



# IGMP and MLD Configuration Guide, 17.2.0

# Contents

About this Guide.....	7
IGMP and MLD overview.....	8
Comparing IGMP and MLD.....	8
IGMP.....	8
Joining and leaving a multicast group by using IGMP.....	8
IGMP messages.....	8
IGMP versions.....	9
MLD.....	9
Joining and leaving a multicast group using MLD.....	9
MLD messages.....	9
MLD versions.....	10
Supported standards.....	10
Supported MIBs.....	10
IGMP and MLD configuration.....	11
IGMP commands.....	12
interfaces ip igmp.....	12
interfaces ip igmp access-group.....	14
interfaces ip igmp enforce-router-alert.....	15
interfaces ip igmp immediate-leave group-list.....	16



- interfaces ip igmp join-group..... 16
- interfaces ip igmp last-member-query-count..... 17
- interfaces ip igmp last-member-query-interval..... 18
- interfaces ip igmp limit..... 19
- interfaces ip igmp limit-exception..... 19
- interfaces ip igmp offlink..... 20
- interfaces ip igmp querier-timeout..... 21
- interfaces ip igmp query-interval..... 21
- interfaces ip igmp query-max-response-time..... 22
- interfaces ip igmp robustness-variable..... 23
- interfaces ip igmp startup-query-count..... 23
- interfaces ip igmp startup-query-interval..... 24
- interfaces ip igmp version..... 25
- interfaces ip igmp static-group source..... 25
- monitor protocol multicast..... 26
- protocols igmp limit..... 27
- protocols igmp log..... 27
- protocols igmp ssm-map..... 28
- protocols igmp ssm-map static access-list source..... 28
- reset ip igmp..... 29
- show ip igmp groups..... 30



- show ip igmp interface..... 30
- show ip igmp ssm-map..... 31
- show monitoring protocols multicast igmp..... 31
- MLD Commands..... 33
  - interfaces interface ipv6 mld..... 33
  - interfaces interface ipv6 mld access-group acl6-name..... 35
  - interfaces interface ipv6 mld immediate-leave group-list acl6-name..... 36
  - interfaces interface ipv6 mld last-member-query-count count..... 37
  - interfaces interface ipv6 mld last-member-query-interval interval..... 38
  - interfaces interface ipv6 mld limit limit..... 38
  - interfaces interface ipv6 mld limit-exception acl6-name..... 39
  - interfaces interface ipv6 mld querier-timeout interval..... 40
  - interfaces interface ipv6 mld query-interval interval..... 40
  - interfaces interface ipv6 mld query-max-response-time interval..... 41
  - interfaces interface ipv6 mld robustness-variable variable..... 42
  - interfaces interface ipv6 mld static-group group source source..... 43
  - interfaces interface ipv6 mld version version..... 43
- monitor protocol multicast (MLD)..... 44
- protocols mld limit limit..... 44
- protocols mld log..... 45
- protocols mld ssm-map..... 46
- protocols mld ssm-map static acl6-name source source..... 46



reset ipv6 mld.....	47
show ipv6 mld groups.....	48
show ipv6 mld interface.....	48
show ipv6 mld ssm-map.....	49
show monitoring protocols multicast mld.....	49
VRF support.....	50
Command support for VRF routing instances.....	50
List of Acronyms.....	53

# Copyright Statement

© 2017 AT&T Intellectual Property. All rights reserved. AT&T and Globe logo are registered trademarks of AT&T Intellectual Property. All other marks are the property of their respective owners.

The training materials and other content provided herein for assistance in training on the Vyatta vRouter may have references to Brocade as the Vyatta vRouter was formerly a Brocade product prior to AT&T's acquisition of Vyatta. Brocade remains a separate company and is not affiliated to AT&T.



# About This Guide

This guide describes how to configure IGMP and MLD on AT&T products that run on the AT&T Vyatta Network Operating System (referred to as a virtual router, vRouter, or router in the guide).



# IGMP and MLD overview

---

## Comparing IGMP and MLD

One of the main aspects of multicast routing is managing the network devices that receive the multicast. The key abstraction in multicasting is the multicast group: a given multicast stream is associated with a group, which is identified by a specific multicast IP address. Group members receive the multicast stream; nonmembers do not.

In IPv4 networks, multicast group membership is managed by using IGMP. In IPv6, the corresponding protocol for managing group membership is MLD. For more information on multicast routing fundamentals and Protocol Independent Multicast (PIM), refer to AT&T Vyatta Network Operating System Multicast Routing Configuration Guide and AT&T Vyatta Network Operating System PIM Configuration Guide.

---

## IGMP

This section presents the following topics:

- [Joining and leaving a multicast group by using IGMP \(page 8\)](#)
- [IGMP messages \(page 8\)](#)
- [IGMP versions \(page 9\)](#)

### Joining and leaving a multicast group by using IGMP

This section describes the behavior of IGMP version 2.

IGMP allows a network host to inform a router that it is interested in receiving a particular multicast stream.

To begin, the multicast group is assigned a multicast address (that is, an IP address in the 224.0.0.0/4 class D address space). Hosts register to receive the stream join the group by sending an IGMP Report to the upstream multicast router. The router then adds that group to the list of multicast groups that should be forwarded onto the local subnet.

The router does not maintain state about which hosts on the subnet are to receive traffic for the group. Instead, the router continues to send traffic to the subnet until either a timeout value expires or there are no more hosts in that group on the subnet.

When a host no longer wants to receive multicast traffic, it sends the router an IGMP Leave message. After receiving this message, the router sends a query to the local subnet to determine whether any group members remain, sending the message to all hosts on the subnet, at the multicast All-Hosts address (224.0.0.1). If any host responds, the router continues to send to the group; if not, the router removes the multicast group from its forwarding list and stops sending to the group.

**Note:** The behavior of IGMP version 1 and version 3 varies from version 2.

### IGMP messages

IGMP communicates in three types of messages:

- **Report (Join):** A host sends an unsolicited message to the upstream multicast router signaling that it wants to become a member of a specific multicast group.
- **Leave Group (Leave):** A host in a multicast group sends a message to the upstream multicast router signaling that it is leaving a multicast group.
- **Query:** The multicast router sends a message to the local router to determine which groups have members on the attached network, or to determine if a specific group has members on the attached network.





## IGMP versions

Three versions of IGMP are specified:

- IGMPv1, defined by RFC 1112, *Host Extensions for IP Multicasting*
- IGMPv2, defined by RFC 2236, *Internet Group Management Protocol, Version 2*
- IGMPv3, defined by RFC 3376, *Internet Group Management Protocol, Version 3* and updated by RFC 4604, *Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast*

Relative to IGMPv1, IGMPv2 adds the ability for a host to leave a multicast group. Relative to IGMPv2, IGMPv3 adds support for source-specific multicast. For more information on IGMPv3 support for source-specific multicast, refer to AT&T Vyatta Network Operating System Multicast Routing Configuration Guide.

---

## MLD

This section presents the following topics:

- [Joining and leaving a multicast group using MLD \(page 9\)](#)
- [MLD messages \(page 9\)](#)
- [MLD versions \(page 10\)](#)

## Joining and leaving a multicast group using MLD

MLD exchanges group information between hosts and multicast routers on IPv6 networks. MLD is based on IGMP; however, while IGMP is a distinct transport-layer protocol, MLD is an extension of Internet Control Message Protocol for Internet Protocol version 6 (ICMPv6).

An IPv6 host that wants to receive a multicast stream joins the multicast group of address range FF00::/8, by sending a Multicast Listener Report message with the multicast group of interest as the destination address. The router registers the host for that multicast group and forwards multicast traffic to the local network.

An IPv6 host in a multicast group also receives Report messages sent to the multicast address from other hosts joining the group. Having hosts track these Reports allows the network to manage the MLD leave process.

When a host leaves a multicast group, it checks to see whether it ever received a Report for another host in the group. If not, the host knows it was the last host to join the group. In this case, the host sends a Multicast Listener Done message to signal the router that it is leaving the group. The message is sent to the IPv6 All-Routers multicast group address (FF02::2).

When the router receives a Done message, it responds with a multicast address-specific Multicast Listener Query. The Query requests any remaining group members to report their existence with a Report message. If any group member responds to the Query, the router continues multicast forwarding for this group. If no host responds, the router stops forwarding.

## MLD messages

MLD messages usually use the IPv6 link-local as the source address. The hop limit is always set to 1, to prevent the router from forwarding the message. MLD messages are of three types:

- [Multicast listener report \(page 9\)](#)
- [Multicast listener done \(page 10\)](#)
- [Multicast listener query \(page 10\)](#)

### Multicast listener report

The Multicast Listener Report message is sent by a listening IPv6 host. The message either is unsolicited, sent to report its interest in receiving specific multicast traffic, or sent to respond to a Multicast Listener Query message from the multicast router.

