal` Optional. Enables the generation of debug messages for internal events.

`profile` Optional. Enables the generation of debug messages for profile events.

`pvc` Optional. Enables the generation of debug messages for permanent virtual circuit (PVC) events.

`rcm` Optional. Enables the generation of debug messages for Router Configuration Manager (RCM) configuration-related changes in the RCM process.

1.31.4 Default

The generation of debug messages for all 802.1Q-related events is disabled. The default, if the generation of debug messages is enabled without any optional keywords, is to generate debug messages for all types of 802.1Q-related events.

1.31.5 Usage Guidelines

Use the **`debug dot1q`** command to enable the generation of debug messages for various types of 802.1Q-related events.

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.

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 for 802.1Q events.

1.31.6 Examples

The following example shows how to enable the generation of debug messages for 802.1Q PVC events:

```
[local]Redback#debug dot1q pvc
```

1.32 debug ext-community-list

debug [**boot** {**active** | **standby**} | **switchover**] **ext-community-list**

no debug [**boot** {**active** | **standby**} | **switchover**] **ext-community-list**

1.32.1 Purpose

Enables the generation of debug messages for the maintenance of extended community lists and for the comparison of Border Gateway Protocol (BGP) extended community attributes with extended community lists.

1.32.2 Command Mode

exec (10)

1.32.3 Syntax Description

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

1.32.4 Default

None

1.32.5 Usage Guidelines

Use the **`debug ext-community-list`** command to enable the generation of debug messages for the maintenance of extended community lists and for the comparison of BGP extended community attributes with extended community lists.

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.

1.32.6 Examples

The following example shows how to enable the generation of debug messages for extended community lists:

```
[local]Redback#debug ext-community-list
```

1.33 debug frame-relay l2vpn

`debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`frame-relay l2vpn`**

`no debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`frame-relay l2vpn`**

1.33.1 Purpose

Enables the generation of debug messages for Layer 2 Virtual Private Network (L2VPN)-enabled Frame Relay permanent virtual circuits (PVCs).

1.33.2 Command Mode

exec (10)

1.33.3 Syntax Description

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

1.33.4 Default

The generation of debug messages is disabled.

1.33.5 Usage Guidelines

Use the **debug frame-relay l2vpn** command to enable the generation of debug messages for L2VPN-enabled Frame Relay PVCs.

Note: The SmartEdge 100 router does not support Frame Relay PVCs.

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.

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.

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 L2VPN-enabled Frame Relay debug messages.

1.33.6 Examples

The following example shows how to enable the generation of L2VPN-enabled Frame Relay debug messages:

```
[local]Redback#debug frame-relay l2vpn
```

1.34 debug frame-relay lmi

For ports on Packet over SONET/SDH (POS) traffic cards, the syntax is:

debug [**boot** {**active** | **standby**} | **switchover**] **frame-relay lmi** *slot*/*port* [**dlci** *dlci*]]

no debug [**boot** {**active** | **standby**} | **switchover**] **frame-relay lmi** [*slot*/*port* [**dlci** *dlci*]]

1.34.1 Purpose

Enables the generation of debug messages for all Frame Relay Local Management Interface (LMI) packet exchanges with a service provider.

1.34.2 Command Mode

exec (10)

1.34.3 Syntax Description

boot	Optional. Enables the generation of debug messages during a system reload.
active	Enables the generation of debug messages for the active controller card.
standby	Enables the generation of debug messages for the standby controller card.
switchover	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
slot	Chassis slot number of the traffic card for which the generation of debug messages is enabled.
port	Optional. Port number for which the generation of debug messages is enabled.
dlci *dlci*	Optional. Data-link connection identifier (DLCI) of the configured permanent virtual circuit (PVC) for which the generation of debug messages is requested. The range of values is 16 to 991.

1.34.4 Default

The generation of debug messages is disabled for all Frame Relay LMI messages. The default, if the generation of debug messages is enabled without the optional *slot*, *port*, *chan-num*, and *sub-chan-num* arguments, is to generate debug messages for LMI packet messages for all Frame Relay ports in all slots.

1.34.5 Usage Guidelines

Use the **debug frame-relay lmi** command to enable the generation of debug messages for all Frame Relay LMI packet messages with a service provider.

Note: The SmartEdge 100 router does not support Frame Relay PVCs.

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.

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.

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 LMI packet messages.

1.34.6 Examples

The following example shows how to enable the generation of debug messages for all Frame Relay LMI packet messages on channel `2` on port `1` of the channelized OC-12 traffic card in slot `3`:

```
[local]Redback#debug frame-relay lmi 3/1:2
```

1.35 debug frame-relay packet

For ports on Asynchronous Transfer Mode (ATM) OC, Ethernet, or Packet over SONET/SDH (POS) traffic cards, the syntax is:

debug [**boot** {**active** | **standby**} | **switchover**] **frame-relay packet** [*slot*/*port* [**dlci** *dlci*]]

no debug [**boot** {**active** | **standby**} | **switchover**] **frame-relay packet** [*slot*/*port* [**dlci** *dlci*]]

1.35.1 Purpose

Enables the generation of debug messages for Frame Relay Local Management Interface (LMI) packet messages, except those relating to LMI permanent virtual circuit (PVC) status.

1.35.2 Command Mode

exec (10)

1.35.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
slot	Optional. Chassis slot number of the traffic card for which the generation of debug messages is enabled. If not specified, enables the generation of debug messages for all Frame Relay ports in all slots.
port	Required if you enter the *slot* argument. Port number for which the generation of debug messages is enabled.
`dlci` *dlci*	Optional. Data-link connection identifier (DLCI) of the configured permanent virtual circuit (PVC) for which the generation of debug messages is requested. The range of values is 16 to 991.

1.35.4 Default

The generation of debug messages for Frame Relay LMI packet messages, except for those related to LMI PVC status, is disabled.

1.35.5 Usage Guidelines

Use the **`debug frame-relay packet`** command to enable the generation of debug messages for Frame Relay LMI packet messages, except those relating to LMI PVC status, with the service provider. To enable the generation of debug messages for all Frame Relay LMI messages, including PVC status, use the **`debug frame-relay lmi`** command.

Note: The SmartEdge 100 router does not support Frame Relay PVCs.

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.

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.

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 Frame Relay LMI packet messages.

1.35.6 Examples

The following example shows how to display typical debug messages when the generation of debug messages for Frame Relay LMI packet messages is enabled:

```
Aug 13 22:29:11: %FR-7-PKT: Port 3/1:4.31/6/2/30 (i) Packet length 14
Aug 13 22:29:11: %FR-7-PKT: Port 3/1:5.31/6/2/38 (i) Packet length 14
Aug 13 22:29:11: %FR-7-PKT: Port 3/1:6.31/6/2/46 (i) Packet length 14
Aug 13 22:29:13: %FR-7-PKT: Port 3/1:7.31/6/2/54 (o) Packet length 14
Aug 13 22:29:13: %FR-7-PKT: Port 3/1:8.31/6/2/62 (o) Packet length 14
Aug 13 22:29:13: %FR-7-PKT: Port 3/1:9.31/6/2/70 (o) Packet length 14
```

1.36 debug gre

debug [**boot** {**active** | **standby**} | **switchover**] **gre** [**config** | **peer** | **rcm** | **rib** | **tunnel**]

no debug [**boot** {**active** | **standby**} | **switchover**] **gre** [**config** | **peer** | **rcm** | **rib** | **tunnel**]

1.36.1 Purpose

Enables the generation of debug messages for Generic Routing Encapsulation (GRE)-related events.

1.36.2 Command Mode

exec (10)

1.36.3 Syntax Description

boot	Optional. Enables the generation of debug messages during a system reload.
active	Enables the generation of debug messages for the active controller card.
standby	Enables the generation of debug messages for the standby controller card.
switchover	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
config	Optional. Enables the generation of debug messages for GRE configuration events.
peer	Optional. Enables the generation of debug messages for GRE-related tunnel events.
rcm	Optional. Enables the generation of debug messages for GRE-related Router Configuration Manager (RCM) events.

rib Optional. Enables the generation of debug messages for GRE-related Routing Information Base (RIB) events.

tunnel Optional. Enables the generation of debug messages for GRE-related tunnel and tunnel circuit events.

1.36.4 Default

The generation of debug messages for GRE-related events is disabled.

1.36.5 Usage Guidelines

Use the **debug gre** command to enable the generation of debug messages for GRE-related events.

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.

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.

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 GRE-related events.

1.36.6 Examples

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

```
[local]Redback#debug gre config
```

1.37 debug gsmp

debug [{**boot** {**active** | **standby**} | **switchover**}] **gsmp packet** *event-type*

no debug [{**boot** {**active** | **standby**} | **switchover**}] **gsmp packet**

1.37.1 Purpose

Enables the generation of debug messages for General Switch Management Protocol (GSMP) packets.

