VYATTA, INC. | Vyatta System

Guide to IPv6 Support

Vyatta
Suite 200
1301 Shoreway Road
Belmont, CA 94002
vyatta.com
650 413 7200
1 888 VYATTA 1 (US and Canada)

ISSUE DATE: April 2010

DOCUMENT REVISION. R6.0 v03

RELEASED WITH: R6.0

PART NO. A0-0226-10-0004

Contents

Quick Reference to Commands

Use this section to help you quickly locate a command.

Quick List of Examples

Use this list to help you locate examples you'd like to try or look at.

Preface

This guide describes the Vyatta system's support for IP version 6 (IPv6).

This preface provides information about using this guide. The following topics are covered:

- Intended Audience
- Organization of This Guide
- Document Conventions
- Vyatta Publications

Intended Audience

This guide is intended for experienced system and network administrators. Depending on the functionality to be used, readers should have specific knowledge in the following areas:

- Networking and data communications
- TCP/IP protocols
- General router configuration
- Routing protocols

- Network administration
- Network security

Organization of This Guide

This guide has the following aid to help you find the information you are looking for:

- Quick Reference to Commands

  Use this section to help you quickly locate a command.

This guide has the following chapters and appendixes:

Chapter	Description	Page
Chapter 1: IPv6 Support Overview	This chapter provides an overview of the Vyatta system's IPv6 implementation.	1
Chapter 2: IPv6 Configuration Examples	This chapter provides configuration examples for IPv6.	5
Chapter 3: IPv6 Commands	This chapter describes IPv6 commands	27
Chapter 4: IPv6 Interface Commands	This chapter describes commands for configuring IPv6 on various interfaces.	97
Glossary of Acronyms		176

Document Conventions

This guide contains advisory paragraphs and uses typographic conventions.

Advisory Paragraphs

This guide uses the following advisory paragraphs:

Warnings alert you to situations that may pose a threat to personal safety, as in the following example:

WARNING *Switch off power at the main breaker before attempting to connect the remote cable to the service power at the utility box.*

Cautions alert you to situations that might cause harm to your system or damage to equipment, or that may affect service, as in the following example:

CAUTION *Restarting a running system will interrupt service.*

Notes provide information you might need to avoid problems or configuration errors:

NOTE *You must create and configure network interfaces before enabling them for routing protocols.*

Typographic Conventions

This document uses the following typographic conventions:

`Monospace`	Examples, command-line output, and representations of configuration nodes.
`bold Monospace`	Your input: something you type at a command line.
bold	Commands, keywords, and file names, when mentioned inline. Objects in the user interface, such as tabs, buttons, screens, and panes.
italics	An argument or variable where you supply a value.
<key>	A key on your keyboard, such as <Enter>. Combinations of keys are joined by plus signs (“+”), as in <Ctrl>+c.
[*arg1* | *arg2*]	Enumerated options for completing a syntax. An example is [enable | disable].
num1–numN	A inclusive range of numbers. An example is 1–65535, which means 1 through 65535, inclusive.
arg1..argN	A range of enumerated values. An example is eth0..eth3, which means eth0, eth1, eth2, or eth3.
arg[arg...] arg[,arg...]	A value that can optionally represent a list of elements (a space-separated list in the first case and a comma-separated list in the second case).

Vyatta Publications

More information about the Vyatta system is available in the Vyatta technical library, and on www.vyatta.com and www.vyatta.org.

Full product documentation is provided in the Vyatta technical library. To see what documentation is available for your release, see the *Guide to Vyatta Documentation*. This guide is posted with every release of Vyatta software and provides a great starting point for finding the information you need.

Chapter 1: IPv6 Support Overview

This chapter provides an overview of the Vyatta system's IPv6 implementation.

NOTE *Vyatta support for some IPv6 features is "experimental." Features noted as experimental have not undergone Vyatta's complete QA testing cycle and as such have limited support.*

NOTE *Please note also that that IPv6 forwarding is enabled by default. If you want to disable IPv6 forwarding, use the following command in configuration mode:* ***set system ipv6 disable-forwarding****.*

This chapter presents the following topics:

- IPv6 Background
- IPv6 Addressing
- Special Addresses
- IPv6 Auto-Configuration

IPv6 Background

There are two versions of the Internet Protocol in use today. Version 4 (IPv4) is the version most commonly in use. IPv6 is version 6 of the Internet Protocol. The version currently in use by most devices is version 4 (IPv4). However, there are issues with IPv4, and the Internet Engineering Task Force (IETF) has designated IPv6 to succeed IPv4 as the next-generation protocol for use on the Internet.

IPv6 has a number of advantages over IPv4. The following are four important ones:

- Large address space.

  An IPv4 address consists of four bytes (32 bits). IPv6 addresses consist of 16 bytes (128 bits). The increase from 32 to 128 bits results in a huge number of available addresses: 79 billion billion billion times the addresses available in the IPv4—this is about 10^{38} addresses, or 10^{30} addresses for each person on the planet.

  The expanded address space means that IPv6 does not face the address exhaustion problems predicted imminently for IPv4. Furthermore, the availability for so many addresses means that private address spaces are not required, and address shortgage work-arounds such as Network Address Translation (NAT) can be eliminated. With no private addresses, there need be no hidden networks or hosts, and all devices can be globally reachable. A larger address space also means that features such as multihoming and aggregation are easier to implement.

- Support for mobile devices

  A special protocol, Mobile IP, is required to support mobility. Mobile IP is not automatic in IPv4, and there are several challenges involved in implementing Mobile IP on an IPv4 network. In contrast, Mobile IP was designed into IPv6 from its inception, and is a mandatory feature in a standards-compliant IPv6 protocol stack.

- Flexibility

  IPv6 Multiple levels of hierarchy in the address space. This allows for hierarchical allocation of addressing and more efficient route aggregation. It also permits new kinds of addresses not possible in IPv4, such as link- and site-scoped addressing

- Security

  Because devices can be globally reachable, end-to-end security can be employed, which is not possible on an internetwork with hidden networks and hosts. In addition, IP security (IPSec), which is an “add-on” feature in IPv4 networks, is mandatory in IPv6 networks, designed into the IPv6 protocol stack.