The Multicast Listener Report message is equivalent to an IGMPv2 Host Membership Report message. It is ICMPv6 message type 131.



## Multicast listener done

The Multicast Listener Done message is sent by a host in a specific multicast group to the multicast router, to signal the router that there may not be any further group members on the local subnet. When the router receives a Done message, it queries the subnet for further group members by sending a Multicast Listener Query message.

The Multicast Listener Report message is equivalent to an IGMPv2 Leave Group message. It is ICMPv6 message type 132.

## Multicast listener query

The Multicast Listener Query message is sent by the multicast router to a local subnet, to determine whether any multicast group members still exist on the subnet. Multicast Listener Query messages are of two types:

- General: The router periodically sends a general Query to poll all hosts on the subnet for the presence of any multicast address. Link-local addresses, the All-Nodes multicast address (FF02::1), and reserved multicast addresses (addresses with a scope of 0) and interface-local addresses (addresses with a scope of 1) are excluded from the poll.
- Multicast-address-specific. The router sends a multicast-address-specific query only to members of a specific multicast group on the subnet.

The Multicast Listener Report message is equivalent to an IGMPv2 Host Membership Query message. It is ICMPv6 message type 130.

## MLD versions

Two versions of MLD are specified:

- MLDv1, defined by RFC 2710, *Multicast Listener Discovery for IPv6*
- MLDv2, defined by RFC 3810, *Multicast Listener Discovery version 2 (MLDv2) for IPv6*

MLDv1 is based on IGMPv2. MLDv2 is based on IGMPv3.

In addition, RFC 4604: *Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast* extends MLD to support source-specific multicast. For more information, refer to AT&T Vyatta Network Operating System Multicast Routing Configuration Guide).

---

## Supported standards

The AT&T implementation of IGMP and MLD complies with the following standards:

- RFC 1112: *Host Extensions for IP Multicasting*
- RFC 2236: *Internet Group Management Protocol, Version 2*
- RFC 2710: *Multicast Listener Discovery (MLD) for IPv6*
- RFC 3376: *Internet Group Management Protocol, Version 3*
- RFC 3810: *Multicast Listener Discovery version 2 (MLDv2) for IPv6*
- RFC 4604: *Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast*

---

## Supported MIBs

The AT&T implementation of IGMP supports the following Simple Management Network Protocol (SNMP) management information bases (MIBs).

- IGMP-MIB, RFC2933:*Internet Group Management Protocol MIB*
- IPMROUTE, RFC 2932:*IPv4 Multicast Routing MIB*
- MLD-MIB, RFC 3019:*IP Version 6 Management Information Base for The Multicast Listener Discovery Protocol*

For a list of all MIBs supported, refer to AT&T Vyatta Network Operating System Remote Management Configuration Guide.



---

## IGMP and MLD configuration

IGMP and MLD configurations depend on other multicast-related commands. For this reason, the configuration examples are located elsewhere. For IGMP and MLD configuration examples, refer to AT&T Vyatta Network Operating System Multicast Routing Configuration Guide.



# IGMP Commands

## interfaces <interface> ip igmp

Enables IGMP on an interface.

**Syntax:**

```
set interfaces interface ip igmp
```

**Syntax:**

```
delete interfaces interface ip igmp
```

**Syntax:**

```
show interfaces interface ip igmp
```

**interface**

The type of interface. For detailed keywords and arguments for interfaces that support multicast routing, see the table in the Usage Guidelines below.

**Configuration mode**

```
interfaces interface {  
  ip {  
    igmp {  
    }  
  }  
}
```

Use this command to enable the Internet Group Management Protocol (IGMP) on an interface.

**Note:** Enabling IP on an interface enables the host side functionality of IGMP by default. The `set interfaces interface ip igmp` command enables the router side functionality of the IGMP on the given interface.

**Note:** To use IGMP for multicast routing, multicast routing must be enabled on the router. For information about multicast routing in general, see the AT&T Vyatta Network Operating System Multicast Routing Configuration Guide.

The following table shows the syntax and parameters for interface types. Some of these types may not apply to this command.

Interface Type	Syntax	Parameters
Bridge	bridge <i>brx</i>	<i>brx</i> : The name of a bridge group. The name ranges from br0 through br999.



Interface Type	Syntax	Parameters
Data plane	<code>dataplane dpxy&gt;pzv</code>	<p>dpxypyz: The name of a data plane interface to which the following applies:</p> <ul style="list-style-type: none"><li>dpx identifies a data plane ID</li></ul> <p><b>Note:</b> Currently, only dp0 is supported.</p> <ul style="list-style-type: none"><li>py specifies a physical or virtual PCI slot index</li><li>pz specifies a port index</li></ul> <p>Other supported name formats are the following:</p> <ul style="list-style-type: none"><li>dpxemy—used for LAN-on-motherboard (LOM) devices that do not have a PCI slot. emy specifies an embedded network interface number.</li><li>dpxporty—used for devices in which the PCI slot cannot be identified. porty specifies a port index.</li></ul>
Data plane vif	<code>dataplane dpxypyz vif <i>vif-id</i> [vlan <i>vlan-id</i> ]</code>	<p>dpxypyz: The name of a data plane interface to which the following applies:</p> <ul style="list-style-type: none"><li>dpx specifies a data plane ID</li></ul> <p><b>Note:</b> Currently, only dp0 is supported.</p> <ul style="list-style-type: none"><li>py specifies a physical or virtual PCI slot index</li><li>pz specifies a port index</li></ul> <p>Other supported name formats are the following:</p> <ul style="list-style-type: none"><li>dpxemy—used for LAN-on-motherboard (LOM) devices that do not have a PCI slot. emy specifies an embedded network interface number.</li><li>dpxporty—used for devices in which the PCI slot cannot be identified. porty specifies a port index.</li></ul> <p><i>vif-id</i>: A virtual interface ID. The ID ranges from 1 through 4094.</p> <p><i>vlan-id</i>: The VLAN ID of a virtual interface. The ID ranges from 1 through 4094.</p>



Interface Type	Syntax	Parameters
Loopback	loopback <i>lo</i>	<i>lo</i> : The name of a loopback interface.
OpenVPN	openvpn <i>vtunx</i>	<i>vtunx</i> : The identifier of an OpenVPN interface. The identifier ranges from vtun0 through vtunx, where x is a nonnegative integer.
Tunnel	tunnel <i>tunx</i> or tunnel <i>tunx</i> parameters	<i>tunx</i> : The identifier of a tunnel interface you are defining. The identifier ranges from tun0 through tunx, where x is a nonnegative integer.
Virtual tunnel	vti <i>vtix</i>	<i>vtix</i> : The identifier of a virtual tunnel interface you are defining. The identifier ranges from vti0 through vtix, where x is a nonnegative integer.  <b>Note:</b> This interface does not support IPv6.
VRRP	interface <i>parent-if</i> vrrp vrrp-group <i>group</i> interface	<i>parent-if</i> : The type and identifier of a parent interface; for example, dataplane dp0p1p2 or bridge br999.  <i>group</i> : A VRRP group identifier.  The name of a VRRP interface is not specified. The system internally constructs the interface name from the parent interface identifier plus the VRRP group number; for example, dp0p1p2v99. Note that VRRP interfaces support the same feature set as the parent interface does.

Use the `set` form of this command to enable IGMP on an interface.

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

Use the `show` form of this command to display IGMP configuration.

---

## interfaces <interface> ip igmp access-group <acl>

Applies an access control list to the multicast local membership groups on an interface.

### Syntax:

```
set interfaces interface ip igmp access-group acl
```

### Syntax:

```
delete interfaces interface ip igmp access-group acl
```

### Syntax:



```
show interfaces interface ip igmp access-group acl
```

**interface**

A type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp \(page 12\)](#).

**acl**

A standard IP access control list number. The number ranges from 1 through 99. An access control list is a type of routing policy; see AT&T Vyatta Network Operating System Routing Policies Configuration Guide for information on creating one.

**Configuration mode**

```
interfaces interface {
  ip {
    igmp {
      access-group acl
    }
  }
}
```

Use this command to apply an access control list to the multicast local membership groups on an interface.

Use the `set` form of this command to apply the access control list.

Use the `delete` form of this command to delete the access control list.

Use the `show` form of this command to display the access control list configuration for IGMP.

---

## interfaces <interface> ip igmp enforce-router-alert

Enables strict Router Alert validation for IGMP.

**Syntax:**

```
set interfaces interface ip igmp enforce-router-alert
```

**Syntax:**

```
delete interfaces interface ip igmp enforce-router-alert
```

**Syntax:**

```
show interfaces interface ip igmp enforce-router-alert
```

A strict Router Alert validation is disabled.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp \(page 12\)](#).