1.37.2 Command Mode

exec (10)

1.37.3 Syntax Description

boot	Optional. Enables the generation of debug messages during a system reload.
active	Enables the generation of debug messages for the active controller card.
standby	Enables the generation of debug messages for the standby controller card.
switchover	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
gsmp	Enables the generation of debug messages for GSMP packets.
event-type	Type of events or data for which the generation of debug messages is enabled. See Table 1 for a list of event types and data.

1.37.4 Default

The generation of debug messages for GSMP packets is disabled.

1.37.5 Usage Guidelines

Use the **debug gsmp** command to enable the generation of debug messages for GSMP packets.

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.

Table 4 lists the keywords for the types of events or data for which you can enable the generation of debug messages.

Table 4 GSMP Event Types and Data

Event or Data Type	Description
aaa	Authentication, authorization, and accounting (AAA) interaction
adj	Adjacency protocol
all	Management class packet
cli	Command-line interface (CLI) interaction
event	Event class packet
general	General debugging
header	Packet header
io	Packet input and output
ism	Interface and Circuit State Manager (ISM) interaction
mgmt	Management class packet
neighbor	Neighbor event
packet	Received packet dump
stat	Packet statistics
timer	Timer event
tlv	Packet Tag-Length-Value (TLV)

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 GSMP packets.

1.37.6 Examples

The following example shows how to enable generation of debug messages for GSMP packets:

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

1.38 debug hr

debug [{**boot** {**active** | **standby**} | **switchover**}] **hr** {**all** | **error** | **event** [**detail**] | **policy** [**detail**] | **session** [**detail**] | **subscriber** [**detail**]}

no debug [{**boot** {**active** | **standby**} | **switchover**}] **hr** {**all** | **error** | **event** [**detail**] | **policy** [**detail**] | **session** [**detail**] | **subscriber** [**detail**]}

1.38.1 Purpose

Enables the generation of debug messages for HTTP redirect events and error messages.

1.38.2 Command Mode

exec (10)

1.38.3 Syntax Description

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

`all`	Enables the generation of debug messages for all HTTP redirect events.
`error`	Enables the generation of debug messages for minor error messages.
`event`	Enables the generation of debug messages for general events.
`detail`	Optional. Enables the generation of detailed information for the specified type of events.
`policy`	Enables the generation of debug messages for policy-related events.
`session`	Enables the generation of debug messages for session-related events.
`subscriber`	Enables the generation of debug messages for subscriber-related events.

1.38.4 Default

The generation of debug messages for HTTP redirect events and error messages is disabled.

1.38.5 Usage Guidelines

Use the **`debug hr`** command to enable the generation of debug messages for HTTP redirect events and error messages.

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 debugging 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.

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 for HTTP redirect events and error messages.

1.38.6 Examples

The following example shows how to enable the generation of debug messages for session-related events for the HTTP redirect feature:

```
[local]Redback#debug hr session
```

1.39 debug if

debug [{**boot** {**active** | **standby**}} | {**switchover**}] **if** {**all** | **error** | **rcm**}

no debug [{**boot** {**active** | **standby**}} | {**switchover**}] **if** {**all** | **error** | **rcm**}

1.39.1 Purpose

Enables the generation of debug messages for all configured interfaces in the current context.

1.39.2 Command Mode

exec (10)

1.39.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

standby Enables the generation of debug messages for the standby controller card.

switchover Optional. Enables the generation of debug messages during a switchover from the active to the standby controller card.

all Enables the generation of all debug messages.

`error`	Enables the generation of only error debug messages.
`rcm`	Enables the generation of only Router Configuration Manager (RCM) debug messages.

1.39.4 Default

The generation of debug messages is disabled.

1.39.5 Usage Guidelines

Use the **`debug if`** command to enable the generation of debug messages for all configured interfaces in the current context.

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 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*.

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

1.39.6 Examples

The following example shows how to enable the generation of only RCM debug messages for all configured interfaces:

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

1.40 debug igmp

debug [**boot** {**active** | **standby**} | **switchover**] **igmp** [*group-addr*]

no debug [**boot** {**active** | **standby**} | **switchover**] **igmp** [*group-addr*]

1.40.1 Purpose

Enables the generation Internet Group Management Protocol (IGMP) debug messages.

1.40.2 Command Mode

exec (10)

1.40.3 Syntax Description

boot	Optional. Enables the generation of debug messages during a system reload.
active	Enables the generation of debug messages for the active controller card.
standby	Enables the generation of debug messages for the standby controller card.
switchover	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
group-addr	Optional. IP address of the IGMP group.

1.40.4 Default

The generation of debug messages is disabled.

1.40.5 Usage Guidelines

Use the **debug igmp** command to enable the generation of all IGMP debug messages.

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

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 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 IGMP debug messages.

1.40.6 Examples

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

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

1.41 debug ip access-list

debug [{**boot** {**active** | **standby**} | **switchover**}] **ip access-list**

no debug [{**boot** {**active** | **standby**} | **switchover**}] **ip access-list**

1.41.1 Purpose

Enables the generation of debug messages for IP access control list (ACL) events.

1.41.2 Command Mode

exec (10)

1.41.3 Syntax Description

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

1.41.4 Default

The generation of debug messages is disabled.

1.41.5 Usage Guidelines

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

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 for IP ACLs.

1.41.6 Examples

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

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

1.42 debug ip dns

debug [{**boot** {**active** | **standby**} | **switchover**}] **ip dns**

no debug [{**boot** {**active** | **standby**} | **switchover**}] **ip dns**

1.42.1 Purpose

Enables the generation of debug messages for the Domain Name System (DNS).

1.42.2 Command Mode

exec (10)

1.42.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.

standby Enables the generation of debug messages for the standby controller card.

switchover Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

1.42.4 Default

The generation of debug messages for DNS is disabled.

1.42.5 Usage Guidelines

Use the **debug ip dns** command to enable the generation of debug messages for DNS.

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 when enabling the generation of any debug messages on a production system.

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 debug messages.

1.42.6 Examples

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

```
[local]Redback#debug ip dns
```

1.43 debug ip mrouting

debug [**boot** {**active** | **standby**} | **switchover**] **ip mrouting** [*group-addr*]

no debug [**boot** {**active** | **standby**} | **switchover**] **ip mrouting** [*group-addr*]

1.43.1 Purpose

Enables the generation of debug messages for multicast routing table entries.

1.43.2 Command Mode

exec (10)

1.43.3 Syntax Description

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

1.43.4 Default

The generation of debug messages is disabled.

1.43.5 Usage Guidelines

Use the **debug ip mrouting** command to enable the generation of debug messages for all multicast routing table entries.

Use the *group-addr* argument to enable the generation of debug messages for multicast routing table entries for a specific IGMP group.

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 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 multicast routing table entries.

1.43.6 Examples

The following example shows how to enable the generation of debug messages for multicast routing table entry activity for the multicast group, `224.1.1.1:`

```
[local]Redback#debug ip mrouting 224.1.1.1
```

1.44 debug ip prefix-list

debug [**boot** {**active** | **standby**} | **switchover**] **ip prefix-list**

no debug [**boot** {**active** | **standby**} | **switchover**] **ip prefix-list**

1.44.1 Purpose

Enables the generation of debug messages for the maintenance of IP prefix lists and for the comparison of IP prefix entries to IP prefix lists.

1.44.2 Command Mode

exec (10)

1.44.3 Syntax Description

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

1.44.4 Default

None

1.44.5 Usage Guidelines

Use the **`debug ip prefix-list`** command to enable the generation of debug messages for the maintenance of IP prefix lists and for the comparison of IP prefixes to IP prefix lists.

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 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.

1.44.6 Examples

The following example shows how to enable the generation of debug messages for IP prefix lists:

```
[local]Redback#debug ip prefix-list
```

1.45 debug ip routing

debug [**boot** {**active** | **standby**} | **switchover**] **ip routing** {**all** | **fib-addition** | **fib-deletion** | **redist-addition** | **redist-deletion** | **route-addition** | **route-deletion**}[*ip-addr*/*prefix-length*]

no debug [**boot** {**active** | **standby**} | **switchover**] **ip routing** {**all** | **fib-addition** | **fib-deletion** | **redist-addition** | **redist-deletion** | **route-addition** | **route-deletion**}[*ip-addr*/*prefix-length*]

1.45.1 Purpose

Enables the generation of IP routing debug messages.