IPv6 Addressing

IP addresses generally take the following form:

x:*x*:*x*:*x*:*x*:*x*:*x*:*x*

where *x* is a 16-bit hexadecimal number; for example:

```
1EF7:0000:0000:0000:51DA:27C0:E4C2:0124
```

Addresses are case-insensitive; for example, the following is equivalent to example given above:

```
1EF7:0000:0000:0000:51da:27c0:E4c2:0124
```

Leading zeros are optional; for example, the following is a valid IPv6 address:

```
1EF7:0:0:0:51DA:27C0:E4C2:124
```

IPv6 addresses often contain many bytes with a value of zero. Successive fields of zeros can be represented by replacing them with a double colon, as in the following:

```
1EF7::51DA:27C0:E4C2:124
```

Similarly the following:

```
1EF7::124
```

is equivalent to the following:

```
1EF7:0:0:0:0:0:0:0124
```

and this:

```
0:0:0:0:0:0:0:1
```

is equivalent to this:

```
::1
```

The replacement by the double colon may be made only once within an address, as using the double colon more than once can result in ambiguity. For example, the following:

```
1EF7::27C0::E4C2:0124
```

is ambiguous between these three addresses:

```
1EF7:0000:27C0:0000:0000:0000:E4C2:0124
1EF7:0000:0000:27C0:0000:0000:E4C2:0124
1EF7:0000:0000:0000:27C0:0000:E4C2:0124
```

IPv6 addresses that are extensions of IPv4 addresses can be written in a mixed notation, where the the rightmost four bytes of the IPv6 address are replaced with the four decimal octets of the IPv4 address. In mixed notation, the four hexademical bytes are separated by colons and the four decimal octets are separated by dots, as in the following example:

```
1EF7:0:0:0:192.168.100.51
```

Special Addresses

Like IPv4, IPv6 has some special addresses, which are used by convention for special functions. For unicast addresses, these include the following:

- The unspecified address. This address is used as a placeholder) when no address is available (for example, in an initial DHCP address), or to stand for “any” address. In IPv6, the unspecified address can be represented as either of the following:

  ```
  0:0:0:0:0:0:0:0
  ::
  ```

- The localhost (loopback) interface. The loopback interface is a software interface that represents the local device itself. In IPv4, the address 127.0.0.1 is used by convention for the loopback interface. In IPv6, the loopback interface can be represented by either of the following:

  ```
  0:0:0:0:0:0:0:1
  ::1
  ```

The IPv6 address architecture is quite rich, and includes types of addressing unavailable in IPv4, such as unicast and multicast scoped addresses, aggregatable global addresses, and anycast addresses. Multicast broadcast addresses do not exist in IPv6. For more information about the IPv6 address architecture, consult RFC 4291, *IP Version 6 Addressing Architecture.*

IPv6 Auto-Configuration

IPv6 supports two mechanisms for automatically configuring devices with IP addresses: stateful and stateless. The Vyatta system currently supports only stateless configuration.

In stateful configuration, addressing and service information is distributed by a protocol (DHCPv6) in the same way that the Dynamic Host Configuration Protocol (DHCP) distributes information for IPv4. This information is “stateful” in that both the DHCP server and the DHCP client must maintain the addressing and service information. The Vyatta system does not currently support DHCPv6.

The Vyatta system supports stateless configuration using the Stateless Address Auto-Configuration (SLAAC) protocol, which is a component of the larger Neighbor Discover (ND) protocol. SLAAC has a host component and a router component.

In the host component of SLAAC, the IPv6 system constructs its own unicast global address from the system’s network prefix together with its Ethernet media access control (MAC) address. The device proposes this address to the network, without requiring approval from a server such as a DHCP server. The combination of network prefix and MAC address is assumed to be unique. Stateless auto-configuration is performed by default by most IPv6 systems, including the Vyatta system.

In the router component of SLAAC, routers multicast ND Router Advertisement (RA) and Router Solicitation (RS) packets that include prefix information. Hosts receive these advertisements and use them to form globally unique IPv6 addresses. The RS and RA packets also provide the router discovery function, allowing hosts to locate routers that are configured to serve as default routers. The Vyatta system fully supports router-side SLAAC and router discovery, including all required configurable parameters.

The ND protocol and the router discovery function are specified in RFC 4861. IPv6 Stateless Address Auto-Configuration is described in RFC 4862.

Chapter 2: IPv6 Configuration Examples

This chapter provides configuration examples for IPv6.

This chapter presents the following topics:

- IPv6 Basics
- Static IPv6 Routing
- Dynamic IPv6 Routing with RIPng
- Dynamic IPv6 Routing with BGP
- Tunneling IPv6 traffic in IPv4

IPv6 Basics

This section provides an overview of how to provide basic IPv6 configuration and operation. This section presents the following topics:

- Verify IPv6 Support
- Configure an IPv6 Address on an Interface
- Display the IPv6 Routing Table
- Confirm Connectivity
- Display IPv6 Neighbor Discovery (ND) Cache

- Clear ND Cache

Verify IPv6 Support

A simple step to verify that IPv6 support is available is to ping the loopback interface. To verify IPv6 support, perform the following step in operational mode.

Example 2-1 Confirm IPv6 support

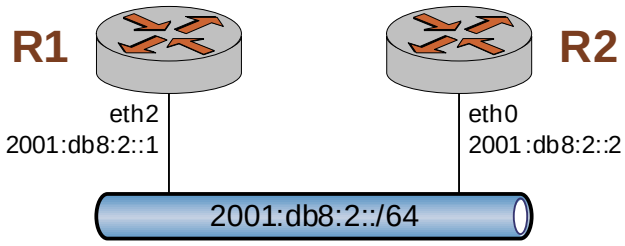
Step	Command

Ping the loopback interface.	`vyatta@R1:~$ ping6 ::1`
	`PING ::1(::1) 56 data bytes`
	`64 bytes from ::1: icmp_seq=1 ttl=64 time=2.13 ms`
	`64 bytes from ::1: icmp_seq=2 ttl=64 time=0.086 ms`
	`^C`
	`--- ::1 ping statistics ---`
	`2 packets transmitted, 2 received, 0% packet loss, time 1006ms`
	`rtt min/avg/max/mdev = 0.086/1.112/2.138/1.026 ms`

Configure an IPv6 Address on an Interface

Figure 2-1 shows a simple network with two IPv6 nodes.

Figure 2-1 IPv6 address on an interface



IPv6 addresses are configured on Ethernet interfaces in the same way that IPv4 addresses are.To configure eth2 on R1, perform the following steps in configuration mode.

Example 2-2 Add an IPv6 address to eth2 on R1

Step	Command
Add the IPv6 address to the eth2 interface.	`vyatta@R1# set interfaces ethernet eth2 address 2001:db8:2::1/64` `[edit]`
Commit the change.	`vyatta@R1# commit` `[edit]`

Change to operational mode	`vyatta@R1# exit`
	`exit`
	`vyatta@R1:~$`

Show the status of the interfaces on R1.	`vyatta@R1:~$ show interfaces`
	`Interface IP Address State Link Description`
	`eth0 - up up`
	`eth1 - up up`
	`eth2 2001:DB8:2::1/64 up up`
	`eth3 - up up`
	`lo 127.0.0.1/8 up up`
	`lo ::1/128 up up`

To configure eth0 on R2, perform the following steps in configuration mode.

Example 2-3 Add an IPv6 address to eth0 on R2

Step	Command
Add the IPv6 address to the eth0 interface.	`vyatta@R2# set interfaces ethernet eth0 address 2001:db8:2::2/64` `[edit]`
Commit the change.	`vyatta@R2# commit` `[edit]`

Change to operational mode	`vyatta@R2# exit`
	`exit`
	`vyatta@R2:~$`

Example 2-3 Add an IPv6 address to eth0 on R2

Show the status of the interfaces on R2.	vyatta@R2:~$ **show interfaces** Interface IP Address State Link Description eth0 2001:DB8:2::2/64up up eth1 - up up eth2 - up up lo 127.0.0.1/8 up up lo ::1/128 up up

Display the IPv6 Routing Table

When an IPv6 address is added to an interface a connected network for it appears in the routing table. To display the routing table, perform the following step in operational mode.