**Configuration mode**

```
interfaces interface {
  ip {
    igmp {
      enforce-router-alert
    }
  }
}
```

Use this command to put strict Router Advertisement (RA) validation into effect for IGMP.

RA validation helps prevent against spoofing attacks. When strict RA validation is in effect, the router silently discards any received RA messages that do not satisfy the validity checks specified in RFC 2461.

Use the `set` form of this command to enable strict RA validation.



Use the `delete` form of this command to restore the default behavior.

Use the `show` form of this command to show RA validation configuration.

---

## interfaces <interface> ip igmp immediate-leave group-list <acl>

Minimizes latency for hosts leaving multicast groups.

**Syntax:**

```
set interfaces interface immediate-leave group-list acl
```

**Syntax:**

```
delete interfaces interface immediate-leave group-list
```

**Syntax:**

```
show interfaces interface immediate-leave group-list
```

Immediate leave is disabled.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp \(page 12\)](#).

**acl**

An access list number used to define the membership group. Supported ranges of values are:

1 to 99: IP access list number.

1300 to 1999: IP access list number in the expanded range.

Access control lists are a type of routing policy; see the AT&T Vyatta Network Operating System Routing Policies Configuration Guide for information on creating them.

**Configuration mode**

```
interfaces interface {
  ip {
    igmp {
      immediate-leave group-list acl
    }
  }
}
```

Use this command to minimize the leave latency in IGMPv2 for IGMP memberships.

When this option is not set, the router sends an IGMP Query message when a receiver host has sent a Leave message. At this point, a timeout interval goes into effect. When this option is set, the Leave message is acted on immediately, without sending the Query or waiting for the timeout period to expire.

This command applies to IGMPv2, and it applies in situations where only one receiver is connected to each interface.

Use the `set` form of this command to enable IGMPv2 immediate leave.

Use the `delete` form of this command to restore the IGMPv2 immediate leave default behavior.

Use the `show` form of this command to view IGMPv2 immediate leave configuration.

---

## interfaces <interface> ip igmp join-group <group>

Allows the router to join a multicast group.

**Syntax:**





```
set interfaces interface ip igmp join-group group [ source source ]
```

**Syntax:**

```
delete interfaces interface ip igmp join-group group
```

**Syntax:**

```
show interfaces interface ip igmp join-group group
```

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp \(page 12\)](#).

**group**

The multicast group being joined. The format is an IPv4 multicast address.

**source**

In source-specific multicast, the multicast source. The format is an IPv4 host address.

**Configuration mode**

```
interfaces interface {
  ip {
    igmp {
      join-group group
      source source
    }
  }
}
```

Use this command to add the router to a multicast group.

Use the set form of this command to add the router to a multicast group.

Use the delete form of this command to have the router leave a multicast group.

Use the show form of this command to show multicast group membership configuration.

---

## interfaces <interface> ip igmp last-member-query-count <count>

Manually sets the last member query count value.

**Syntax:**

```
set interfaces interface ip igmp last-member-query-count count
```

**Syntax:**

```
delete interfaces interface ip igmp last-member-query-count
```

**Syntax:**

```
show interfaces interface ip igmp last-member-query-count
```

The router sends two IGMP Query messages, after which it considers the host to have left the group.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp \(page 12\)](#).

**count**

The number of times the router sends a Query message after receiving a Leave message. The range is 2 to 7. The default is 2.

**Configuration mode**

```
interfaces interface {
```



```
ip {
  igmp {
    last-member-query-count count
  }
}
```

Use this command to set the number of times that the router sends a group-specific or source-group specific Query message when it receives a Leave message from a receiver host.

The router uses this Query to determine whether any members of the multicast group remain on the network. The command is sent at the interval set in [interfaces <interface> ip igmp last-member-query-interval <interval> \(page 18\)](#). If the router receives no response to the configured number of queries, the router stops forwarding to that network.

Use the `set` form of this command to set the number of last-member queries sent.

Use the `delete` form of this command to restore the default value for last-member queries.

Use the `show` form of this command to show last-member query configuration.

---

## interfaces <interface> ip igmp last-member-query-interval <interval>

Specifies the frequency at which IGMP group-specific host queries are sent.

### Syntax:

```
set interfaces interface ip igmp last-member-query-interval interval
```

### Syntax:

```
delete interfaces interface ip igmp last-member-query-interval
```

### Syntax:

```
show interfaces interface ip igmp last-member-query-interval
```

The router waits 1000 milliseconds between last-member queries.

### *interface*

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp \(page 12\)](#).

### *interval*

The interval between last-member queries, in milliseconds. The range is 1000 to 25500. The default is 1000.

### Configuration mode

```
interfaces interface {
  ip {
    igmp {
      last-member-query-interval interval
    }
  }
}
```

Use this command to set the interval between IGMP group-specific or source specific Query messages sent by the router to determine whether any receivers remain in a multicast group.

The router uses this Query to determine whether any members of the multicast group remain on the network. If it receives no response to the configured number of queries (as set in [interfaces <interface> ip igmp last-member-query-count <count> \(page 17\)](#)), the router stops forwarding to that network.

Use the `set` form of this command to set the interval for last-member queries.

Use the `delete` form of this command to restore the default interval for last-member queries.



Use the `show` form of this command to show last-member query interval configuration.

---

## interfaces <interface> ip igmp limit <limit>

Sets the limit for IGMP group memberships on an interface.

**Syntax:**

```
set interfaces interface ip igmp limit limit
```

**Syntax:**

```
delete interfaces interface ip igmp limit limit
```

**Syntax:**

```
show interfaces interface ip igmp limit
```

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp \(page 12\)](#).

**limit**

The maximum number of multicast group memberships that can be defined for the network served by the interface. The range is 1 to 2097152. By default, a limit of 5000 is applied.

**Configuration mode**

```
interfaces interface {
  ip {
    igmp {
      limit limit
    }
  }
}
```

Use this command to set an interface-specific limit on the number of multicast group memberships to be served by an interface.

When this option is in effect and the maximum is reached, the router ignores all further local requests for membership.

Use the `set` form of this command to set the limit on multicast group memberships on an interface.

Use the `delete` form of this command to restore the default behavior for multicast group membership limits.

Use the `show` form of this command to show between to and static group membership limit configuration.

---

## interfaces <interface> ip igmp limit-exception <acl>

Specifies multicast groups unaffected by the IGMP group membership limits on an interface.

**Syntax:**

```
set interfaces interface ip igmp limit-exception acl
```

**Syntax:**

```
delete interfaces interface ip igmp limit-exception acl
```

**Syntax:**

```
show interfaces interface ip igmp limit-exception
```

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp \(page 12\)](#).

**acl**



An access list number used to define the membership group. Supported ranges of values are:

1 to 99: IP access list number.

1300 to 1999: IP access list number in the expanded range.

Access control lists are a type of routing policy; see the AT&T Vyatta Network Operating System Routing Policies Configuration Guide for information on creating them.

### Configuration mode

```
interfaces interface {
  ip {
    igmp {
      limit-exception acl
    }
  }
}
```

Use this command to specify the multicast groups that are an exception to the membership limits imposed by `interfaces <interface> ip igmp limit <limit>` ([page 19](#)). As such, this command is dependent on `interfaces <interface> ip igmp limit <limit>` ([page 19](#)) being set.

Use the `set` form of this command to specify the multicast groups that are unaffected by the IGMP group membership limits on an interface.

Use the `delete` form of this command to remove the list of multicast groups that are unaffected by the IGMP group membership limits on an interface.

Use the `show` form of this command to show group membership limit exception configuration.

---

## interfaces <interface> ip igmp offlink

Allows multicast transmissions to be forwarded off-link.

### Syntax:

```
set interfaces interface ip igmp offlink
```

### Syntax:

```
delete interfaces interface ip igmp offlink
```

### Syntax:

```
show interfaces interface ip igmp offlink
```

### *interface*

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp](#) ([page 12](#)).

### Configuration mode

```
interfaces interface {
  ip {
    igmp {
      offlink
    }
  }
}
```

Use this command to enable IGMP off-link on the system.

Use the `set` form of this command to set IGMP off-link.

Use the `delete` form of this command to delete IGMP off-link.

Use the `show` form of this command to show IGMP interface configuration.



---

## interfaces <interface> ip igmp querier-timeout <interval>

Sets the interval before the system takes over as querier on an interface.

**Syntax:**

```
set interfaces interface ip igmp querier-timeout interval
```

**Syntax:**

```
delete interfaces interface ip igmp querier-timeout
```

**Syntax:**

```
show interfaces interface ip igmp querier-timeout
```

The router waits to receive a query for 255 seconds before taking over as querier.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp \(page 12\)](#).

**interval**

The amount of time, in seconds, the router waits before taking over as querier when the previous querier fails to send an IGMP Query. The range is 60 to 300. The default is 255.