1.45.2 Command Mode

exec

1.45.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

standby Enables the generation of debug messages for the standby controller card.

switchover Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

all Enables the generation of messages only for events related to the Routing Information Base (RIB).

fib-addition Enables the generation of messages only for events related to route additions to the Forwarding Information Base (FIB).

fib-deletion Enables the generation of messages only for events relating to routes removed from the FIB.

redist-addition Enables the generation of messages only for events relating to the addition of redistributed routes to the client protocol.

redist-deletion Enables the generation of messages only for events relating to the removal of redistributed routes from the client protocol.

route-addition Enables the generation of messages only for events relating to the addition of routes to the routing table.

route-deletion Enables the generation of messages only for events relating to the removal of routes from the routing table.

ip-addr/prefix-length Optional. IP address (in the form ***A.B.C.D)*** and prefix length, separated by the slash (/) character. The range of values for the ***prefix-length*** argument is 0 to 32. Only messages for events for routes that match the IP address range are generated.

1.45.4 Default

The generation of debug messages is disabled.

1.45.5 Usage Guidelines

Use the **debug ip routing** command to enable the generation of IP routing debug messages.

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 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.

1.45.6 Examples

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

```
[local]Redback#debug ip routing route-addition
```

The following example shows how to enable the generation of debug messages for the addition of routes in the IP address range `1.0.0.0/8`:

```
[local]Redback#debug ip routing route-addition 1.0.0.0/8
```

1.46 debug ipmul

`debug ipmul {mcastmgr | misc | ppa}`

`no debug ipmul {mcastmgr | misc | ppa}`

1.46.1 Purpose

Enables the generation of debug messages by the IGMP snooper when it is communicating with multicast manager.

1.46.2 Command Mode

exec

1.46.3 Syntax Description

`mcastmgr`	Enables the generation of debug messages for IGMP multicast manager processes.
`misc`	Enables the generation of debug messages for miscellaneous IGMP functions.
`ppa`	Enables the generation of debug messages for IGMP PPA processes.

1.46.4 Default

The generation of debug messages is disabled.

1.46.5 Usage Guidelines

Use the **`debug ipmul`** command to enable the generation of debug messages by the IGMP snooper when it is communicating with multicast manager.

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 IGMP multicast manager debug messages:

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

1.47 debug iprwlock

debug [{**boot** {**active** | **standby**} | **switchover**}] **iprwlock**

no debug [{**boot** {**active** | **standby**} | **switchover**}] **iprwlock**

1.47.1 Purpose

Enables the generation of debug messages for IP read-write lock events.

1.47.2 Command Mode

exec (10)

1.47.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

standby Enables the generation of debug messages for the standby controller card.

switchover Optional. Enables the generation of debug messages during a switchover from the active to the standby controller card.

1.47.4 Default

The generation of debug messages for IP read-write lock events is disabled.

1.47.5 Usage Guidelines

Use the **debug iprwlock** command to enable the generation of debug messages for IP read-write lock 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 debug 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.

1.47.6 Examples

The following example shows how to enable the generation of debug messages for IP read-write lock events:

```
[local]Redback#debug iprwlock
```

1.48 debug ipv6 access-list

debug ipv6 access-list

no debug ipv6 access-list

1.48.1 Purpose

Enables the generation of debug messages for IPv6 ACL events.

1.48.2 Command Mode

exec (10)

1.48.3 Syntax Description

This command has no keywords or arguments.

1.48.4 Default

No default messages are generated.

1.48.5 Usage Guidelines

Use the **`debug ipv6 access-list`** command to enable the generation of debug messages for IPv6 ACL events.

Use the **`no`** version of this command to disable the generation of those debug messages.

1.48.6 Examples

The following example shows how to enable the generation of debug messages for IPv6 ACL events:

```
[local]Redback#debug ipv6 access-list
```

1.49 debug ipv6 policy access-list

`debug ipv6 policy access-list`

`no debug ipv6 policy access-list`

1.49.1 Purpose

Enables the generation of debug messages for IPv6 policy ACL events.

1.49.2 Command Mode

exec (10)

1.49.3 Syntax Description

This command has no keywords or arguments.

1.49.4 Default

No default messages are generated.

1.49.5 Usage Guidelines

Use the **debug ipv6 policy access-list** command to enable the generation of debug messages for IPv6 policy ACL events.

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

1.49.6 Examples

The following example shows how to enable the generation of debug messages for IPv6 policy ACL events:

```
[local]Redback#debug ipv6 policy access-list
```

1.50 debug ipv6 prefix-library

debug ipv6 prefix-library {**all** | **error** | **general**}

1.50.1 Purpose

Enables the generation of debug messages for the IPv6 prefix library.

1.50.2 Command Mode

exec (10)

1.50.3 Syntax Description

all	Enables the generation of debug messages for all IPv6 prefix library events.
error	Enables the generation of debug messages for minor errors related to the IPv6 prefix library.
general	Enables the generation of debug messages for general IPv6 prefix library events.

1.50.4 Default

The generation of debug messages is disabled.

1.50.5 Usage Guidelines

Use the **debug ipv6 prefix-library** command to enable the generation of debug messages for the IPv6 prefix library.

Use the **no** version of this command to disable the generation of debug messages related to the IPv6 prefix library.

1.50.6 Examples

The following example shows how to enable the generation of debug messages for all IPv6 prefix library events:

```
d
[local]Redback>debug ipv6 prefix-library
```

1.51 debug ipv6 prefix-list

debug [**boot** {**active** | **standby**} | **switchover**] **ipv6 prefix-list**

no debug [**boot** {**active** | **standby**} | **switchover**] **ipv6 prefix-list**

1.51.1 Purpose

Enables the generation of debug messages for the maintenance of IP Version 6 (IPv6) prefix lists and for the comparison of IPv6 prefix entries to IPv6 prefix lists.

1.51.2 Command Mode

exec (10)

1.51.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

1.51.4 Default

None

1.51.5 Usage Guidelines

Use the **`debug ipv6 prefix-list`** command to enable the generation of debug messages for the maintenance of IPv6 prefix lists and for the comparison of IPv6 prefixes to IPv6 prefix lists.

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 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.

1.51.6 Examples

The following example shows how to enable the generation of debug messages for IPv6 prefix lists:

```
[local]Redback#debug ipv6 prefix-list
```

1.52 debug ipv6 routing

debug [**boot** {**active** | **standby**} | **switchover**] **ipv6 routing** {**all** | **fib-addition** | **fib-deletion** | **redist-addition** | **redist-deletion** | **route-addition** | **route-deletion**} [***ipv6-addr/prefix-length***]

no debug [**boot** {**active** | **standby**} | **switchover**] **ipv6 routing** {**all** | **fib-addition** | **fib-deletion** | **redist-addition** | **redist-deletion** | **route-addition** | **route-deletion**} [***ipv6-addr/prefix-length***]

1.52.1 Purpose

Enables the generation of IP Version 6 (IPv6) routing debug messages.

1.52.2 Command Mode

exec

1.52.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
`all`	Enables the generation of messages only for events related to the Routing Information Base (RIB).
`fib-addition`	Enables the generation of messages only for events related to route additions to the Forwarding Information Base (FIB).
`fib-deletion`	Enables the generation of messages only for events relating to routes removed from the FIB.

`redist-addition`	Enables the generation of messages only for events relating to the addition of redistributed routes to the client protocol.
`redist-deletion`	Enables the generation of messages only for events relating to the removal of redistributed routes from the client protocol.
`route-addition`	Enables the generation of messages only for events relating to the addition of routes to the routing table.
`route-deletion`	Enables the generation of messages only for events relating to the removal of routes from the routing table.
`ipv6-addr/prefix-length`	Optional. IP address in the form ***`A:B:C:D:E:F:G:H`*** and prefix length, separated by the slash (`/`) character. The range of values for the ***`prefix-length`*** argument is 0 to 128. Only messages for events for routes that match the IPv6 address range are generated.

1.52.4 Default

The generation of debug messages is disabled.

1.52.5 Usage Guidelines

Use the **`debug ipv6 routing`** command to enable the generation of IPv6 routing debug messages.

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 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.

1.52.6 Examples

The following example shows how to enable the generation of IPv6 route addition debug messages:

```
[local]Redback#debug ipv6 routing route-addition
```

The following example shows how to enable the generation of debug messages for the addition of routes in the IPv6 address range, 2F1E:5400::1000/64:

```
[local]Redback#debug ipv6 routing route-addition 2F1E:5400::1000/64
```

1.53 debug isis adjacency

debug [**boot** {**active** | **standby**} | **switchover**] **isis adjacency** [**detail**] [**interface** *if-name*] [**prefix-list** *pl-name*] [**system-id** *sys-id*]

no debug [**boot** {**active** | **standby**} | **switchover**] **isis adjacency**

1.53.1 Purpose

Enables the generation of Intermediate System-to-Intermediate System (IS-IS) adjacency debug messages.

1.53.2 Command Mode

exec (10)

1.53.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

standby Enables the generation of debug messages for the standby controller card.

switchover	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
detail	Optional. Provides detailed information with debug messages.
interface *if-name*	Optional. Interface name. You can choose to enable debug messages for events related to the specified interface.
prefix-list *pl-name*	Optional. Prefix list name. You can choose to enable debug messages for events related to the specified prefix list.
system-id *sys-id*	Optional. System ID of the specific system for which you want to enable debug messages. The system ID is the six bytes before the last byte of the network entity title (NET) and is expressed in the *xxxx.xxxx.xxxx* format.