Example 2-4 Display the IPv6 routing table

Step	Command
Show the routing table.	vyatta@R1:~$ **show ipv6 route** Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3, I - ISIS, B - BGP, * - FIB route. C>* ::1/128 is directly connected, lo C>* 2001:db8:2::/64 is directly connected, eth2 C * fe80::/64 is directly connected, eth2 C * fe80::/64 is directly connected, eth1 C>* fe80::/64 is directly connected, eth0 K>* ff00::/8 is directly connected, eth2

Confirm Connectivity

To confirm that R1 and R2 can communicate, use the **ping6** command. To confirm connectivity, perform the following step in operational mode.

Example 2-5 Confirm connectivity between R1 and R2

Step	Command
Ping R2 from R1.	vyatta@R1:~$ **ping6 2001:db8:2::2** PING 2001:db8:2::2(2001:db8:2::2) 56 data bytes 64 bytes from 2001:db8:2::2: icmp_seq=1 ttl=64 time=6.52 ms 64 bytes from 2001:db8:2::2: icmp_seq=2 ttl=64 time=0.333 ms ^C --- 2001:db8:2::2 ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 1013ms rtt min/avg/max/mdev = 0.333/3.427/6.522/3.095 ms

Display IPv6 Neighbor Discovery (ND) Cache

To display a list of neighbors in the Neighbor Discovery (ND) cache, use the **show ipv6 neighbors** command. To display the ND cache, perform the following step in operational mode.

Example 2-6 Display the ND cache

Step	Command
Display the list of known neighbors.	vyatta@R1:~$ **show ipv6 neighbors** 2001:db8:2::2 dev eth2 lladdr 00:0c:29:4e:fc:b6 router REACHABLE fe80::20c:29ff:fe4e:fcb6 dev eth2 lladdr 00:0c:29:4e:fc:b6 DELAY

Clear ND Cache

To clear the Neighbor Discovery (ND) cache, use the **clear ipv6 neighbors** command. To clear the ND cache on interface eth2, perform the following step in operational mode.

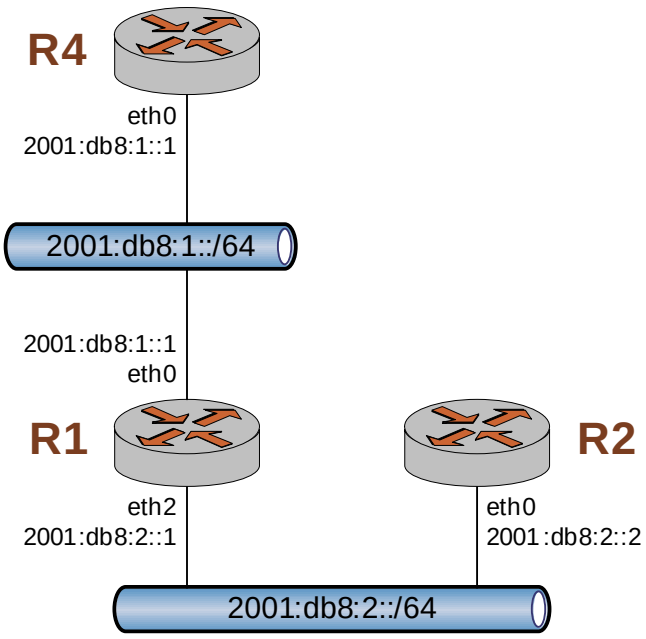
Example 2-7 Clear the ND cache

Step	Command
Clear the list of known neighbors on eth2.	vyatta@R1:~$ **clear ipv6 neighbors interface eth2**

Static IPv6 Routing

Figure 2-2 shows a more complex network with three nodes. In this example we will show configuration of the nodes using static routes to enable R2 and R4 to communicate via R1.

Figure 2-2 Static IPv6 routing example



Verify that IPv6 Forwarding is Enabled

In order for R1 to be able to pass data between interfaces eth0 and eth2 (i.e. between R4 and R2) it must be configured to enable forwarding. To determine if forwarding is enabled, perform the following step in operational mode.

Example 2-8 Determine if forwarding is enabled on R1

Step	Command
Display the state of IPv6 forwarding on R1.	`vyatta@R1:~$ show ipv6 forwarding` `ipv6 forwarding is off`

If forwarding is not enabled, as is the case in Example 2-8, the system must be configured to enable forwarding. To enable forwarding, perform the following steps in configuration mode.

Example 2-9 Enable forwarding on R1

Step	Command
Enable forwarding on R1.	`vyatta@R1# delete system ipv6 disable-forwarding` `[edit]`
Commit the change.	`vyatta@R1# commit` `[edit]`

Change to operational mode	`vyatta@R1# exit`
	`exit`
	`vyatta@R1:~$`

Display the state of IPv6 forwarding on R1.	`vyatta@R1:~$ show ipv6 forwarding` `ipv6 forwarding is on`

Add the Default IPv6 Route

On R4, all traffic that is not routed elsewhere will be sent to R1. To configure the default route, perform the following steps in configuration mode.

Example 2-10 Add the default route on R4

Step	Command
Add the default route on R4.	`vyatta@R4# set protocols static route6 ::/0 next-hop 2001:db8:1::1` `[edit]`
Commit the change.	`vyatta@R4# commit` `[edit]`

Change to operational mode.	`vyatta@R4# exit`
	`exit`
	`vyatta@R4:~$`

Example 2-10 Add the default route on R4

Verify the default route in the routing table.	vyatta@R4:~$ **show ipv6 route** Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3, I - ISIS, B - BGP, * - FIB route. S>* ::/0 [1/0] via 2001:db8:1::1, eth0 C>* ::1/128 is directly connected, lo C>* 2001:db8:1::/64 is directly connected, eth0 C * fe80::/64 is directly connected, eth1 C>* fe80::/64 is directly connected, eth0 K>* ff00::/8 is directly connected, eth0

Add a Static IPv6 Route

As an alternative to the default route we created on R4, we'll create a static route on R2. To configure a static route to the 2001:db8:1::/64 network, perform the following steps in configuration mode.

Example 2-11 Add a static route on R2

Step	Command
Add a static route on R2.	vyatta@R2# **set protocols static route6 2001:db8:1::/64 next-hop 2001:db8:2::1** [edit]
Commit the change.	vyatta@R2# **commit** [edit]
Change to operational mode.	vyatta@R2# **exit** exit vyatta@R2:~$
Verify the static route in the routing table.	vyatta@R2:~$ **show ipv6 route** Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3, I - ISIS, B - BGP, * - FIB route. C>* ::1/128 is directly connected, lo S>* 2001:db8:1::/64 [1/0] via 2001:db8:2::1, eth0 C>* 2001:db8:2::/64 is directly connected, eth0 C * fe80::/64 is directly connected, eth1 C>* fe80::/64 is directly connected, eth0 K>* ff00::/8 is directly connected, eth0

Confirm Connectivity

To confirm that R2 and R4 can communicate, use the **ping6** command. To confirm connectivity between R2 and R4, perform the following step in operational mode.