**Configuration mode**

```
interfaces interface {
  ip {
    igmp {
      querier-timeout interval
    }
  }
}
```

Use this command to specify how long the router waits to receive an IGMP query from the previous querier. When this interval expires, the router takes over as querier.

Use the `set` form of this command to set the querier timeout interval.

Use the `delete` form of this command to restore the default querier timeout interval.

Use the `show` form of this command to show querier timeout interval configuration.

---

## interfaces <interface> ip igmp query-interval <interval>

Specifies the frequency at which IGMP host queries are sent.

**Syntax:**

```
set interfaces interface ip igmp query-interval interval
```

**Syntax:**

```
delete interfaces interface ip igmp query-interval
```

**Syntax:**

```
show interfaces interface ip igmp query-interval
```

The router sends IGMP Query messages at intervals of 125 seconds.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp \(page 12\)](#).

**interval**



The interval, in seconds, at which the router sends IGMP Query messages. The range is 2 to 18000. The default is 125.

### Configuration mode

```
interfaces interface {
  ip {
    igmp {
      query-interval interval
    }
  }
}
```

Use this command to set the frequency with which the router sends IGMP host Query messages.

**Note:** The interval for `query-interval` must be greater than the interval for `query-max-response-time` used in `interfaces <interface> ip igmp query-max-response-time <interval>` ([page 22](#)).

Use the `set` form of this command to set the query interval.

Use the `delete` form of this command to restore the default query interval.

Use the `show` form of this command to show query interval configuration.

---

## interfaces <interface> ip igmp query-max-response-time <interval>

Specifies the maximum response time advertised in IGMP queries.

### Syntax:

```
set interfaces interface ip igmp query-max-response-time interval
```

### Syntax:

```
delete interfaces interface ip igmp query-max-response-time
```

### Syntax:

```
show interfaces interface ip igmp query-max-response-time
```

The router waits 10 seconds for a response to an IGMP Query before deleting the multicast group.

### interface

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp](#) ([page 12](#)).

### interval

The amount of time, in seconds, that the router advertises as the maximum delay before a responder can respond to an IGMP Query. The range is 1 to 240. The default is 20.

### Configuration mode

```
interfaces interface {
  ip {
    igmp {
      query-max-response-time interval
    }
  }
}
```

Use this command to set the value to be advertised as the maximum time the router will wait to receive a response to IGMP Query messages. When this interval expires, the router deletes the multicast group.

Use the `set` form of this command to set the maximum query response time.



Use the `delete` form of this command to restore the default maximum query response time.

Use the `show` form of this command to show maximum query response time configuration.

---

## interfaces <interface> ip igmp robustness-variable <variable>

Specifies the value of the robustness variable on an interface.

**Syntax:**

```
set interfaces interface ip igmp robustness-variable variable
```

**Syntax:**

```
delete interfaces interface ip igmp robustness-variable
```

**Syntax:**

```
show interfaces interface ip igmp robustness-variable
```

The robustness variable is set to 2.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp \(page 12\)](#).

**variable**

The value for the robustness variable. The range is 2 to 7. The default is 2.

**Configuration mode**

```
interfaces interface {
  ip {
    igmp {
      robustness-variable variable
    }
  }
}
```

Use this command to set the robustness variable for an interface.

The robustness variable specifies how many IGMP refresh packets for a given state can be lost before the system times out and changes state. This helps tune the network for expected packet loss.

Use the `set` form of this command to set the robustness variable value.

Use the `delete` form of this command to restore the default robustness variable value.

Use the `show` form of this command to show robustness variable configuration.

---

## interfaces <interface> ip igmp startup-query-count <count>

Specifies the number of IGMP Query messages to be sent on startup for an interface.

**Syntax:**

```
set interfaces interface ip igmp startup-query-count count
```

**Syntax:**

```
delete interfaces interface ip igmp startup-query-count
```

**Syntax:**

```
show interfaces interface ip igmp startup-query-count
```

Two IGMP Query messages are sent when the router starts up.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp](#) (page 12).

**count**

The number of IGMP Query messages to be sent when the router starts up. The range is 2 to 10. The default is 2.

**Configuration mode**

```
interfaces interface {
  ip {
    igmp {
      startup-query-count count
    }
  }
}
```

Use this command to specify the number of IGMP Query messages to be sent when the router starts up.

Use the `set` form of this command to set the query startup count.

Use the `delete` form of this command to restore the default value for query startup count.

Use the `show` form of this command to show query startup count configuration.

---

## interfaces <interface> ip igmp startup-query-interval <interval>

Sets the interval at which IGMP Query messages will be sent on startup for an interface.

**Syntax:**

```
set interfaces interface ip igmp startup-query-interval interval
```

**Syntax:**

```
delete interfaces interface ip igmp startup-query-interval
```

**Syntax:**

```
show interfaces interface ip igmp startup-query-interval
```

At startup, Query messages are sent at 31-second intervals.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp](#) (page 12).

**interval**

The interval, in seconds, between IGMP Query messages sent when the router starts up. The range is 1 to 18000. The default is 31.

**Configuration mode**

```
interfaces interface {
  ip {
    igmp {
      startup-query-interval interval
    }
  }
}
```

Use this command to specify the interval at which IGMP query messages are sent when the router starts.

Use the `set` form of this command to set the query startup interval.





Use the `delete` form of this command to restore the default query startup interval.

Use the `show` form of this command to show query startup interval configuration.

---

## interfaces <interface> ip igmp version <version>

Sets the IGMP version in use on an interface.

**Syntax:**

```
set interfaces interface ip igmp version version
```

**Syntax:**

```
delete interfaces interface ip igmp version
```

**Syntax:**

```
show interfaces interface ip igmp version
```

IGMPv3 is used on the router.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp \(page 12\)](#).

**version**

The IGMP version number. Supported values are 1, 2, and 3. The default is 3.

**Configuration mode**

```
interfaces interface {  
  ip {  
    igmp {  
      version version  
    }  
  }  
}
```

Use this command to specify which IGMP version the router should use for multicast routing.

Use the `set` form of this command to set the IGMP version number.

Use the `delete` form of this command to restore the default IGMP version number.

Use the `show` form of this command to show IGMP version number configuration.

---

## interfaces <interface> ip igmp static-group <group> source <source>

Specifies static multicast group membership on an interface.

**Syntax:**

```
set interfaces interface ip igmp static-group group source source
```

**Syntax:**

```
delete interfaces interface ip igmp static-group group source source
```

**Syntax:**

```
show interfaces interface ip igmp static-group group source source
```

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ip igmp \(page 12\)](#).

**group**



The IP multicast address of the group the router is being made a member of.

**source**

The static source of multicast packets. Supported values are:

**x.x.x.x**: The IP address of a multicast source.

**ssm-map**: Use Source Specific Multicast (SSM) mapping to determine the multicast source associated with this group.

**Configuration mode**

```
interfaces interface {
  ip {
    igmp {
      static-group group
      {
        source source
      }
    }
  }
}
```

Use this command to specify static multicast group membership on an interface.

When the multicast group is statically configured on an interface, packets to the group are fast-switched out the interface ensuring all upstream routers maintain routing information for the group.

When the **ssm-map** keyword is used, the router uses Source Specific Multicast (SSM) mapping to determine the multicast source associated with this group. The resulting (S, G) channels are statically forwarded.

Use the **set** form of this command to specify static multicast group membership on an interface.

Use the **delete** form of this command to remove multicast group membership on the interface.

Use the **show** form of this command to show static group membership configuration.

---

## monitor protocol multicast

Enables IGMP debugging.

**Syntax:**

```
monitor protocol multicast [ enable | disable ] igmp
```

IGMP debugging is disabled.

**enable**

Enables the specified debugging option.

**disable**

Disables the specified debugging option.

**igmp**

Specifies debugging of IGMP.

**Operational mode**

Use this command to enable debugging for IGMP.

When enabled, debugging messages are generated for all interfaces running the IGMP protocol.

The following example starts IGMP debugging.

```
vyatta@vyatta:~$monitor protocol multicast enable igmp
```



---

## protocols igmp limit <limit>

Sets a global limit on the number of IGMP groups.

**Syntax:**

```
set protocols igmp limit limit
```

**Syntax:**

```
delete protocols igmp limit limit
```

**Syntax:**

```
show protocols igmp limit
```

**limit**

The maximum number of IGMP multicast groups. The number ranges from 1 through 2097152. By default, no limit is applied.

**Configuration mode**

```
protocols {  
  igmp {  
    limit limit  
  }  
}
```

Use this command to set a global limit on the number of multicast groups. When the limit is reached, the router ignores all further local requests for membership.

Use the `set` form of this command to set a limit on the number of multicast groups.