1.53.4 Default

The generation of debug messages is disabled.

1.53.5 Usage Guidelines

Use the **debug isis adjacency** command to enable the generation of IS-IS adjacency debug messages. This command provides information on changing states of adjacencies, and helps identify potential configuration problems, such as duplicate system IDs, nonmatching IP addresses, and so on, that prevent adjacencies from being formed.

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 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 IS-IS adjacency debug messages.

1.53.6 Examples

The following example shows how to enable the generation of IS-IS adjacency debug messages:

```
[local]Redback#debug isis adjacency detail
```

The following example displays typical debug messages generated by the system when the generation of IS-IS adjacency debug messages is enabled:

```
Oct 31 19:47:40: %ISIS-7-ADJ: send L1 LAN IIH on intf redback

Oct 31 19:47:41: %ISIS-7-ADJ: send L2 LAN IIH on intf redback

Oct 31 19:47:44: %ISIS-7-ADJ: rcvd L2 LAN IIH from
0010.7bcc.4b7e on intf redback

Oct 31 19:47:48: %ISIS-7-ADJ: send L1 LAN IIH on intf redback
```

1.54 debug isis all

`debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`isis all`**

`no debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`isis all`**

1.54.1 Purpose

Enables the generation of all intermediate system-to-intermediate-system (IS-IS) debug messages.

1.54.2 Command Mode

exec (10)

1.54.3 Syntax Description

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

1.54.4 Default

The generation of debug messages is disabled.

1.54.5 Usage Guidelines

Use the **`debug isis all`** command to enable the generation of all IS-IS debug messages.

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 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 IS-IS debug messages.

1.54.6 Examples

The following example enables the generation of all IS-IS debug messages:

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

1.55 debug isis bfd

debug [**boot** {**active** | **standby**} | **switchover**] **isis bfd**

no debug [**boot** {**active** | **standby**} | **switchover**] **isis bfd**

1.55.1 Purpose

Enables the generation of Intermediate System-to-Intermediate System (IS-IS) Bidirectional Forwarding Detection (BFD) event debug messages.

1.55.2 Command Mode

exec (10)

1.55.3 Syntax Description

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

1.55.4 Default

The generation of debug messages is disabled.

1.55.5 Usage Guidelines

Use the **debug isis bfd** command to enable the generation of IS-IS BFD event debug messages.

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 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 IS-IS BFD event debug messages.

1.55.6 Examples

The following example shows how to enable generation of IS-IS BFD event debug messages:

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

1.56 debug isis circuit

debug [**boot** {**active** | **standby**} | **switchover**] **isis circuit** [**detail**] [**interface** *if-name*] [**prefix-list** *pl-name*] [**system-id** *sys-id*]

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

1.56.1 Purpose

Enables the generation of Intermediate System-to-Intermediate System (IS-IS) circuit debug messages.

1.56.2 Command Mode

exec (10)

1.56.3 Syntax Description

`boot` Optional. Enables the generation of debug messages during a system reload.

`active` Enables the generation of debug messages for the active controller card.

`standby` Enables the generation of debug messages for the standby controller card.

`switchover` Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

`detail` Optional. Provides detailed information with debug messages.

`interface` *`if-name`* Optional. Interface name. You can choose to enable debug messages for events related to the specified interface.

`prefix-list` *`pl-name`* Optional. Prefix list name. You can choose to enable debug messages for events related to the specified prefix list.

`system-id` *`sys-id`* Optional. System ID of the specific system for which you want to enable debug messages. The system ID is the six bytes before the last byte of the network entity title (NET) and is expressed in the *`xxxx.xxxx.xxxx`* format.

1.56.4 Default

The generation of debug messages is disabled.

1.56.5 Usage Guidelines

Use the **`debug isis circuit`** command to enable the generation of IS-IS circuit debug messages.

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 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 IS-IS circuit debug messages.

1.56.6 Examples

The following example shows how to enable generation of detailed IS-IS circuit debug messages:

```
[local]Redback#debug isis circuit detail
```

The following example displays typical debug messages generated by the system when the generation of detailed IS-IS circuit debug messages is enabled:

```
Oct 31 19:52:02: %ISIS-7-CKT: isis circuit ericsson(1) down

Oct 31 19:56:40: %ISIS-7-CKT: isis circuit ericsson(1) up
```

1.57 debug isis graceful-restart

debug isis graceful-restart

no debug isis graceful-restart

1.57.1 Purpose

Enables the generation of Intermediate System-to-Intermediate System (IS-IS) graceful restart event messages.

1.57.2 Command Mode

exec (10)

1.57.3 Syntax Description

This command has no keywords or arguments.

1.57.4 Default

The generation of IS-IS graceful restart event messages is disabled.

1.57.5 Usage Guidelines

Use the **debug isis graceful-restart** command to enable the generation of IS-IS graceful restart event messages.

Use the **no** form of this command to disable the generation of IS-IS graceful restart event messages.

1.57.6 Examples

The following example shows how to enable the generation of IS-IS graceful restart event messages:

```
[local]Redback #debug isis graceful-restart
```

1.58 debug isis hello-packets

debug [**boot** {**active** | **standby**} | **switchover**] **isis hello-packets** [**send** | **receive**] [**detail**] [**interface** ***if-name***] [**prefix-list** ***pl-name***] [**system-id** ***sys-id***]

```
no debug [boot {active | standby} | switchover] isis
hello-packets
```

1.58.1 Purpose

Enables the generation of Intermediate System-to-Intermediate System (IS-IS) Hello packet debug messages.

1.58.2 Command Mode

exec (10)

1.58.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
`send`	Optional. Enables the generation of debug messages for sent packets.
`receive`	Optional. Enables the generation of debug messages for received packets.
`detail`	Optional. Provides detailed information with debug messages.
`interface` *`if-name`*	Optional. Interface name. You can choose to enable debug messages for events related to the specified interface.
`prefix-list` *`pl-name`*	Optional. Prefix list name. You can choose to enable debug messages for events related to the specified prefix list.
`system-id` *`sys-id`*	Optional. System ID of the specific system for which you want to enable debug messages. The system ID is the six bytes before the last byte of the network entity title (NET) and is expressed in the *`xxxx.xxxx.xxxx`* format.

1.58.4 Default

The generation of debug messages is disabled.

1.58.5 Usage Guidelines

Use the **`debug isis hello-packets`** command to enable the generation of IS-IS Hello packet debug messages.

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 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 IS-IS Hello packet debug messages.

1.58.6 Examples

The following example shows how to enable generation of IS-IS Hello packet debug messages:

```
[local]Redback#debug isis hello-packets
```

The following example displays typical debug messages generated by the system when the generation of IS-IS Hello packet debug messages is enabled:

```
Oct 31 19:44:28: %ISIS-7-ADJPKT_SD: send L2 IIH Pkt Len: 1480,
HT:   24 IS: 0008.0008.0008

Oct 31 19:46:29: %ISIS-7-ADJPKT_RV: recv L2 IIH Pkt Len: 1497,
HT:   30 IS: 1111.2222.3333

Oct 31 19:44:29: %ISIS-7-ADJPKT_SD: send L1 IIH Pkt Len: 1480,
HT:   24 IS: 0008.0008.0008
```

1.59 debug isis local-updates

debug [**boot** {**active** | **standby**} | **switchover**] **isis local-updates** [**detail**] [**interface** ***if-name***] [**prefix-list** ***pl-name***] [**system-id** ***sys-id***]

no debug [**boot** {**active** | **standby**} | **switchover**] **isis local-updates**

1.59.1 Purpose

Enables the generation of Intermediate System-to-Intermediate System (IS-IS) locally generated link-state protocol data unit (LSP) event debug messages.

1.59.2 Command Mode

exec (10)

1.59.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

standby Enables the generation of debug messages for the standby controller card.

switchover Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

detail Optional. Provides detailed information with debug messages.

interface ***if-name*** Optional. Interface name. You can choose to enable debug messages for events related to the specified interface.

`prefix-list` *`pl-name`*	Optional. Prefix list name. You can choose to enable debug messages for events related to the specified prefix list.
`system-id` *`sys-id`*	Optional. System ID of the specific system for which you want to enable debug messages. The system ID is the six bytes before the last byte of the network entity title (NET) and is expressed in the *`xxxx.xxxx.xxxx`* format.

1.59.4 Default

The generation of debug messages is disabled.

1.59.5 Usage Guidelines

Use the **`debug isis local-updates`** command to enable the generation of IS-IS locally generated LSP event debug messages.

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 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 IS-IS locally generated LSP event debug messages.