Example 2-12 Confirm connectivity between R2 and R4

Step	Command

Ping R4 from R2.	`vyatta@R2:~$ ping6 2001:db8:1::4`
	`PING 2001:db8:1::4(2001:db8:1::4) 56 data bytes`
	`64 bytes from 2001:db8:1::4: icmp_seq=1 ttl=63 time=5.65 ms`
	`64 bytes from 2001:db8:1::4: icmp_seq=2 ttl=63 time=0.382 ms`
	`^C`
	`--- 2001:db8:1::4 ping statistics ---`
	`2 packets transmitted, 2 received, 0% packet loss, time 1011ms`
	`rtt min/avg/max/mdev = 0.382/3.016/5.650/2.634 ms`

As an alternative, use **traceroute6** to verify that the goes from R2 to R1 to R4. To confirm connectivity between R2 and R4 through R1 using **traceroute6**, perform the following step in operational mode.

Example 2-13 Confirm connectivity between R2 and R4 via R1

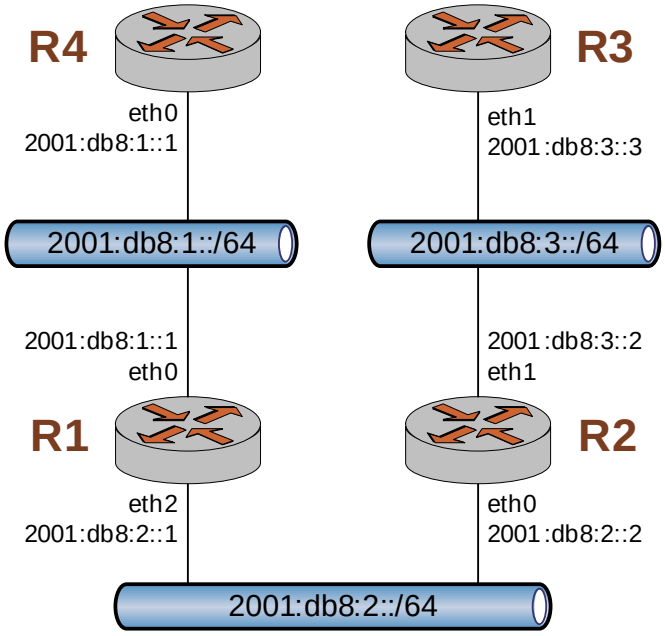
Step	Command

Trace the route from R2 to R4.	`vyatta@R2:~$ traceroute6 2001:db8:1::4`
	`traceroute to 2001:db8:1::4 (2001:db8:1::4), 30 hops max, 40 byte packets`
	`1 (2001:db8:2::1) 4.448 ms 4.148 ms 4.092 ms`
	`2 (2001:db8:1::4) 4.297 ms 4.306 ms 4.308 ms`

Dynamic IPv6 Routing with RIPng

Figure 2-3 shows a network with four nodes. In this example we will show configuration of the nodes using dynamic IPv6 routing with RIPng to enable R3 and R4 to communicate via R1 and R2.

Figure 2-3 Dynamic IPv6 routing example - RIPng



Enable Forwarding on R1 and R2

For R1 to be able to pass data between interfaces eth0 and eth2, and for R2 to be able to pass data between interfaces eth0 and eth1, they must be configured to enable forwarding. To enable forwarding on R1, perform the following step in configuration mode..

Example 2-14 Enable forwarding on R1

Step	Command
Enable forwarding on R1.	`vyatta@R1# delete system ipv6 disable-forwarding` `[edit]`
Commit the change.	`vyatta@R1# commit` `[edit]`

To enable forwarding on R2, perform the following steps in configuration mode..

Example 2-15 Enable forwarding on R2

Step	Command
Enable forwarding on R2.	`vyatta@R2# delete system ipv6 disable-forwarding` `[edit]`

Example 2-15 Enable forwarding on R2

Commit the change.	`vyatta@R2# commit` `[edit]`

Enable RIPng on an Interface

In order to enable dynamic routing using RIPng, it must be enabled on the interfaces that are to use it. To enable RIPng on R1, perform the following steps in configuration mode.

Example 2-16 Enable RIPng on R1

Step	Command
Enable RIPng on eth0.	`vyatta@R1# set protocols ripng interface eth0` `[edit]`
Enable RIPng on eth2.	`vyatta@R1# set protocols ripng interface eth2` `[edit]`
Commit the change.	`vyatta@R1# commit` `[edit]`

Change to operational mode.	`vyatta@R1# exit`
	`exit`
	`vyatta@R1:~$`

Example 2-16 Enable RIPng on R1

Verify the status of RIPng.	vyatta@R1:~$ **show ipv6 ripng status** Routing Protocol is "RIPng" Sending updates every 30 seconds with +/-50%, next due in 4 seconds Timeout after 180 seconds, garbage collect after 120 seconds Outgoing update filter list for all interface is not set Incoming update filter list for all interface is not set Default redistribution metric is 1 Redistributing: Default version control: send version 1, receive version 1 Interface Send Recv eth0 1 1 eth2 1 1 Routing for Networks: eth0 eth2 Routing Information Sources: Gateway BadPackets BadRoutes Distance Last Update fe80::20c:29ff:fed7:c4a4 0 0 120 00:00:25

Advertise Connected Networks

The **redistribute** command is then used to advertise the connected networks. To advertise connected networks on R1, perform the following steps in configuration mode.

Example 2-17 Advertise connected networks on R1

Step	Command
Advertise connected networks via ripng.	vyatta@R1# **set protocols ripng redistribute connected** [edit]
Commit the change.	vyatta@R1# **commit** [edit]

Confirm Visibility of Remote Networks

After enabling RIPng on the other interfaces of R2, R3, and R4, and advertising connected networks on R2, we can check the routing table of R4 to verify that it has learned the network. To confirm visibility of remote networks on R4, perform the following step in operational mode.

Example 2-18 Confirm visibility of remote networks on R4

Step	Command
Trace the route from R2 to R4.	vyatta@R4:~$ **show ipv6 route** Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3, I - ISIS, B - BGP, * - FIB route. S>* ::/0 [1/0] via 2001:db8:1::1, eth0 C>* ::1/128 is directly connected, lo C>* 2001:db8:1::/64 is directly connected, eth0 R>* 2001:db8:2::/64 [120/2] via fe80::20c:29ff:fed6:816c, eth0, 00:43:00 R>* 2001:db8:3::/64 [120/3] via fe80::20c:29ff:fed6:816c, eth0, 00:00:03 C>* fe80::/64 is directly connected, eth0

The "R" in the first column indicates that two routes have been learned from RIPng. Since there is now a route for 2001:db8:3::/64 we should be able to ping R3. To confirm connectivity, perform the following steps in operational mode.

Example 2-19 Confirm connectivity between R4 and R3

Step	Command
Ping R3 from R4.	vyatta@R4:~$ **ping6 2001:db8:3::3** PING 2001:db8:3::3(2001:db8:3::3) 56 data bytes 64 bytes from 2001:db8:3::3: icmp_seq=1 ttl=62 time=5.98 ms 64 bytes from 2001:db8:3::3: icmp_seq=2 ttl=62 time=0.603 ms ^C --- 2001:db8:3::3 ping statistics --- 2 packets transmitted, 2 received, 0% packet loss, time 1011ms rtt min/avg/max/mdev = 0.603/3.294/5.986/2.692 ms

Example 2-19 Confirm connectivity between R4 and R3

Display the RIPng status.