Use the `delete` form of this command to remove the limit applied on the number of multicast groups.

Use the `show` form of this command to display the limit on the number of multicast groups.

---

## protocols igmp log

Enables IGMP logs.

**Syntax:**

```
set protocols igmp log{ all | decode | encode | events | fsm | tib }
```

**Syntax:**

```
delete protocols igmp log{ all | decode | encode | events | fsm | tib }
```

**Syntax:**

```
show protocols igmp log{ all | decode | encode | events | fsm | tib }
```

None

**all**

Enables all IGMP logs.

**decode**

Enables only IGMP decode logs.

**encode**

Enables only IGMP encode logs.

**events**

Enables only IGMP event logs.

**fsm**

Enables only IGMP finite-state machine (FSM) logs.

**tib**



Enables only IGMP tree-information-base (TIB) logs.

### Configuration mode

```
protocols {
  igmp {
    log {
      all
      decode
      encode
      fsm
      tib
    }
  }
}
```

Use the `set` form of this command to enable Internet Group Management Protocol (IGMP) logs.

Use the `delete` form of this command to disable IGMP logs.

Use the `show` form of this command to view IGMP logging configuration.

---

## protocols igmp ssm-map

Enables source-specific multicast mapping globally.

### Syntax:

```
set protocols igmp ssm-map
```

### Syntax:

```
delete protocols igmp ssm-map
```

### Syntax:

```
show protocols ip igmp
```

SSM mapping is disabled.

### Configuration mode

```
protocols {
  igmp {
    ssm-map
  }
}
```

Use this command to globally enable source-specific multicast (SSM) mapping for groups in a configured SSM range. The range is configured globally using `protocols igmp ssm-map static access-list <acl> source <source>` ([page 28](#)).

A value set at the interface level overrides the global value.

Use the `set` form of this command to enable SSM mapping.

Use the `delete` form of this command to restore the default behavior for SSM mapping.

Use the `show` form of this command to show SSM mapping configuration.

---

## protocols igmp ssm-map static access-list <acl> source <source>

Globally associates a multicast source for static SSM map group.

### Syntax:



```
set protocols igmp ssm-map static access-list acl source source
```

**Syntax:**

```
delete protocols igmp ssm-map static access-list acl source source
```

**Syntax:**

```
show protocols igmp ssm-map static access-list acl
```

***acl***

The name of an IPv4 access control list to be used for filtering membership groups. Supported ranges of values are:

1 to 99: IP access list number.

1300 to 1999: IP access list number in the expanded range.

Access control lists are a type of routing policy; see the AT&T Vyatta Network Operating System Routing Policies Configuration Guide for information on creating them.

***source***

The source address to associate with SSM mapping. The format is an IPv4 address in dotted quad format.

**Configuration mode**

```
protocols {
  igmp {
    ssm-map {
      static {
        access-list acl{
          source source
        }
      }
    }
  }
}
```

Use this command to globally define a group for static SSM mapping.

A value set at the interface level overrides the global value.

This command statically assigns source values to IGMPv1 and IGMPv2 groups to translate the sources represented with the wildcard in (\*,G) entries to (S, G) entries.

Use the `set` form of this command to associate the specified group with SSM mapping.

Use the `delete` form of this command to delete the SSM mapping association.

Use the `show` form of this command to show SSM group association configuration.

---

## reset ip igmp

Clears the specified IGMP local memberships.

**Syntax:**

```
reset ip igmp [ group group [ interface ] | interface interface ]
```

***group***

Clears the specified multicast group and deletes IGMP group cache entries. The format is an IPv4 multicast address.

***interface***

Clears the specified multicast group learned from the specified interface. The format is an interface type, as described in [interfaces <interface> ip igmp \(page 12\)](#).

***interface interface***



Clears all multicast groups learned from the specified interface. The format is an interface type, as described in [interfaces <interface> ip igmp \(page 12\)](#).

### Operational mode

Use this command to clear IGMP group membership information.

The following example clears group membership information for the multicast group 224.1.1.1.

```
vyatta@vyatta:~$reset ip igmp group 224.1.1.1
```

The following example clears group membership information for interface dp0p1p2.

```
vyatta@vyatta:~$reset ip igmp interface dp0p1p2
```

---

## show ip igmp groups

Displays the multicast groups with receivers connected to the system and learned through IGMP.

### Syntax:

```
show ip igmp groups [ [ group-address group [ detail ] | interface interface [ group [ detail ] ] [ detail ] | detail ] ]
```

When used with no option, displays all available group information in summary format.

### *group*

Shows multicast group information for the specified IPv4 multicast group.

### *interface*

Shows multicast group information for the specified interface. For a list of supported interfaces, see [interfaces <interface> ip igmp \(page 12\)](#).

### *detail*

Provides detailed group information.

### Operational mode

Use this command to display the multicast groups with receivers connected to the system and learned through IGMP.

The following example shows group membership information.

```
vyatta@vyatta:~$show ip igmp groups
IGMP Connected Group Membership
Group Address  Interface  Uptime    Expires   Last Reporter
225.0.0.2     dp0s4     00:00:06 00:04:16 10.0.3.6
225.0.0.3     dp0s4     00:00:06 00:04:16 10.0.3.6
225.0.0.4     dp0s4     00:00:06 00:04:16 10.0.3.6
```

---

## show ip igmp interface

Displays the operational state of IGMP on an interface.

### Syntax:

```
show ip igmp interface [ interface ]
```



When used with no option, this command displays the operational state of all IGMP-enabled interfaces.

### **interface**

Displays the operational state of the specified interface.

### **Operational mode**

Use this command to display the state of IGMP on interfaces.

The following example shows IGMP interface information for interface dp0p1p2.

```
vyatta@vyatta:~$show ip igmp interface dp0p1p2
Interface dp0p1s1 (Index 9)
  IGMP Enabled, Active, Forced Querier, Configured for version 3
  Internet address is 10.10.1.2
  IGMP interface limit is 5000
  IGMP interface has 0 group-record states
  IGMP activity: 0 joins, 0 leaves
  IGMP query interval is 126 seconds
  IGMP Startup query interval is 31 seconds
  IGMP Startup query count is 2
  IGMP querier timeout is 257 seconds
  IGMP max query response time is 10 seconds
  Group Membership interval is 262 seconds
  IGMP Last member query count is 2
  Last member query response interval is 1000 milliseconds
```

---

## **show ip igmp ssm-map**

Displays information about IGMP SSM-mapping.

### **Syntax:**

```
show ip igmp ssm-map [ group ]
```

When used with no option, this command displays all SSM-mapping information.

### **group**

Displays SSM mapping information for the specified group. The format is an IP address of an IPv4 multicast group.

### **Operational mode**

Use this command to display information about SSM mapping.

The following example shows IGMP SSM mapping information for multicast group 232.10.10.10.

```
vyatta@vyatta:~$show ip igmp ssm-map 232.10.10.10
Group address: 232.10.10.10
Database      : Static
Source list   : 10.10.10.10
```

---

## **show monitoring protocols multicast igmp**

Displays IGMP debugging status.

### **Syntax:**

```
show monitoring protocols multicast igmp
```

### **Operational mode**



Use this command to show the status of IGMP debugging.

The following example shows the status of IGMP debugging.

```
vyatta@vyatta:~$show monitoring protocols multicast igmp
IGMP Debugging status:
IGMP Decoder debugging is on
IGMP Encoder debugging is on
IGMP Events debugging is on
IGMP FSM debugging is on
IGMP Tree-Info-Base (TIB) debugging is on
```





# MLD Commands

## interfaces <interface> ipv6 mld

Enables MLD on an interface.

**Syntax:**

```
set interfaces interface ipv6 mld
```

**Syntax:**

```
delete interfaces interface ipv6 mld
```

**Syntax:**

```
show interfaces interface ipv6 mld
```

**interface**

The type of interface. For detailed keywords and arguments for interfaces that support multicast routing, see the table in the Usage Guidelines below.

**Configuration mode**

```
interfaces interface {  
  ipv6 {  
    mld {  
    }  
  }  
}
```

Use this command to enable Multicast Listener Discovery (MLD) on an interface.

This command enables MLD operation in stand-alone mode, and can be used to learn local membership information prior to enabling a multicast routing protocol on the interface.

**Note:** Enabling IP on an interface enables the host-side functionality of MLD by default. The `set interfaces interface ipv6 mld` command enables the router-side functionality of the MLD on the given interface.

**Note:** To use MLD for multicast routing, multicast routing must be enabled on the router. For information about multicast routing in general, see the AT&T Vyatta Network Operating System Multicast Routing Configuration Guide.

The following table shows the syntax and parameters for interface types. Some of these types may not apply to this command.

Interface Type	Syntax	Parameters
Bridge	bridge <i>brx</i>	<i>brx</i> : The name of a bridge group. The name ranges from br0 through br999.