1.59.6 Examples

The following example shows how to enable the generation of IS-IS locally generated LSP event debug messages:

```
[local]Redback#debug isis local-updates
```

The following example displays typical debug messages generated by the system when the generation of IS-IS locally generated LSP event debug messages is enabled:

```
Jun 28 23:57:56: %ISIS-7-LOC_UPDT: Building L1 LSP
for pid 0 frag num 0 len 48

Jun 28 23:57:56: %ISIS-7-LOC_UPDT: Building L2 LSP
for pid 0 frag num 0 len 75

Jun 28 23:57:56: %ISIS-7-LOC_UPDT: Building L2
Pnode LSP for pid 1 frag num 0 len 55
```

1.60 debug isis lsp-packets

debug [**boot** {**active** | **standby**} | **switchover**] **isis lsp-packets** [**send** | **receive**] [**detail**] [**interface** ***if-name***] [**prefix-list** ***pl-name***] [**system-id** ***sys-id***] [**send** | **receive**] [**detail**]

no debug [**boot** {**active** | **standby**} | **switchover**] **isis lsp-packets**

1.60.1 Purpose

Enables the generation of Intermediate System-to-Intermediate System (IS-IS) link-state protocol data unit (LSP) debug messages.

1.60.2 Command Mode

exec (10)

1.60.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

standby Enables the generation of debug messages for the standby controller card.

switchover	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
send	Optional. Enables the generation of debug messages for sent packets.
receive	Optional. Enables the generation of debug messages for received packets.
detail	Optional. Provides detailed information with debug messages.
interface ***if-name***	Optional. Interface name. You can choose to enable debug messages for events related to the specified interface.
prefix-list ***pl-name***	Optional. Prefix list name. You can choose to enable debug messages for events related to the specified prefix list.
system-id ***sys-id***	Optional. System ID of the specific system for which you want to enable debug messages. The system ID is the six bytes before the last byte of the network entity title (NET) and is expressed in the ***xxxx.xxxx.xxxx*** format.

1.60.4 Default

The generation of debug messages is disabled.

1.60.5 Usage Guidelines

Use the **debug isis lsp-packets** command to enable the generation of IS-IS LSP debug messages.

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 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 IS-IS LSP debug messages.

1.60.6 Examples

The following example shows how to enable the generation of IS-IS LSP debug messages:

```
[local]Redback#debug isis lsp-packets
```

The following example displays typical debug messages generated by the system when the generation of IS-IS LSP debug messages is enabled:

```
Oct 31 20:26:38: %ISIS-7-LSPPKT_RV: rcvd L2 LSP on intf redback

Oct 31 20:26:38: %ISIS-7-LSPPKT_RV: recv L2 LSP Pkt Len:  118,
LSPID 1111.2222.3333.0000, LT: 1190, Seq 0x477

Oct 31 20:26:39: %ISIS-7-LSPPKT_SD: Sent L2 LSP 0008.0008.0008.0000
on intf redback
```

1.61 debug isis policy

debug [**boot** {**active** | **standby**} | **switchover**] **isis policy** [**interface** ***if-name***] [**prefix-list** ***pl-name***] [**system-id** ***sys-id***]

no debug [**boot** {**active** | **standby**} | **switchover**] **isis policy**

1.61.1 Purpose

Enables the generation of Intermediate System-to-Intermediate System (IS-IS) routing policy debug messages.

1.61.2 Command Mode

exec (10)

1.61.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
`interface` ***`if-name`***	Optional. Interface name. You can choose to enable debug messages for events related to the specified interface.
`prefix-list` ***`pl-name`***	Optional. Prefix list name. You can choose to enable debug messages for events related to the specified prefix list.
`system-id` ***`sys-id`***	Optional. System ID of the specific system for which you want to enable debug messages. The system ID is the six bytes before the last byte of the network entity title (NET) and is expressed in the ***`xxxx.xxxx.xxxx`*** format.

1.61.4 Default

The generation of debug messages is disabled.

1.61.5 Usage Guidelines

Use the **`debug isis policy`** command to enable the generation of IS-IS routing policy debug messages.

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 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 IS-IS routing policy debug messages.

1.61.6 Examples

The following example shows how to enable the generation of IS-IS routing policy debug messages:

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

The following example displays typical debug messages generated by the system when the generation of IS-IS routing policy debug messages is enabled:

```
Feb 20 00:53:53: %ISIS-7-POLICY: policy change
notification name isis1

Feb 20 00:54:24: %ISIS-7-POLICY: prefix-list check pass
192.168.1.0/24 dest level-2
```

1.62 debug isis protocol-errors

debug [**boot** {**active** | **standby**} | **switchover**] **isis protocol-errors** [**interface** ***if-name***] [**prefix-list** ***pl-name***] [**system-id** ***sys-id***]

no debug [**boot** {**active** | **standby**} | **switchover**] **isis protocol-errors**

1.62.1 Purpose

Enables the generation of debug messages for incorrectly formatted or corrupted intermediate system-to-intermediate system (IS-IS) packets.

1.62.2 Command Mode

exec (10)

1.62.3 Syntax Description

`boot`
Optional. Enables the generation of debug messages during a system reload.

`active`
Enables the generation of debug messages for the active controller card.

`standby`
Enables the generation of debug messages for the standby controller card.

`switchover`
Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

`interface` *`if-name`*
Optional. Interface name. You can choose to enable debug messages for events related to the specified interface.

`prefix-list` *`pl-name`*
Optional. Prefix list name. You can choose to enable debug messages for events related to the specified prefix list.

`system-id` *`sys-id`*
Optional. System ID of the specific system for which you want to enable debug messages. The system ID is the six bytes before the last byte of the network entity title (NET) and is expressed in the *`xxxx.xxxx.xxxx`* format.

1.62.4 Default

The generation of debug messages is disabled.

1.62.5 Usage Guidelines

Use the **`debug isis protocol-errors`** command to enable the generation of debug messages for incorrectly formatted or corrupted IS-IS packets.

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 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 incorrectly formatted or corrupted IS-IS packets.

1.62.6 Examples

The following example shows how to enable the generation of debug messages for incorrectly formatted or corrupted IS-IS packets:

```
[local]Redback#debug isis protocol-errors
```

1.63 debug isis redistribution

debug isis redistribution [**detail**]

no debug isis redistribution [**detail**]

1.63.1 Purpose

Enables the generation of debug messages for events relating to Intermediate System-to-Intermediate System (IS-IS) route redistribution.

1.63.2 Command Mode

exec (10)

1.63.3 Syntax Description

detail	Optional. Provides detailed information with debug messages.

1.63.4 Default

The generation of debug messages is disabled.

1.63.5 Usage Guidelines

Use the **debug isis redistribution** command to enable the generation of debug messages for events relating to IS-IS route redistribution.

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 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 events relating to IS-IS route redistribution.

1.63.6 Examples

The following example shows how to enable the generation of debug messages for events relating to IS-IS route redistribution:

```
[local]Redback#debug isis redistribution detail
```

1.64 debug isis routes

debug [**boot** {**active** | **standby**} | **switchover**] **isis routes** [**detail**] [**interface** *if-name*] [**prefix-list** *pl-name*] [**system-id** *sys-id*]

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

1.64.1 Purpose

Enables the generation of debug messages for events relating to interaction between the Router Information Base (RIB) and Intermediate System-to-Intermediate System (IS-IS).

1.64.2 Command Mode

exec (10)

1.64.3 Syntax Description

`boot` Optional. Enables the generation of debug messages during a system reload.

`active` Enables the generation of debug messages for the active controller card.

`standby` Enables the generation of debug messages for the standby controller card.

`switchover` Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

`detail` Optional. Provides detailed information with debug messages.

`interface` *`if-name`* Optional. Interface name. You can choose to enable debug messages for events related to the specified interface.

`prefix-list` *`pl-name`* Optional. Prefix list name. You can choose to enable debug messages for events related to the specified prefix list.

`system-id` *`sys-id`* Optional. System ID of the specific system for which you want to enable debug messages. The system ID is the six bytes before the last byte of the network entity title (NET) and is expressed in the *`xxxx.xxxx.xxxx`* format.

1.64.4 Default

The generation of debug messages is disabled.

1.64.5 Usage Guidelines

Use the **`debug isis routes`** command to enable the generation of debug messages for events relating to interaction between the RIB and IS-IS.

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 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 events relating to interaction between the RIB and IS-IS.