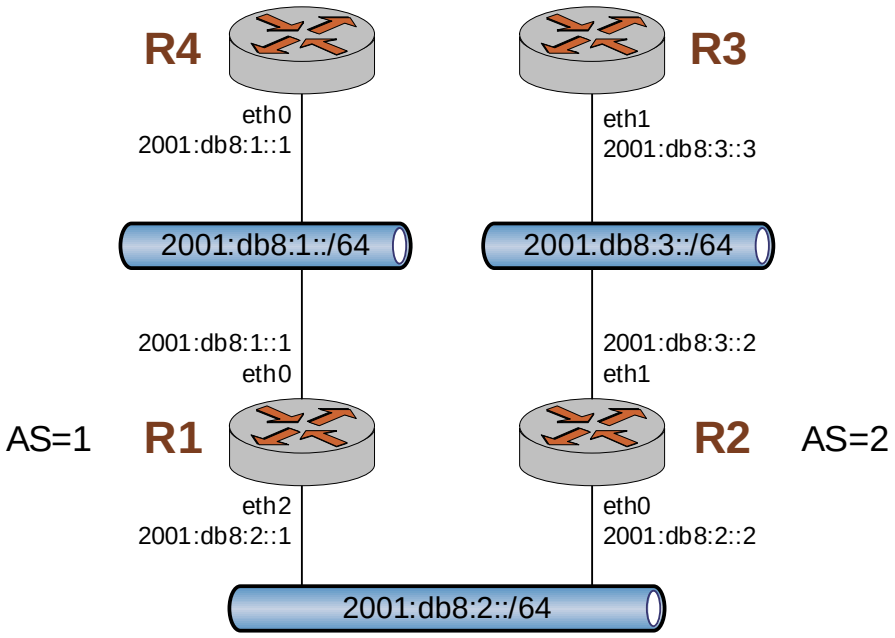
```
vyatta@R4:~$ show ipv6 ripng
Codes: R - RIPng, C - connected, S - Static, O - OSPF,
B - BGP
Sub-codes:
      (n) - normal, (s) - static, (d) - default, (r)
- redistribute,
      (i) - interface, (a/S) - aggregated/Suppressed

   Network      Next Hop                      Via
Metric Tag Time
C(i) 2001:db8:1::/64
                ::                      self      1   0
R(n) 2001:db8:2::/64
                fe80::20c:29ff:fed6:816c    eth0
2    0  02:56
R(n) 2001:db8:3::/64
                fe80::20c:29ff:fed6:816c    eth0
3    0  02:56
```

Dynamic IPv6 Routing with BGP

Figure 2-4 shows a network with four nodes. In this example we will show configuration of the nodes using dynamic IPv6 routing with BGP to enable R3 and R4 to communicate via R1 and R2.

Figure 2-4 Dynamic IPv6 routing example - BGP



Enable Forwarding on R1 and R2

For R1 to be able to pass data between interfaces eth0 and eth2, and for R2 to be able to pass data between interfaces eth0 and eth1, they must be configured to enable forwarding. To enable forwarding on R1, perform the following steps in configuration mode..

Example 2-20 Enable forwarding on R1

Step	Command
Enable forwarding on R1.	`vyatta@R1# delete system ipv6 disable-forwarding` `[edit]`
Commit the change.	`vyatta@R1# commit` `[edit]`

To enable forwarding on R2, perform the following steps in configuration mode..

Example 2-21 Enable forwarding on R2

Step	Command
Enable forwarding on R2.	`vyatta@R2# delete system ipv6 disable-forwarding` `[edit]`

Example 2-21 Enable forwarding on R2

Commit the change.	`vyatta@R2# commit` `[edit]`

Configure IPv6 BGP Peer

R1 and R2 must be configured to determine how to access remote Autonomous Systems (AS). To configure R1 to see AS=2, perform the following steps in configuration mode..

Example 2-22 Configure R1 to see AS=2

Step	Command
Configure a BGP peer on R1.	`vyatta@R1# set protocols bgp 1 neighbor 2001:db8:2::2 remote-as 2` `[edit]`
Commit the change.	`vyatta@R1# commit` `[edit]`

Similarly, To configure R2 to see AS=1, perform the following steps in configuration mode..

Example 2-23 Configure R2 to see AS=1

Step	Command
Configure a BGP peer on R2.	`vyatta@R2# set protocols bgp 2 neighbor 2001:db8:2::1 remote-as 1` `[edit]`
Commit the change.	`vyatta@R2# commit` `[edit]`

To confirm that the peer session is "Established", perform the following step in operational mode.

Example 2-24 Confirm peer session is established

Step	Command
Display the status of the BGP neighbor.	vyatta@R1:~$ **show ip bgp neighbors 2001:db8:2::2** BGP neighbor is 2001:db8:2::2, remote AS 2, local AS 1, external link BGP version 4, remote router ID 172.16.139.160 BGP state = Established, up for 00:01:24 Last read 00:00:24, hold time is 180, keepalive interval is 60 seconds Neighbor capabilities: 4 Byte AS: advertised and received Route refresh: advertised and received(old & new) Address family IPv6 Unicast: advertised and received Message statistics: Inq depth is 0 Outq depth is 0 Sent Rcvd Opens: 2 0 Notifications: 0 0 Updates: 0 0 Keepalives: 3 2 Route Refresh: 0 0 Capability: 0 0 Total: 5 2 Minimum time between advertisement runs is 30 seconds For address family: IPv6 Unicast Community attribute sent to this neighbor(both) 0 accepted prefixes Connections established 1; dropped 0 Last reset never Local host: 2001:db8:2::1, Local port: 179 Foreign host: 2001:db8:2::2, Foreign port: 55711 Nexthop: 172.16.117.128 Nexthop global: 2001:db8:2::1 Nexthop local: fe80::20c:29ff:fed6:8180 BGP connection: shared network Read thread: on Write thread: off

Advertise Connected Networks

The **redistribute** command is then used to advertise the connected networks. To advertise connected networks on R1, perform the following steps in configuration mode.

Example 2-25 Advertise connected networks on R1

Step	Command
Advertise connected networks via bgp.	vyatta@R1# **set protocols bgp 1 ipv6 redistribute connected** [edit]
Commit the change.	vyatta@R1# **commit** [edit]

Confirm Advertised Routes

To see which routes are being advertised by R1 and which routes have been learned from peers, perform the following steps in operational mode.

Example 2-26 Confirm routes advertised and learned by R1

Step	Command
Display routes advertised by R1.	vyatta@R1:~$ **show ipv6 bgp neighbors 2001:db8:2::2 advertised-routes** BGP table version is 0, local router ID is 172.16.117.128 Status codes: s suppressed, d damped, h history, * valid, > best, i - internal, r RIB-failure, S Stale, R Removed Origin codes: i - IGP, e - EGP, ? - incomplete Network Next Hop Metric LocPrf Weight Path *> 2001:db8:1::/64 2001:db8:2::1 1 32768 ? *> 2001:db8:2::/64 2001:db8:2::1 1 32768 ? Total number of prefixes 2

Example 2-26 Confirm routes advertised and learned by R1

Display routes learned by R1.

```
vyatta@R1:~$ show ipv6 bgp neighbors 2001:db8:2::2
routes
BGP table version is 0, local router ID is
172.16.117.128
Status codes: s suppressed, d damped, h history, *
valid, > best, i - internal,
              r RIB-failure, S Stale, R Removed
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf
Weight Path
*  2001:db8:2::/64  2001:db8:2::2         1
0 2 ?
*> 2001:db8:3::/64  2001:db8:2::2         1
0 2 ?

Total number of prefixes 2
```

Display the routing table.