Interface Type	Syntax	Parameters
Data plane	<code>dataplane dpxy&gt;pzv</code>	<p>dpxypyz: The name of a data plane interface to which the following applies:</p> <ul style="list-style-type: none"><li>dpx identifies a data plane ID</li></ul> <p><b>Note:</b> Currently, only dp0 is supported.</p> <ul style="list-style-type: none"><li>py specifies a physical or virtual PCI slot index</li><li>pz specifies a port index</li></ul> <p>Other supported name formats are the following:</p> <ul style="list-style-type: none"><li>dpxemy—used for LAN-on-motherboard (LOM) devices that do not have a PCI slot. emy specifies an embedded network interface number.</li><li>dpxporty—used for devices in which the PCI slot cannot be identified. porty specifies a port index.</li></ul>
Data plane vif	<code>dataplane dpxypyz vif <i>vif-id</i> [vlan <i>vlan-id</i> ]</code>	<p>dpxypyz: The name of a data plane interface to which the following applies:</p> <ul style="list-style-type: none"><li>dpx specifies a data plane ID</li></ul> <p><b>Note:</b> Currently, only dp0 is supported.</p> <ul style="list-style-type: none"><li>py specifies a physical or virtual PCI slot index</li><li>pz specifies a port index</li></ul> <p>Other supported name formats are the following:</p> <ul style="list-style-type: none"><li>dpxemy—used for LAN-on-motherboard (LOM) devices that do not have a PCI slot. emy specifies an embedded network interface number.</li><li>dpxporty—used for devices in which the PCI slot cannot be identified. porty specifies a port index.</li></ul> <p><i>vif-id</i>: A virtual interface ID. The ID ranges from 1 through 4094.</p> <p><i>vlan-id</i>: The VLAN ID of a virtual interface. The ID ranges from 1 through 4094.</p>



Interface Type	Syntax	Parameters
Loopback	loopback <i>lo</i>	<i>lo</i> : The name of a loopback interface.
OpenVPN	openvpn <i>vtunx</i>	<i>vtunx</i> : The identifier of an OpenVPN interface. The identifier ranges from vtun0 through vtunx, where x is a nonnegative integer.
Tunnel	tunnel <i>tunx</i> or tunnel <i>tunx</i> parameters	<i>tunx</i> : The identifier of a tunnel interface you are defining. The identifier ranges from tun0 through tunx, where x is a nonnegative integer.
Virtual tunnel	vti <i>vtix</i>	<i>vtix</i> : The identifier of a virtual tunnel interface you are defining. The identifier ranges from vti0 through vtix, where x is a nonnegative integer.  <b>Note:</b> This interface does not support IPv6.
VRRP	interface <i>parent-if</i> vrrp vrrp-group <i>group</i> interface	<i>parent-if</i> : The type and identifier of a parent interface; for example, dataplane dp0p1p2 or bridge br999.  <i>group</i> : A VRRP group identifier.  The name of a VRRP interface is not specified. The system internally constructs the interface name from the parent interface identifier plus the VRRP group number; for example, dp0p1p2v99. Note that VRRP interfaces support the same feature set as the parent interface does.

Use the `set` form of this command to enable MLD on an interface.

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

Use the `show` form of this command to display MLD configuration.

---

## interfaces <interface> ipv6 mld access-group <acl6-name>

Controls the multicast local membership groups learned on an interface.

### Syntax:

```
set interfaces interface ipv6 mld access-group acl6-name
```

### Syntax:

```
delete interfaces interface ipv6 mld access-group
```

### Syntax:



```
show interfaces interface ipv6 mld access-group
```

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ipv6 mld \(page 33\)](#).

**acl6-name**

The name of an IPv6 access control list to be used for filtering membership groups. Access control lists are a type of routing policy; see the AT&T Vyatta Network Operating System Routing Policies Configuration Guide for information on creating them.

**Configuration mode**

```
interfaces interface {
  ipv6 {
    mld {
      access-group acl6-name
    }
  }
}
```

Use this command to use an access control list to control multicast local membership groups learned on an interface.

This command applies to interfaces configured for MLD Layer 3 multicast protocols.

Use the `set` form of this command to apply the access control list.

Use the `delete` form of this command to stop access-list filtering for MLD.

Use the `show` form of this command to show access-list configuration for MLD.

---

## interfaces <interface> ipv6 mld immediate-leave group-list <acl6-name>

Minimizes latency for hosts leaving multicast groups.

**Syntax:**

```
set interfaces interface ipv6 mld immediate-leave group-list acl6-name
```

**Syntax:**

```
delete interfaces interface ipv6 mld immediate-leave group-list
```

**Syntax:**

```
show interfaces interface ipv6 mld immediate-leave group-list
```

Immediate leave is disabled.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ipv6 mld \(page 33\)](#).

**acl6-name**

The name of an IPv6 access control list to be used to define the membership group. Access control lists are a type of routing policy; see the AT&T Vyatta Network Operating System Routing Policies Configuration Guide for information on creating them.

**Configuration mode**

```
interfaces interface {
  ipv6 {
    mld {
      immediate-leave {
        group-list acl6-name
      }
    }
  }
}
```



```
    }  
  }  
}
```

Use this command to minimize the leave latency for MLD memberships.

When this option is not set, the router sends a Query message when a receiver host has sent a Leave message. At this point, a timeout interval goes into effect. When this option is set, the Leave message is acted on immediately, without sending the Query or waiting for the timeout period to expire.

This command applies to interfaces configured for MLD Layer 3 multicast protocols. It applies when only one receiver host is connected to each interface.

Use the `set` form of this command to enable MLD immediate leave.

Use the `delete` form of this command to restore the MLD immediate leave default behavior.

Use the `show` form of this command to view MLD immediate leave configuration.

---

## interfaces <interface> ipv6 mld last-member-query-count <count>

Manually sets the last member query count value.

### Syntax:

```
set interfaces interface ipv6 mld last-member-query-count count
```

### Syntax:

```
delete interfaces interface ipv6 mld last-member-query-count
```

### Syntax:

```
show interfaces interface ipv6 mld last-member-query-count
```

The router sends two Query messages, after which it considers the host to have left the group.

### *interface*

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ipv6 mld \(page 33\)](#).

### *count*

The number of times the router sends a Query message after receiving a Leave message. The range is 2 to 7. The default is 2.

### Configuration mode

```
interfaces interface {  
  ipv6 {  
    mld {  
      last-member-query-count count  
    }  
  }  
}
```

Use this command to set the number of times that the router sends a group-specific or source-group specific Query message when it receives a Leave message from a receiver host.

This command applies to interfaces configured for MLD Layer 3 multicast protocols.

The router uses this Query to determine whether any members of the multicast group remain on the network. The command is sent at the interval set in [interfaces <interface> ipv6 mld last-member-query-interval <interval> \(page 38\)](#). If the router receives no response to the configured number of queries, the router stops forwarding to that network.

Use the `set` form of this command to set the number of last-member queries sent.



Use the `delete` form of this command to restore the default value for last-member queries.

Use the `show` form of this command to show last-member query configuration.

---

## interfaces <interface> ipv6 mld last-member-query-interval <interval>

Specifies the frequency at which MLD group-specific host queries are sent.

**Syntax:**

```
set interfaces interface ipv6 mld last-member-query-interval interval
```

**Syntax:**

```
delete interfaces interface ipv6 mld last-member-query-interval
```

**Syntax:**

```
show interfaces interface ipv6 mld last-member-query-interval
```

The router waits 1000 milliseconds between last-member queries.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ipv6 mld](#) (page 33).

**interval**

The interval between last-member queries, in milliseconds. The range is 1000 to 25500. The default is 1000.

**Configuration mode**

```
interfaces interface {
  ipv6 {
    mld {
      last-member-query-interval interval
    }
  }
}
```

Use this command to set the interval between MLD group-specific or source-specific Query messages sent by the router to determine whether any receivers remain in a multicast group.

The router uses this Query to determine whether any members of the multicast group remain on the network. If it receives no response to the configured number of queries (as set in [interfaces <interface> ipv6 mld last-member-query-count <count>](#) (page 37)), the router stops forwarding to that network.

This command applies to interfaces configured for MLD Layer 3 multicast protocols.

Use the `set` form of this command to set the interval for last-member queries.

Use the `delete` form of this command to restore the default interval for last-member queries.

Use the `show` form of this command to show last-member query interval configuration.

---

## interfaces <interface> ipv6 mld limit <limit>

Sets the limit for MLD group memberships on an interface.

**Syntax:**

```
set interfaces interface ipv6 mld limit limit
```

**Syntax:**

```
delete interfaces interface ipv6 mld limit
```

**Syntax:**

```
show interfaces interface ipv6 mld limit
```

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ipv6 mld \(page 33\)](#).