1.64.6 Examples

The following example shows how to enable the generation of debug messages for events relating to interaction between the RIB and IS-IS:

```
[local]Redback#debug isis routes detail
```

The following example displays typical debug messages generated by the system when the generation of debug messages for events relating to interaction between the RIB and IS-IS is enabled:

```
ISIS: starting to add L1 routes to local RIB
ISIS: adding L1 route 15.15.1.0/255.255.255.0 version 0 to local RIB
ISIS: bumping route 15.15.1.0/255.255.255.0 from version 0 to version 2
ISIS: ending addition of L1 routes to local RIB
ISIS: starting to download routes into global RIB, levels run 1
ISIS: proceeding with download by walking old version local RIB routes
ISIS: proceeding with download by walking new version local RIB routes
ISIS: ending download of routes into global RIB, old version bumped to 1
```

1.65 debug isis snp-packets

debug [**boot** {**active** | **standby**} | **switchover**] **isis snp-packets** [**csnp** | **psnp**] [**send** | **receive**] [**detail**] [**interface** ***if-name***] [**prefix-list** ***pl-name***] [**system-id** ***sys-id***]

no debug [**boot** {**active** | **standby**} | **switchover**] **isis snp-packets**

1.65.1 Purpose

Enables the generation of debug messages related to intermediate system-to-intermediate-system (IS-IS) complete sequence number protocol data units (CSNPs) and partial sequence number protocol data units (PSNPs).

1.65.2 Command Mode

exec (10)

1.65.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

standby Enables the generation of debug messages for the standby controller card.

switchover Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

csnp Optional. Enables the generation of debug messages for CSNPs.

psnp Optional. Enables the generation of debug messages for PSNPs.

`send`
Optional. Enables the generation of debug messages for sent packets.

`receive`
Optional. Enables the generation of debug messages for received packets.

`detail`
Optional. Provides detailed information with debug messages.

`interface` *`if-name`*
Optional. Interface name. You can choose to enable the generation of debug messages for events related to the specified interface.

`prefix-list` *`pl-name`*
Optional. Prefix list name. You can choose to enable the generation of debug messages for events related to the specified prefix list.

`system-id` *`sys-id`*
Optional. System ID of the specific system for which you want to enable debug messages. The system ID is the six bytes before the last byte of the network entity title (NET) and is expressed in the *`xxxx.xxxx.xxxx`* format.

1.65.4 Default

The generation of debug messages is disabled.

1.65.5 Usage Guidelines

Use the **`debug isis snp-packets`** command to enable the generation of debug messages related to IS-IS CSNPs and PSNPs.

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 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 related to IS-IS CSNPs and PSNPs.

1.65.6 Examples

The following example shows how to enable the generation of debug messages related to IS-IS CSNPs and PSNPs:

```
[local]Redback#debug isis snp-packets detail
```

The following example displays typical debug messages generated by the system when the generation of debug messages related to IS-IS CSNPs and PSNPs is enabled:

```
Feb 20 00:12:04: %ISIS-7-CSNPPKT_SD:
sent L1 CSNP on intf 1/1

Feb 20 00:12:04: %ISIS-7-CSNPPKT_SD:
send L1 CSNP Pkt Len:  195, fr IS 1010.1010

.a2a2, 0000.0000.0000.0000-ffff.ffff.ffff.ffff


   0    83 21 01 06 18 01 00 03 00 c3 10 10 10 10 a2 a2

  16    00 00 00 00 00 00 00 00 00 ff ff ff ff ff ff ff

  32    ff 09 a0 02 3f 10 10 10 10 a1 a1 00 00 00 00 03

  48    2b 28 04 02 fc 10 10 10 10 a1 a1 02 00 00 00 03

  64    1a 22 ae 03 00 10 10 10 10 a2 a2 00 00 00 00 02

  80    c2 68 c3 03 00 10 10 10 10 a2 a2 01 00 00 00 02

  96    b1 b2 1b 03 07 10 10 10 10 a3 a3 00 00 00 00 03

 112    08 c3 ec 02 8b 10 10 10 10 c2 c2 00 00 00 00 03

 128    36 7c f2 04 1d 10 10 10 10 c2 c2 01 00 00 00 03

 144    15 96 c2 01 b7 10 10 10 10 c2 c2 02 00 00 00 03

 160    09 a7 bc 02 94 10 10 10 10 e2 e2 00 00 00 00 02

 176    d1 ed 4c 02 94 10 10 10 10 e2 e2 02 00 00 00 02

 192    b5 8f 34
```

1.66 debug isis spf-events

debug [**boot** {**active** | **standby**} | **switchover**] **isis spf-events** [**detail**] [**interface** *if-name*] [**prefix-list** *pl-name*] [**system-id** *sys-id*]

no debug [**boot** {**active** | **standby**} | **switchover**] **isis spf-events**

1.66.1 Purpose

Enables the generation of debug messages for events related to Shortest Path First (SPF) computation within Intermediate System-to-Intermediate System (IS-IS).

1.66.2 Command Mode

exec (10)

1.66.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
`detail`	Optional. Provides detailed information with debug messages.
`interface` *if-name*	Optional. Interface name. You can choose to enable debug messages for events related to the specified interface.
`prefix-list` *pl-name*	Optional. Prefix list name. You can choose to enable debug messages for events related to the specified prefix list.
`system-id` *sys-id*	Optional. System ID of the specific system for which you want to enable debug messages. The system ID is the six bytes before the last byte of the network entity title (NET) and is expressed in the *xxxx.xxxx.xxxx* format.

1.66.4 Default

The generation of debug messages is disabled.

1.66.5 Usage Guidelines

Use the **debug isis spf-events** command to enable the generation of debug messages for events related to SPF computation within IS-IS.

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 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 events related to SPF computation within IS-IS.

1.66.6 Examples

The following example shows how to enable the generation of debug messages for events related to SPF computation within IS-IS:

```
[local]Redback#debug isis spf-events detail
```

The following example displays typical debug messages generated by the system when the generation of debug messages related to SPF computation within IS-IS is enabled:

```
ISIS: Created L1 background job 0x2815f720
ISIS: Running L1 background job 0x2815f720
ISIS: shortest path for L1 metric default
ISIS: ... following path entry @ pointer $2818e0cc added to L1 tent
ISIS:       Local Destination 02aa.0a00.0088.00 at cost 0
ISIS: ... minimal path entry found in L1 tent @ pointer $2818e0cc
ISIS:       Local Destination 02aa.0a00.0088.00 at cost 0
ISIS: ... added path entry @ pointer $2818e0cc to L1 paths
ISIS:       Local Destination 02aa.0a00.0088.00 at cost 0
ISIS: ... minimal path entry found in L1 tent @ pointer $0
ISIS: Cancelled L1 background job 0x2815f720
```

1.67 debug isis spf-triggers

debug [**boot** {**active** | **standby**} | **switchover**] **isis spf-triggers** [**detail**] [**interface** *if-name*] [**prefix-list** *pl-name*] [**system-id** *sys-id*]

no debug [**boot** {**active** | **standby**} | **switchover**] **isis spf-triggers**

1.67.1 Purpose

Enables the generation of debug messages for events related to the causes of triggered Shortest Path First (SPF) runs within an Intermediate System-to-Intermediate System (IS-IS).

1.67.2 Command Mode

exec (10)

1.67.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

standby Enables the generation of debug messages for the standby controller card.

switchover Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

`detail` Optional. Provides detailed information with debug messages.

`interface` *`if-name`* Optional. Interface name. You can choose to enable debug messages for events related to the specified interface.

`prefix-list` *`pl-name`* Optional. Prefix list name. You can choose to enable debug messages for events related to the specified prefix list.

`system-id` *`sys-id`* Optional. System ID of the specific system for which you want to enable debug messages. The system ID is the six bytes before the last byte of the network entity title (NET) and is expressed in the *`xxxx.xxxx.xxxx`* format.

1.67.4 Default

The generation of debug messages is disabled.

1.67.5 Usage Guidelines

Use the **`debug isis spf-triggers`** command to enable the generation of debug messages for events related to the causes of triggered SPF runs within IS-IS.

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 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 events related to the causes of triggered SPF runs within IS-IS.

1.67.6 Examples

The following example shows how to enable the generation of debug messages for events related to the causes of triggered SPF runs within IS-IS:

```
[local]Redback#debug isis spf-triggers
```

The following example displays typical debug messages generated by the system when the generation of debug messages related to the causes of triggered SPF runs within IS-IS is enabled:

```
Oct 31 20:52:37: %ISIS-7-SPF_TRIG: trigger
L2 spf due to updating lsp

Oct 31 20:52:37: %ISIS-7-SPF_TRIG: trigger
L2 spf due to adding new lsp

Oct 31 20:55:06: %ISIS-7-SPF_TRIG: trigger
L1 spf due to updating lsp
```

1.68 debug key-chain

debug [{**boot** {**active** | **standby**} | **switchover**}] **key-chain**

no debug [{**boot** {**active** | **standby**} | **switchover**}] **key-chain**

1.68.1 Purpose

Enables the generation of debug messages for key chain maintenance.

1.68.2 Command Mode

exec (10)

1.68.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

standby	Enables the generation of debug messages for the standby controller card.
switchover	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

1.68.4 Default

The generation of debug messages for key chain maintenance is disabled.