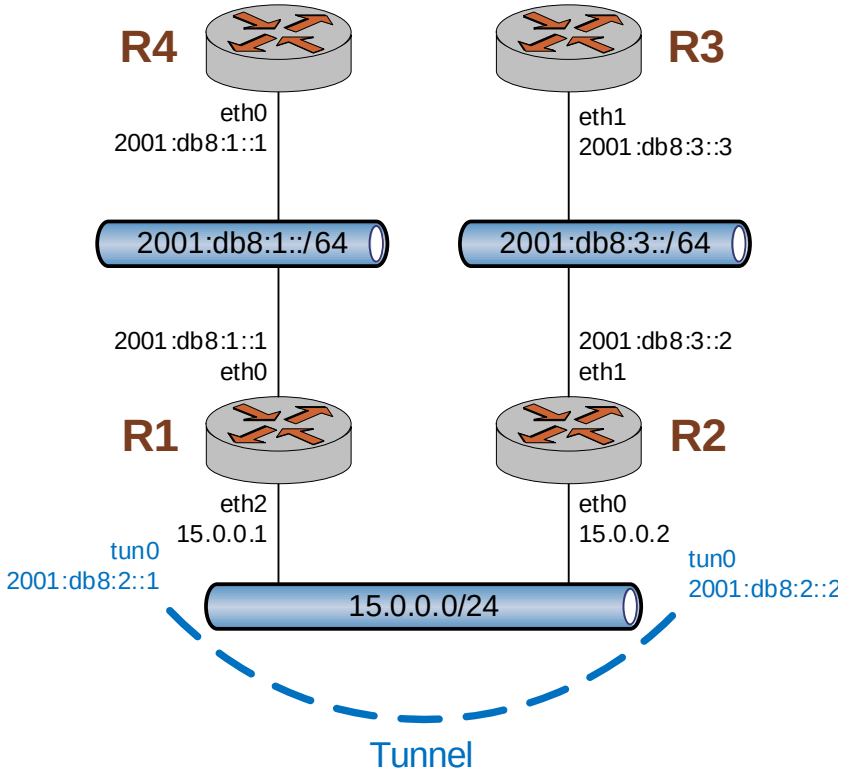
```
vyatta@R1:~$ show ipv6 route
Codes: K - kernel route, C - connected, S - static, R
- RIPng, O - OSPFv3,
       I - ISIS, B - BGP, * - FIB route.

C>* ::1/128 is directly connected, lo
C>* 2001:db8:1::/64 is directly connected, eth0
C>* 2001:db8:2::/64 is directly connected, eth2
B>* 2001:db8:3::/64 [20/1] via
fe80::20c:29ff:fe4e:fcb6, eth2, 00:22:47
C * fe80::/64 is directly connected, eth2
C>* fe80::/64 is directly connected, eth0
```

Tunneling IPv6 traffic in IPv4

Figure 2-5 shows a network with four nodes. R1 and R2 each have an interface configured as IPv6 and an interface configured as IPv4. In this example we will show configuration of the nodes using tunneling over IPv4 to enable R3 and R4 to communicate via R1 and R2.

Figure 2-5 Tunneling IPv6 traffic in IPv4 example



We assume that all interfaces have been configured with IP addresses as shown in the example. We will also assume that R1 and R2 have forwarding enabled.

Create SIT tunnel

R1 and R2 must be configured to create a tunnel between them in order to encapsulate the IPv6 traffic. To configure R1 to create a tunnel using SIT (Simple Internet Transition) encapsulation between 15.0.0.1 and 15.0.0.2, perform the following steps in configuration mode..

Example 2-27 Configure tunnel interface on R1

Step	Command
Create a tunnel with SIT encapsulation.	`vyatta@R1# set interfaces tunnel tun0 encapsulation sit` `[edit]`
Specify the local IP address.	`vyatta@R1# set interfaces tunnel tun0 local-ip 15.0.0.1` `[edit]`

Example 2-27 Configure tunnel interface on R1

Specify the remote IP address.	`vyatta@R1# set interfaces tunnel tun0 remote-ip 15.0.0.2` `[edit]`
Configure the IPv6 address on the interface.	`vyatta@R1# set interfaces tunnel tun0 address 2001:db8:2::1/64` `[edit]`
Commit the change.	`vyatta@R1# commit` `[edit]`

To configure R2 to create a tunnel using SIT (Simple Internet Transition) encapsulation between 15.0.0.2 and 15.0.0.1, perform the following steps in configuration mode..

Example 2-28 Configure tunnel interface on R2

Step	Command
Create a tunnel with SIT encapsulation.	`vyatta@R2# set interfaces tunnel tun0 encapsulation sit` `[edit]`
Specify the local IP address.	`vyatta@R2# set interfaces tunnel tun0 local-ip 15.0.0.2` `[edit]`
Specify the remote IP address.	`vyatta@R2# set interfaces tunnel tun0 remote-ip 15.0.0.1` `[edit]`
Configure the IPv6 address on the interface.	`vyatta@R2# set interfaces tunnel tun0 address 2001:db8:2::2/64` `[edit]`
Commit the change.	`vyatta@R2# commit` `[edit]`

At this point there is connectivity between R1 and R2 across the tunnel interface. The following shows a capture of a ping from 2001:db8:2::1 to 2001:db8:2::2. Notice that the IPv6 ping packet in encapsulated by the IPv4 header:

Example 2-29 Capture of ping

```
Frame 22 (138 bytes on wire, 138 bytes captured)
Ethernet II, Src: Vmware_d6:81:80 (00:0c:29:d6:81:80), Dst: Vmware_4e:fc:b6
(00:0c:29:4e:fc:b6)
```

```
    Destination: Vmware_4e:fc:b6 (00:0c:29:4e:fc:b6)
    Source: Vmware_d6:81:80 (00:0c:29:d6:81:80)
    Type: IP (0x0800)
Internet Protocol, Src: 15.0.0.1 (15.0.0.1), Dst: 15.0.0.2 (15.0.0.2)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 124
    Identification: 0x0000 (0)
    Flags: 0x04 (Don't Fragment)
    Fragment offset: 0
    Time to live: 255
    Protocol: IPv6 (0x29)
    Header checksum: 0x5d56 [correct]
    Source: 15.0.0.1 (15.0.0.1)
    Destination: 15.0.0.2 (15.0.0.2)
Internet Protocol Version 6
    0110 .... = Version: 6
    .... 0000 0000 .... .... .... .... .... = Traffic class: 0x00000000
    .... .... .... 0000 0000 0000 0000 0000 = Flowlabel: 0x00000000
    Payload length: 64
    Next header: ICMPv6 (0x3a)
    Hop limit: 64
    Source: 2001:db8:2::1 (2001:db8:2::1)
    Destination: 2001:db8:2::2 (2001:db8:2::2)
Internet Control Message Protocol v6
    Type: 129 (Echo reply)
    Code: 0
    Checksum: 0x2fca [correct]
    ID: 0xe825
    Sequence: 0x001b
    Data (56 bytes)

0000  9b a8 25 49 58 0c 07 00 08 09 0a 0b 0c 0d 0e 0f   ..%IX...........
0010  10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f   ................
0020  20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f    !"#$%&'()*+,-./
0030  30 31 32 33 34 35 36 37                          01234567
```

Chapter 3: IPv6 Commands

This chapter describes IPv6 commands

This chapter contains the following commands.