**limit**

The maximum number of multicast group memberships that can be defined for the network served by the interface. The range is 1 to 2097152. By default, a limit of 5000 is applied.

**Configuration mode**

```
interfaces interface {  
    ipv6 {  
        mld {  
            limit limit  
        }  
    }  
}
```

Use this command to set an interface-specific limit on the number of multicast group memberships to be served by an interface.

This command applies to interfaces configured for MLD Layer 3 multicast protocols.

When this option is in effect and the maximum is reached, the router ignores all further local requests for membership.

Use the `set` form of this command to set the limit on multicast group memberships on an interface.

Use the `delete` form of this command to restore the default behavior for multicast group membership limits.

Use the `show` form of this command to show group membership limit configuration.

---

**interfaces <interface> ipv6 mld limit-exception <acl6-name>**

Specifies multicast groups unaffected by the MLD group membership limits on an interface.

**Syntax:**

```
set interfaces interface ipv6 mld limit-exception acl6-name
```

**Syntax:**

```
delete interfaces interface ipv6 mld limit-exception acl6-name
```

**Syntax:**

```
show interfaces interface ipv6 mld limit-exception
```

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ipv6 mld \(page 33\)](#).

**acl6-name**

The name of an IPv6 access control list to be used to define the membership groups. Access control lists are a type of routing policy; see the AT&T Vyatta Network Operating System Routing Policies Configuration Guide for information on creating them.

**Configuration mode**

```
interfaces interface {  
    ipv6 {  
        mld {  
            limit-exception acl6-name  
        }  
    }  
}
```



```
}  
}
```

Use this command to specify the multicast groups that are an exception to the membership limits imposed by `interfaces <interface> ipv6 mld limit <limit>` (page 38). As such, this command is dependent on `interfaces <interface> ipv6 mld limit <limit>` (page 38) being set.

Use the `set` form of this command to specify the multicast groups that are unaffected by the MLD group membership limits on an interface.

Use the `delete` form of this command to remove the list of multicast groups that are unaffected by the MLD group membership limits on an interface.

Use the `show` form of this command to show group membership limit exception configuration.

---

## **interfaces <interface> ipv6 mld querier-timeout <interval>**

Sets the interval before the system takes over as querier on an interface.

### **Syntax:**

```
set interfaces interface ipv6 mld querier-timeout interval
```

### **Syntax:**

```
delete interfaces interface ipv6 mld querier-timeout
```

### **Syntax:**

```
show interfaces interface ipv6 mld querier-timeout
```

The router waits to receive a query for 255 seconds before taking over as querier.

### **interface**

The type of interface. For a list of supported interfaces and detailed syntax, see `interfaces <interface> ipv6 mld` (page 33).

### **interval**

The amount of time, in seconds, the router waits before taking over as querier when the previous querier fails to send an MLD Query. The range is 60 to 300. The default is 255.

### **Configuration mode**

```
interfaces interface {  
  ipv6 {  
    mld {  
      querier-timeout interval  
    }  
  }  
}
```

Use this command to specify how long the router waits to receive an MLD query from the previous querier. When this interval expires, the router takes over as querier.

This command applies to interfaces configured for MLD Layer 3 multicast protocols.

Use the `set` form of this command to set the querier timeout interval.

Use the `delete` form of this command to restore the default querier timeout interval.

Use the `show` form of this command to show querier timeout interval configuration.

---

## **interfaces <interface> ipv6 mld query-interval <interval>**

Specifies the frequency at which MLD host queries are sent.

### **Syntax:**

```
set interfaces interface ipv6 mld query-interval interval
```



**Syntax:**

delete interfaces *interface* **ipv6 mld query-interval**

**Syntax:**

show interfaces *interface* **ipv6 mld query-interval**

The router sends MLD Query messages at intervals of 125 seconds.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ipv6 mld \(page 33\)](#).

**interval**

The interval, in seconds, at which the router sends MLD Query messages. The range is 1 to 18000. The default is 125.

**Configuration mode**

```
interfaces interface {
  ipv6 {
    mld {
      query-interval interval
    }
  }
}
```

Use this command to set the frequency with which the router sends MLD host Query messages.

This command applies to interfaces configured for MLD Layer 3 multicast protocols.

Use the set form of this command to set the query interval.

Use the delete form of this command to restore the default query interval.

Use the show form of this command to show query interval configuration.

---

## **interfaces <interface> ipv6 mld query-max-response-time <interval>**

Specifies the maximum response time advertised in MLD queries.

**Syntax:**

set interfaces *interface* **ipv6 mld query-max-response-time interval**

**Syntax:**

delete interfaces *interface* **ipv6 mld query-max-response-time**

**Syntax:**

show interfaces *interface* **ipv6 mld query-max-response-time**

The router waits 10 seconds for a response to an MLD Query before deleting the multicast group.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ipv6 mld \(page 33\)](#).

**interval**

The amount of time, in seconds, that the router advertises as the maximum delay before a responder can respond to an MLD Query. The range is 1 to 240.

**Configuration mode**

```
interfaces interface {
```



```
ipv6 {
  mld {
    query-max-response-time interval
  }
}
```

Use this command to set the value to be advertised as the maximum time the router will wait to receive a response to MLD Query messages. When this interval expires, the router deletes the multicast group.

This command applies to interfaces configured for MLD Layer 3 multicast protocols.

Use the `set` form of this command to set the maximum query response time.

Use the `delete` form of this command to restore the default maximum query response time.

Use the `show` form of this command to show maximum query response time configuration.

---

## interfaces <interface> ipv6 mld robustness-variable <variable>

Specifies the value of the robustness variable on an interface.

### Syntax:

```
set interfaces interface ipv6 mld robustness-variable variable
```

### Syntax:

```
delete interfaces interface ipv6 mld robustness-variable
```

### Syntax:

```
show interfaces interface ipv6 mld robustness-variable
```

The robustness variable is set to 2.

### *interface*

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ipv6 mld \(page 33\)](#).

### *variable*

The robustness variable. The range is 2 to 7.

### Configuration mode

```
interfaces interface {
  ipv6 {
    mld {
      robustness-variable variable
    }
  }
}
```

Use this command to set the robustness variable for an interface.

This command applies to interfaces configured for MLD Layer 3 multicast protocols.

The robustness variable specifies how many MLD refresh packets for a given state can be lost before the system times out and changes state. This helps tune the network for expected packet loss.

Use the `set` form of this command to set the robustness variable value.

Use the `delete` form of this command to restore the default robustness variable value.

Use the `show` form of this command to show robustness variable configuration.



---

## interfaces <interface> ipv6 mld static-group <group> source <source>

Specifies static multicast group membership on an interface.

**Syntax:**

set interfaces *interface* ipv6 mld static-group *group* source *source*

**Syntax:**

delete interfaces *interface* ipv6 mld static-group *group* source *source*

**Syntax:**

show interfaces *interface* ipv6 mld static-group *group* source

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ipv6 mld \(page 33\)](#).

**group**

The IPv6 multicast address of the group the router is being made a member of.

**source**

The static source of multicast packets. Supported values are:

*x:x:x:x:x:x*: The IPv6 address of a multicast source.

**ssm-map**: Use Source Specific Multicast (SSM) mapping to determine the multicast source associated with this group.

**Configuration mode**

```
interfaces interface {
  ipv6 {
    mld {
      static-group group
      {
        source source
      }
    }
  }
}
```

Use this command to add MLD group or source records to a specific interface.

When the router is statically configured to be a member of a group, packets to the group are fast-switched out the interface ensuring all upstream routers maintain routing information for the group.

When the **ssm-map** keyword is used, the router uses Source Specific Multicast (SSM) mapping to determine the multicast source associated with this group. The resulting (S, G) channels are statically forwarded.

Use the **set** form of this command to specify static multicast group membership on an interface.

Use the **delete** form of this command to remove multicast group membership on the interface.

Use the **show** form of this command to static group membership configuration.

---

## interfaces <interface> ipv6 mld version <version>

Sets the MLD version in use on an interface.

**Syntax:**

set interfaces *interface* ipv6 mld version *version*

**Syntax:**



delete interfaces *interface* **ipv6 mld version**

**Syntax:**

show interfaces *interface* **ipv6 mld version**

MLDv2 is used.

**interface**

The type of interface. For a list of supported interfaces and detailed syntax, see [interfaces <interface> ipv6 mld \(page 33\)](#).

**version**

The MLD version number. Supported values are 1 and 2. The default is 2.

**Configuration mode**

```
interfaces interface {
  ipv6 {
    mld {
      version version
    }
  }
}
```

Use this command to specify which MLD version the router should use for multicast routing.

This command applies to interfaces configured for MLD Layer 3 multicast protocols.

Use the `set` form of this command to set the MLD version number.