1.68.5 Usage Guidelines

Use the **debug key-chain** command to enable the generation of debug messages for key chain maintenance.

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 when enabling the generation of any debug messages on a production system.

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 debug messages for key chain maintenance.

1.68.6 Examples

The following example enables debug messages for key chain maintenance:

```
[local]Redback#debug policy key-chain
```

1.69 debug l2tp

debug [**boot** {**active** | **standby**} | **switchover**] **l2tp** ***l2tp-event***

no debug [**boot** {**active** | **standby**} | **switchover**] **l2tp** ***l2tp-event***

1.69.1 Purpose

Enables the generation of debug messages for Layer 2 Tunneling Protocol (L2TP)-related events.

1.69.2 Command Mode

exec (10)

1.69.3 Syntax Description

boot	Optional. Enables the generation of debug messages during a system reload.
active	Enables the generation of debug messages for the active controller card.
standby	Enables the generation of debug messages for the standby controller card.
switchover	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
l2tp-event	Event type (Table 5) for which debug messages are generated.

1.69.4 Default

The generation of debug messages for L2TP-related events is disabled.

1.69.5 Usage Guidelines

Use the **debug l2tp** command to enable the generation of debug messages for L2TP-related events.

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.

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.

Table 5 lists the types of L2TP events for which you can generate debug messages.

Table 5 L2TP Event Types

Event Type Keyword	Description
`aaa`	L2TP-related authentication, authorization, and accounting (AAA) events.
`all`	All L2TP-related events.
`avp`	L2TP attribute-value pairs (AVPs) transmitted or received in L2TP control messages.
`circuit`	L2TP-related circuit events.
`group`	L2TP-related group events, including the selection of a peer for a given session.
`ipc`	L2TP-related interprocess communication (IPC) events.
`ism both`	L2TP-related Interface and Circuit State Manager (ISM) messages in both directions.
`ism in`	L2TP-related ISM incoming messages.
`ism out`	L2TP-related ISM outgoing messages.
`misc`	L2TP-related miscellaneous events.
`packet`	L2TP-related packet transmit (TX) and receive (RX) events for L2TP control messages.
`peer`	L2TP-related peer events.
`ppa`	L2TP-related Packet Processing ASIC (PPA) events.
`ppp`	L2TP-related Point-to-Point Protocol (PPP) events.
`rcm`	L2TP-related Router Configuration Manager (RCM) events.
`route`	L2TP routes to peers events.
`ses-abort`	L2TP-related abnormal termination of session events.
`ses-fsm`	L2TP-related session finite-state-machine events.
`ses-setup`	L2TP-related session setup events.
`tun-fsm`	L2TP-related tunnel finite-state-machine events.

Table 5 L2TP Event Types

Event Type Keyword	Description
tun-setup	L2TP-related tunnel setup events.
window	L2TP-related control window events, including out-of-order or retransmitted control 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 for L2TP-related events.

1.69.6 Examples

The following example shows how to enable the generation of debug messages for L2TP packet Tx and Rx events:

```
[local]Redback#debug l2tp packet
```

1.70 debug l2vpn

debug l2vpn [**all** | **config**]

1.70.1 Purpose

Enables the generation of debug messages for Layer 2 VPN (L2VPN) events.

1.70.2 Command Mode

exec (10)

1.70.3 Syntax Description

`all` Enables the generation of debug messages for all L2VPN events.

`config` Enables the generation of debug messages for L2VPN configuration events.

1.70.4 Default

No debug messages are generated for L2VPN events.

1.70.5 Usage Guidelines

Use the **`debug l2vpn`** command to enable the generation of debug messages for all L2VPN events.

1.70.6 Examples

The following example shows how to enable the generation of debug messages for all L2VPN events:

```
[local]Redback#debug l2vpn all
```

1.71 debug ldp all

`debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`ldp all`**

`no debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`ldp all`**

1.71.1 Purpose

Enables the generation of debug messages for all Label Distribution Protocol (LDP) events.

1.71.2 Command Mode

exec (10)

1.71.3 Syntax Description

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

1.71.4 Default

Disabled

1.71.5 Usage Guidelines

Use the **`debug ldp all`** command to enable the generation of debug messages for all LDP events.

Whenever possible, use a more specific **`debug ldp`** command to prevent the generation of unnecessary debug messages.

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 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 all LDP events.

1.71.6 Examples

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

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

1.72 debug ldp filter

debug [**boot** {**active** | **standby**} | **switchover**] **ldp filter** {**interface** *if-name* | **neighbor** *ip-addr* | **prefix** *ip-addr/prefix-length* [**exact-match**]}

no debug [**boot** {**active** | **standby**} | **switchover**] **ldp filter** {**interface** *if-name* | **neighbor** *ip-addr* | **prefix** *ip-addr/prefix-length* [**exact-match**]}

1.72.1 Purpose

Limits the generation of Label Distribution Protocol (LDP) debug messages to specific interfaces, neighbor IP addresses, or IP prefixes.

1.72.2 Command Mode

exec (10)

1.72.3 Syntax Description

`boot` Optional. Enables the generation of debug messages during a system reload.

`active` Enables the generation of debug messages for the active controller card.

`standby` Enables the generation of debug messages for the standby controller card.

`switchover` Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

`interface` *`if-name`*	Name of the interface for which LDP debug messages are to be generated. Debug messages are generated only for LDP messages that are sent from or received by the specified interface.
`neighbor` *`ip-addr`*	Neighbor IP address, in the form *`A.B.C.D`*, for which LDP debug messages are to be generated. Debug messages are generated only for LDP messages sent to or received by the neighbor at the specified IP address.
`prefix`	Enables the generation of LDP debug messages for neighbors with IP addresses that match the specified IP prefix.
`ip-addr/prefix-length`	Specifies the IP address, in the form *`A.B.C.D`*, and the prefix length, separated by the slash (/) character. The range of values for the *`prefix-length`* argument is 0 to 32.
`exact-match`	Optional. Used with the *`ip-addr/prefix-length`* construct. Performs an exact match on the IP prefix. If this keyword is not specified, a longest prefix match is performed.

1.72.4 Default

Disabled

1.72.5 Usage Guidelines

Use the **`debug ldp filter`** command to limit the generation of debug messages to specific interfaces, neighbor IP addresses, or IP prefixes. Filters reduce the number of debug messages that are generated.

If the LDP neighbor's transport IP address differs from its router ID, the IP address specified in the **`neighbor`** *`ip-addr`* construct must be the LDP neighbor's transport IP address.

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 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 LDP debug messages for specific interfaces, neighbor IP addresses, or IP prefixes.

1.72.6 Examples

The following example shows how to enable the generation of LDP debug messages for the neighbor at IP address, `10.1.1.1`:

```
[local]Redback#debug ldp filter neighbor 10.1.1.1
```

1.73 debug ldp message

debug [**boot** {**active** | **standby**} | **switchover**] **ldp message** {***msg-type*** [**detail**] | {**dump** | **receive** | **send**}}

no debug [**boot** {**active** | **standby**} | **switchover**] **ldp message** {***msg-type*** [**detail**] | {**dump** | **receive** | **send**}}

1.73.1 Purpose

Enables the generation of debug messages for Label Distribution Protocol (LDP) messages.

1.73.2 Command Mode

exec (10)

1.73.3 Syntax Description

boot Optional. Enables the generation of debug messages during a system reload.

active Enables the generation of debug messages for the active controller card.

standby Enables the generation of debug messages for the standby controller card.

switchover Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

msg-type Type of LDP message for which the generation of debug messages is enabled; see Table 6 for a list of values for this argument.

detail Optional. Displays detailed information in the debug messages.

dump Enables hex output for generated LDP debug messages.

receive Enables the generation of debug messages for LDP messages received from LDP peers.

send Enables the generation of debug messages for LDP messages sent to LDP peers.

1.73.4 Default

Disabled

1.73.5 Usage Guidelines

Use the **debug ldp message** command to enable the generation of debug messages for LDP messages. Table 6 lists the keywords for the ***msg-type*** argument.

Table 6 LDP Message Types

Keyword	Message Type
address	LDP address and address withdraw messages
general	LDP general messages
hello	LDP hello messages
initialization	LDP initialization messages
keepalive	LDP keepalive messages
label	LDP label mapping and label withdraw messages
notification	LDP notification messages
release	LDP label release messages

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 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 LDP messages.

1.73.6 Examples

The following example shows how to enable the generation of debug messages for LDP label mapping and withdrawal messages received from LDP peers, but disables the generation of LDP debug messages sent to peers:

```
[local]Redback#debug ldp message label

[local]Redback#no debug ldp message send
```

1.74 debug ldp protocol

`debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`ldp protocol`** {**`event`** | **`ism`** | **`l2vpn`** | **`lm`** | **`policy`** | **`rib`** | **`session`** | **`update`**} [**`detail`**]

`no debug` [**`boot`** {**`active`** | **`standby`**} | **`switchover`**] **`ldp protocol`** {**`event`** | **`ism`** | **`l2vpn`** | **`lm`** | **`policy`** | **`rib`** | **`session`** | **`update`**} [**`detail`**]

1.74.1 Purpose

Enables the generation of debug messages for Label Distribution Protocol (LDP) protocol interaction.