Configuration Commands	
BGP	
protocols bgp <asn> address-family ipv6-unicast aggregate-address <ipv6net>	Specifies a block of IPv6 addresses to aggregate.
protocols bgp <asn> address-family ipv6-unicast network <ipv6net>	Specifies an IPv6 network to be advertised by the BGP routing process.
protocols bgp <asn> address-family ipv6-unicast redistribute connected	Redistributes directly connected routes into BGP.
protocols bgp <asn> address-family ipv6-unicast redistribute kernel	Redistributes kernel routes into BGP.
protocols bgp <asn> address-family ipv6-unicast redistribute ospfv3	Redistributes routes learned from OSPFv3 into BGP.
protocols bgp <asn> address-family ipv6-unicast redistribute ripng	Redistributes routes learned from RIPng into BGP.
protocols bgp <asn> address-family ipv6-unicast redistribute static	Redistributes static routes into BGP.
protocols bgp <asn> neighbor <ipv6> address-family ipv6-unicast distribute-list export <acl-num>	Applies an access list to filter outbound routing updates to this neighbor.
protocols bgp <asn> neighbor <ipv6> address-family ipv6-unicast distribute-list import <acl-num>	Applies an access list to filter inbound routing updates from this neighbor.
protocols bgp <asn> neighbor <ipv6> address-family ipv6-unicast prefix-list export <list-name>	Applies an prefix list to filter updates to this neighbor.
protocols bgp <asn> neighbor <ipv6> address-family ipv6-unicast prefix-list import <list-name>	Applies an prefix list to filter updates from this neighbor.
RIPng	
protocols ripng aggregate-address <ipv6net>	Specifies an aggregate RIPng route announcement.
protocols ripng network <ipv6net>	Specifies a network for the RIPng.
protocols ripng route <ipv6net>	Specifies a RIPng static route.
Static Routes	
protocols static interface-route6 <subnet> next-hop-interface <ethx>	Allows you to configure the next hop interface for an interface-based IPv6 static route.
protocols static route6 <subnet> blackhole	Allows you to configure a blackhole IPv6 static route.

protocols static route6 <subnet> next-hop <address>	Allows you to configure the next hop for an IPv6 static route.
System Commands	
system ipv6 blacklist	Disables IPv6 on the system.
system ipv6 disable	Disables assignment of IPv6 addresses on all interfaces.
system ipv6 disable-forwarding	Disables IPv6 forwarding on all interfaces.
system ipv6 strict-dad	Disables IPv6 operation on an interface when DAD fails for a link-local address.
Operational Commands	
BGP	
show ipv6 bgp	Displays BGP routes.
show ipv6 bgp community <community>	Displays BGP routes belonging to the specified BGP community.
show ipv6 bgp community-list <list-name>	Displays BGP routes permitted by the specified community list.
show ipv6 bgp filter-list <list-num>	Displays routes matching a list of autonomous system paths.
show ipv6 bgp neighbors	Displays BGP neighbor information.
show ipv6 bgp neighbors <ipv6> advertised-routes	Displays advertised routes for a BGP neighbor.
show ipv6 bgp neighbors <ipv6> received-routes	Displays routes received from a BGP neighbor.
show ipv6 bgp neighbors <ipv6> routes	Displays all received and accepted routes from a BGP neighbor.
show ipv6 bgp prefix-list <list-name>	Displays BGP routes matching a prefix list.
show ipv6 bgp regexp <regexp>	Displays routes matching an AS path regular expression.
System Management	
clear ipv6 neighbors address <ipv6>	Clears a specific IPv6 address from the IPv6 ND cache.
clear ipv6 neighbors interface <ethx>	Clears the system's IPv6 ND cache for a specific interface.
show ipv6 neighbors	Displays the system's IPv6 ND cache.
Forwarding and Routing	
clear ipv6 prefix-list	Clears prefix list statistics or status.
clear ipv6 route cache	Flushes the kernel IPv6 route cache.

show ipv6 route	Displays IPv6 routes stored in the RIB and FIB.
show ipv6 route <ipv6net> longer-prefixes	Displays IPv6 prefixes longer than a specified prefix.
show ipv6 route bgp	Displays IPv6 BGP routes.
show ipv6 route cache	Displays the kernel IPv6 route cache.
show ipv6 route connected	Displays IPv6 connected routes.
show ipv6 route forward	Displays IPv6 routes stored in the FIB.
show ipv6 route kernel	Displays IPv6 kernel routes.
show ipv6 route ripng	Displays IPv6 RIPng routes.
show ipv6 route static	Displays IPv6 static routes.
Diagnostics	
ping6 <host>	Sends ICMP ECHO_REQUEST packets to IPv6 network hosts.
traceroute6 <host>	Displays the route packets take to an IPv6 network host.
RIPng	
show ipv6 ripng	Displays information for the Routing Information Protocol next generation (RIPng).

clear ipv6 neighbors address <ipv6>

Clears a specific IPv6 address from the IPv6 ND cache.

Syntax

clear ipv6 neighbors address *ipv6*

Command Mode

Operational mode.

Parameters

ipv6	Clears the ND (Neighbor Discovery) cache of the specified IPv6 address.

Default

None.

Usage Guidelines

Use this command to remove entries associated with a specific IPv6 address from the Neighbor Discovery cache.

clear ipv6 neighbors interface <ethx>

Clears the system's IPv6 ND cache for a specific interface.

Syntax

clear ipv6 neighbors interface *eth0..eth23*

Command Mode

Operational mode.

Parameters

eth0..eth23	Clears the entire IPv6 ND (Neighbor Discovery) cache for the specified Ethernet interface. The range of values is **eth0** to **eth23**.

Default

None.

Usage Guidelines

Use this command to remove entries associated with an Ethernet interface from the IPv6 Neighbor Discovery cache.

clear ipv6 prefix-list

Clears prefix list statistics or status.

Syntax

clear ipv6 prefix-list [*list-name* [*ipv6net*]]

Command Mode

Operational mode.

Parameters

list-name	Optional. Clears statistics for the specified prefix list.
ipv6net	Optional. Clears statistics for the specified network.

Default

Statistics for all prefix-lists are cleared.

Usage Guidelines

Use this command to clear prefix list statistics or status.

clear ipv6 route cache

Flushes the kernel IPv6 route cache.

Syntax

clear ipv6 route cache [*ipv6net*]

Command Mode

Operational mode.