Use the `delete` form of this command to restore the default MLD version number.

Use the `show` form of this command to show MLD version number configuration.

---

## monitor protocol multicast

Enables MLD debugging.

**Syntax:**

monitor protocol multicast { **enable** | **disable** } **mld**

MLD debugging is disabled.

**enable**

Enables the specified debugging option.

**disable**

Disables the specified debugging option.

**mld**

Specifies debugging of MLD.

**Operational mode**

Use this command to enable debugging for MLD.

When enabled, debugging messages are generated for all interfaces running the MLD protocol.

The following example starts MLD debugging.

```
vyatta@vyatta:~$monitor protocol multicast enable mld
```

---

## protocols mld limit <limit>

Sets the global limit for MLD group memberships.

**Syntax:**

```
set protocols mld limit limit
```

**Syntax:**

```
delete protocols mld limit
```

**Syntax:**

```
show protocols mld limit
```

No limit is applied.

***limit***

The maximum number of multicast groups that can be learned. The range is 1 to 2097152. By default, no limit is applied.

**Configuration mode**

```
protocols {  
  mld {  
    limit limit  
  }  
}
```

Use this command to set a global limit on the number of multicast groups that is learned.

When this option is in effect and the maximum is reached, the router ignores all further local requests for membership.

Use the `set` form of this command to set the limit on multicast groups.

Use the `delete` form of this command to restore the default behavior for multicast group limits.

Use the `show` form of this command to show group membership limit configuration.

---

## protocols mld log

Enables MLD logs.

**Syntax:**

```
set protocols mld log { all | decode | encode | events | fsm | tib }
```

**Syntax:**

```
delete protocols mld log { all | decode | encode | events | fsm | tib }
```

**Syntax:**

```
show protocols mld log { all | decode | encode | events | fsm | tib }
```

None

**all**

Enables all MLD logs.

**decode**

Enables only MLD decode logs.

**encode**

Enables only MLD encode logs.

**events**

Enables only MLD event logs.

**fsm**

Enables only MLD finite-state machine (FSM) logs.

**tib**

Enables only MLD thread-information-block (TIB) logs.



## Configuration mode

```
protocols {
  mld {
    log {
      all
      decode
      encode
      fsm
      tib
    }
  }
}
```

Use the `set` form of this command to enable Multicast Listener Discovery (MLD) logs.

Use the `delete` form of this command to remove MLD logs.

Use the `show` form of this command to view MLD logs.

---

## protocols mld ssm-map

Enables source-specific multicast mapping globally.

**Syntax:**

```
set protocols mld ssm-map
```

**Syntax:**

```
delete protocols mld ssm-map
```

**Syntax:**

```
show protocols mld ssm-map
```

SSM mapping is disabled.

**Configuration mode**

```
protocols {
  mld {
    ssm-map
  }
}
```

Use this command to globally enable source-specific multicast (SSM) mapping for groups in a configured SSM range. The range is configured globally using `protocols mld ssm-map static <acl6-name> source <source>` ([page 46](#)).

A value set at the interface level overrides the global value.

Use the `set` form of this command to enable SSM mapping.

Use the `delete` form of this command to restore the default behavior for SSM mapping.

Use the `show` form of this command to show SSM mapping configuration.

---

## protocols mld ssm-map static <acl6-name> source <source>

Globally associates a multicast source for static SSM map group.

**Syntax:**

```
set protocols mld ssm-map static acl6-name source source
```

**Syntax:**



```
delete protocols mld ssm-map static acl6-name source source
```

**Syntax:**

```
show protocols gmp ssm-map static acl6-name source
```

***acl6-name***

The name of an IPv6 access control list to be used for filtering membership groups. Access control lists are a type of routing policy; see the AT&T Vyatta Network Operating System Routing Policies Configuration Guide for information on creating them.

***source***

The IPv6 source address to associate with SSM mapping.

**Configuration mode**

```
protocols {
  mld {
    ssm-map {
      static acl6-name
      {
        source-address source
      }
    }
  }
}
```

Use this command to globally define a group for static SSM mapping.

A value set at the interface level overrides the global value.

This command statically assigns source values to multicast group entries to translate the sources represented with the wildcard in (\*,G) entries to (S, G) entries.

Use the set form of this command to associate the specified group with SSM mapping.

Use the delete form of this command to delete the SSM mapping association.

Use the show form of this command to show SSM group association configuration.

---

## reset ipv6 mld

Clears the specified MLD local memberships.

**Syntax:**

```
reset ipv6 mld { group group [ interface ] | interface interface }
```

***group***

Clears the specified multicast group and deletes MLD group cache entries. The format is an IPv6 multicast address.

***interface***

Clears the specified multicast group learned from the specified interface. The format is an interface type, as described in [interfaces <interface> ipv6 mld \(page 33\)](#).

**interface *interface***

Clears all multicast groups learned from the specified interface. The format is an interface type, as described in [interfaces <interface> ipv6 mld \(page 33\)](#).

**Operational mode**

Use this command to clear MLD group membership information.

The following example clears group membership information for the multicast group FF1E::10.

```
vyatta@vyatta:~$reset ipv6 mld group FF1E::10
```



The following example clears group membership information for interface dp0p1p2.

```
vyatta@vyatta:~$reset ipv6 mld group dp0p1p2
```

---

## show ipv6 mld groups

Displays the multicast groups with receivers connected to the system and learned through MLD.

### Syntax:

```
show ipv6 mld groups [ [ group-address group [ detail ] | interface interface [ detail ] | detail ] ]
```

When used with no option, displays all available group information in summary format.

### group

Shows multicast group information for the specified IPv6 multicast group.

### interface

Shows multicast group information for the specified interface. For a list of supported interfaces, see [interfaces <interface> ipv6 mld \(page 33\)](#).

### detail

Provides detailed group information for the specified interface.

### Operational mode

Use this command to display the multicast groups with receivers connected to the system and learned through MLD.

The following example shows MLD group membership information.

```
vyatta@vyatta:~$show ipv6 mld groups
MLD Connected Group Membership
Group Address          Interface          Uptime          Expires          Last Reporter
ffe::10                ge10              00:03:16 00:01:09      fe80::202:b3ff:fe1c:b
```

---

## show ipv6 mld interface

Displays the operational state of MLD on an interface.

### Syntax:

```
show ipv6 mld interface [ interface ]
```

When used with no option, this command displays the operational state of all MLD-enabled interfaces.

### interface

Displays the operational state of the specified interface.

### Operational mode

Use this command to display the state of MLD on interfaces.

The following example shows MLD interface information for interface dp0p1p2.

```
vyatta@vyatta:~$show ipv6 mld interface dp0p1p2
Interface dp0p1p2 (Index 2)
MLD Enabled, Active, Querier, Version 2 (default)
Internet address is fe80::2fd:6cff:fe1c:b
MLD interface has 0 group-record states
```





```
MLD activity: 0 joins, 0 leaves
MLD query interval is 125 seconds
MLD querier timeout is 255 seconds
MLD max query response time is 10 seconds
Last member query response interval is 1000 milliseconds
Group Membership interval is 260 seconds
```

---

## show ipv6 mld ssm-map

Displays information about MLD SSM-mapping.

### Syntax:

```
show ipv6 mld ssm-map [ group ]
```

When used with no option, this command displays all SSM-mapping information.

### *group*

Displays SSM mapping information for the specified group. The format is an IPv6 multicast address.

### Operational mode

Use this command to display information about SSM mapping.

---

## show monitoring protocols multicast mld

Displays MLD debugging status.

### Syntax:

```
show monitoring protocols multicast mld
```

### Operational mode

Use this command to show the status of MLD debugging.

The following example shows the status of MLD debugging.

```
vyatta@vyatta:~$show monitoring protocols multicast mld
MLD Debugging status:
MLD Decoder debugging is on
MLD Encoder debugging is on
MLD Events debugging is on
MLD FSM debugging is on
MLD Tree-Info-Base (TIB) debugging is on
```



# VRF support

## Command support for VRF routing instances

VRF allows an AT&T Vyatta vRouter to support multiple routing tables, one for each VRF routing instance. Some commands in this guide support VRF and can be applied to particular routing instances.

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

### Adding a VRF routing instance to a Configuration mode command

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

#### Example: Configuration mode example: syslog

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

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

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

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

**Example: Configuration mode example: SNMP**

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

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

For more information about configuring SNMP, refer to AT&T Vyatta Network Operating System Remote Management Configuration Guide.

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

**Adding a VRF routing instance to an Operational mode command**

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

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

**Example: Operational mode examples without optional parameters**

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

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

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

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

**Example: Operational mode example with optional parameters**

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

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

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

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

**Example: Operational mode example output: SNMP**

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

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

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

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

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



# List of Acronyms

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



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



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