1.74.2 Command Mode

exec (10)

1.74.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
`event`	Enables the generation of debug messages for LDP events that do not apply to a specific event category.
`ism`	Enables the generation of debug messages for LDP interaction with Interface and Circuit State Manager (ISM) events.
`l2vpn`	Enables the generation of debug messages for LDP interaction with Layer 2 VPN (L2VPN) events.
`lm`	Enables the generation of debug messages for LDP interaction with Label Manager (LM) events.
`policy`	Enables the generation of debug messages for LDP routing policy library usage.
`rib`	Enables the generation of debug messages for the LDP interaction with the Routing Information Base (RIB).
`session`	Enables the generation of debug messages for LDP session events.
`update`	Enables the generation of debug messages for LDP update process.
`detail`	Optional. Displays detailed information in the debug messages.

1.74.4 Default

Disabled

1.74.5 Usage Guidelines

Use the **`debug ldp protocol`** command to enable the generation of debug messages for LDP protocol interaction.

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 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 LDP interaction.

1.74.6 Examples

The following example shows how to enable the generation of debug messages for LDP interaction with L2VPN events:

```
[local]Redback#debug ldp protocol l2vpn
```

1.75 debug lg

debug [**boot** {**active** | **standby**} | **switchover**] **lg** {**acct** | **config** | **ism** | **rcm**}

no debug [**boot** {**active** | **standby**} | **switchover**] **lg** {**acct** | **config** | **ism** | **rcm**}

1.75.1 Purpose

Enables the generation of debug messages for link group events.

1.75.2 Command Mode

exec (10)

1.75.3 Syntax Description

`boot`	Optional. Enables the generation of debug messages during a system reload.
`active`	Enables the generation of debug messages for the active controller card.
`standby`	Enables the generation of debug messages for the standby controller card.
`switchover`	Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.
`acct`	Enables the generation of debug messages for aggregated circuit-related events.
`config`	Enables the generation of debug messages for constituent and aggregated circuit-related events.
`ism`	Enables the generation of debug messages for Interface State and Circuit Manager (ISM) related events.
`rcm`	Enables the generation of debug messages for Router Configuration Manager (RCM) related events.

1.75.4 Default

The generation of debug messages for link groups is disabled.

1.75.5 Usage Guidelines

Use the **`debug lg`** command to enable the generation of debug messages for link group events. Aggregated circuits are the logical ports or channels created by the link group; each aggregated circuit consists of a number of physical circuits, also referred to as constituent circuits.

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.

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.

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 link groups.

1.75.6 Examples

The following example shows how to enable the generation of constituent and aggregated circuit-related debug messages for link groups:

```
[local]Redback#debug lg config
```

1.76 debug lm

debug [**boot** {**active** | **standby**} | **switchover**] **lm** {**all** | **download** | **func** | **in-label** [*label-space-id*] | **intf** | **ism** | **l2vpn** | **lsp** [*ip-addr/prefix-length*] | **msg** [*ip-addr/prefix-length* | **client** [**bgp** [*ip-addr/prefix-length*] | **ldp** [*ip-addr/prefix-length*] | **mpls-static** [*ip-addr/prefix-length*]| **rsvp** [*ip-addr/prefix-length*]]] | **nexthop** [*ip-addr/prefix-length*] | **ping** | **rib** [*ip-addr/prefix-length* | **vpls**]}

no debug [**boot** {**active** | **standby**} | **switchover**] **lm** {**all** | **download** | **intf** | **ism** | **l2vpn** | **lsp** [*ip-addr/prefix-length*] | **msg** [*ip-addr/prefix-length* | **client** [**bgp** [*ip-addr/prefix-length*] | **ldp** [*ip-addr/prefix-length*] | **mpls-static** [*ip-addr/prefix-length*]| **rsvp** [*ip-addr/prefix-length*]]] | **nexthop** [*ip-addr/prefix-length*] | **ping** | **rib** [*ip-addr/prefix-length*]}

1.76.1 Purpose

Enables the generation of debug messages for label manager (LM) activities.

1.76.2 Command Mode

exec (10)

1.76.3 Syntax Description

`boot`
Optional. Enables the generation of debug messages during a system reload.

`active`
Enables the generation of debug messages for the active controller card.

`standby`
Enables the generation of debug messages for the standby controller card.

`switchover`
Optional. Enables the generation of debug messages during a switchover from the active to the standby controller.

`all`
Enables the generation of all LM debug messages.

`download`
Enables the generation of traffic card download event debug messages.

`func`
Enables the generation of debug messages for label manager function flow.

`in-label` []
Enables the generation of in-label debug messages.

`label-space-id`
Optional with the **`in-label`** keyword. Limits the output to in-label debug messages for a specific label space ID; the range is 0 to 1048575.

`intf`
Enables the generation of interface event debug messages.

`ism`
Enables the generation of Interface and Circuit State Manager (ISM) event debug messages.

`l2vpn`
Enables the generation of Layer 2 Virtual Private Network (L2VPN) event debug messages.

`lsp`
Enables the generation of label-switched path (LSP) download debug messages. When used with the *`ip-addr/prefix-length`* construct, only LSP download debug messages for routes that match the IP address range are generated.

`msg`
Enables the generation of internal debug messages exchanged between the LM and other processes within the system. When used with the *`ip-addr/prefix-length`* construct, only internal debug messages for routes that match the IP address range are generated.

ip-addr/prefix-length — Optional. IP address (in the form *A.B.C.D*) and prefix length, separated by the slash (/) character. The range of values for the *prefix-length* argument is 0 to 32. Only debug messages for routes that match the IP address range are generated.

client — Enables the generation of internal debug messages exchanged between the LM and a specific client.

bgp — Optional with the **msg client** construct. Enables the generation of internal debug messages exchanged between the LM and the Border Gateway Protocol (BGP) client. When used with the *ip-addr/prefix-length* construct, only internal debug messages between the LM and the BGP client for routes that match the IP address range are generated.

ldp — Optional with the **msg client** construct. Enables the generation of internal debug messages exchanged between the LM and the Label Distribution Protocol (LSP) client. When used with the *ip-addr/prefix-length* construct, only internal debug messages between the LM and the LDP client for routes that match the IP address range are generated.

mpls-static — Optional with the **msg client** construct. Enables the generation of internal debug messages exchanged between the LM and the Multiprotocol Label Switching (MPLS)-static client. When used with the *ip-addr/prefix-length* construct, only internal debug messages between the LM and the MPLS-static client for routes that match the IP address range are generated.

rsvp — Optional with the **msg client** construct. Enables the generation of internal debug messages exchanged between the LM and the Resource Reservation Protocol (RSVP) client. When used with the *ip-addr/prefix-length* construct, only internal debug messages between the LM and the RSVP client for routes that match the IP address range are generated.

nexthop — Enables the generation of next-hop event debug messages. When used with the *ip-addr/prefix-length* construct, only next-hop event debug messages for routes that match the IP address range are generated.

ping — Enables the generation of RSVP LSP ping event debug messages.

rib — Enables the generation of Routing Information Base (RIB) event debug messages. When used with the *ip-addr/prefix-length* construct, only RIB event debug messages for routes that match the IP address range are generated.

vpls — Enables the generation of VPLS label-management debug messages.

1.76.4 Default

The generation of debug messages is disabled.

1.76.5 Usage Guidelines

Use the **debug lm** command to enable the generation of debug messages for LM activities.

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 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 LM activities.

1.76.6 Examples

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

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

1.77 debug logger

```
debug logger
```

```
no debug logger
```

1.77.1 Purpose

Enables the generation of debug messages for the logging facility (logger).

1.77.2 Command Mode

exec (10)

1.77.3 Syntax Description

This command has no keywords or arguments.

1.77.4 Default

The generation of debug messages for the logger is disabled.

1.77.5 Usage Guidelines

Use the **`debug logger`** command to enable the generation of debug messages for the logger.

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.

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 the logger.

1.77.6 Examples

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

```
[local]Redback#debug logger
```

1.78 debug logger-rcm

`debug logger-rcm`

`no debug logger-rcm`

1.78.1 Purpose

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

1.78.2 Command Mode

exec (10)

1.78.3 Syntax Description

This command has no keywords or arguments.

1.78.4 Default

The generation of debug messages for the RCM logger is disabled.

1.78.5 Usage Guidelines

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

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.

To store these 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 these 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 the logger RCM.

1.78.6 Examples

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

```
[local]Redback#debug logger-rcm
```