Parameters

ipv6net	Optional. Flushes the specified route from the kernel IPv6 route cache.

Default

Flushes the entire IPv6 route cache.

Usage Guidelines

Use this command to flush the kernel IPv6 route cache or a flush a specific route from the cache.

ping6 <host>

Sends ICMP ECHO_REQUEST packets to IPv6 network hosts.

Syntax

ping6 *host*

Command Mode

Operational mode

Parameters

host	The host being pinged. Can be specified either as name (if DNS is being used on the network) or as an IPv6 address.

Usage Guidelines

The `ping6` command is used to test whether an IPv6 network host is reachable or not.

The `ping6` command uses the ICMP protocol's mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a host or gateway. ECHO_REQUEST datagrams (pings) have an IP and ICMP header, followed by a "`struct timeval`" and then an arbitrary number of pad bytes used to fill out the packet.

To interrupt the ping6 command, press **<Ctrl>+c**.

When using ping6 for fault isolation, it should first be run on the local host, to verify that the local network interface is up and running. Then, hosts and gateways further and further away should be "pinged". Round-trip times and packet loss statistics are computed.

If duplicate packets are received, they are not included in the packet loss calculation, although the round-trip time of these packets is used in calculating the minimum/average/maximum round-trip time numbers. When the specified number of packets have been sent (and received) or if the program is terminated, a brief summary is displayed.

protocols bgp <asn> address-family ipv6-unicast aggregate-address <ipv6net>

Specifies a block of IPv6 addresses to aggregate.

Syntax

set protocols bgp *asn* **address-family ipv6-unicast aggregate-address** *ipv6net* [**summary-only**]

delete protocols bgp *asn* **address-family ipv6-unicast aggregate-address** *ipv6net*

show protocols bgp *asn* **address-family ipv6-unicast aggregate-address** [*ipv6net*]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      address-family {
         ipv6-unicast {
            aggregate-address ipv6net {
               summary-only
            }
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number of the AS in which this router resides.
ipv6net	Mandatory. The IPv6 network from which routes are to be aggregated. The format is ipv6-address/prefix.
summary-only	Specifies that aggregated routes are summarized. These routes will not be announced.

Usage Guidelines

Use the **set** form of this command to specify a contiguous block of IPv6 addresses to aggregate.

Use the **delete** form of this command to delete an aggregate address.

Use the **show** form of this command to view aggregate address configuration settings.

protocols bgp <asn> address-family ipv6-unicast network <ipv6net>

Specifies an IPv6 network to be advertised by the BGP routing process.

Syntax

set protocols bgp *asn* **address-family ipv6-unicast network** *ipv6net*

delete protocols bgp *asn* **address-family ipv6-unicast network** *ipv6net*

show protocols bgp *asn* **address-family ipv6-unicast network**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      address-family {
         ipv6-unicast {
            network ipv6net
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
ipv6net	Mandatory. Multi-node. An IPv6 network in the format *ipv6-address*/*prefix*. You can advertise to multiple networks by creating multiple **network** configuration nodes.

Default

None.

Usage Guidelines

Use this command to advertise IPv6 networks to BGP neighbors.

Use the **set** form of this command to specify an IPv6 network to be announced via BGP.

Use the **delete** form of this command to remove an IPv6 network from the list of networks to be announced by BGP.

Use the **show** form of this command to view BGP network advertising configuration settings.

protocols bgp <asn> address-family ipv6-unicast redistribute connected

Redistributes directly connected routes into BGP.

Syntax

set protocols bgp asn **address-family ipv6-unicast redistribute connected** [**metric** *metric* | **route-map** *map-name*]

delete protocols bgp *asn* **address-family ipv6-unicast redistribute connected** [**metric** | **route-map**]

show protocols bgp *asn* **address-family ipv6-unicast redistribute**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      address-family {
         ipv6-unicast {
            redistribute {
               connected {
                  metric u32
                  route-map text
               }
            }
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
metric	Optional. The metric to be applied to redistributed connected routes.

map-name	Optional. The name of a configured route map to be used for redistributing connected routes.

Default

When this command has not been set, directly connected routes are not distributed into BGP.

Usage Guidelines

Use this command to redistribute directly connected routes into BGP.

Use the **set** form of this command to direct the router to redistribute directly connected routes into BGP.

Use the **delete** form of this command to prevent redistribution of directly connected routes into BGP.

Use the **show** form of this command to view route redistribution configuration settings.

protocols bgp <asn> address-family ipv6-unicast redistribute kernel

Redistributes kernel routes into BGP.

Syntax

set protocols bgp asn **address-family ipv6-unicast redistribute kernel** [**metric** *metric* | **route-map** *map-name*]

delete protocols bgp *asn* **address-family ipv6-unicast redistribute kernel** [**metric** | **route-map**]

show protocols bgp *asn* **address-family ipv6-unicast redistribute**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      address-family {
         ipv6-unicast {
            redistribute {
               kernel {
                  metric u32
                  route-map text
               }
            }
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
metric	Optional. The metric to be applied to redistributed kernel routes.

map-name	Optional. The name of a configured route map to be used for redistributing kernel routes.

Default

When this command has not been set, kernel routes are not distributed into BGP.

Usage Guidelines

Use this command to redistribute kernel routes into BGP.

Use the **set** form of this command to direct the router to redistribute kernel routes into BGP.

Use the **delete** form of this command to prevent redistribution of kernel routes into BGP.

Use the **show** form of this command to view route redistribution configuration settings.

protocols bgp <asn> address-family ipv6-unicast redistribute ospfv3

Redistributes routes learned from OSPFv3 into BGP.

Syntax

set protocols bgp asn **address-family ipv6-unicast redistribute ospfv3** [**metric** *metric* | **route-map** *map-name*]

delete protocols bgp *asn* **address-family ipv6-unicast redistribute ospfv3** [**metric** | **route-map**]

show protocols bgp *asn* **address-family ipv6-unicast redistribute**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      address-family {
         ipv6-unicast {
            redistribute {
               ospfv3 {
                  metric u32
                  route-map text
               }
            }
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
metric	Optional. The metric to be applied to redistributed OSPFv3 routes.

map-name	Optional. The name of a configured route map to be used for redistributing OSPFv3 routes.

Default

When this command has not been set, routes learned from OSPFv3 are not distributed into BGP.

Usage Guidelines

Use this command to redistribute Open Shortest Path First version 3 (OSPFv3) routes into BGP.

Use the **set** form of this command to direct the router to redistribute routes learned from OSPFv3 into BGP.

Use the **delete** form of this command to prevent redistribution of routes learned from OSPFv3 into BGP.

Use the **show** form of this command to view route redistribution configuration settings.

protocols bgp <asn> address-family ipv6-unicast redistribute ripng

Redistributes routes learned from RIPng into BGP.

Syntax

set protocols bgp asn **address-family ipv6-unicast redistribute ripng** [**metric** *metric* | **route-map** *map-name*]

delete protocols bgp *asn* **address-family ipv6-unicast redistribute ripng** [**metric** | **route-map**]

show protocols bgp *asn* **address-family ipv6-unicast redistribute**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      address-family {
         ipv6-unicast {
            redistribute {
               ripng {
                  metric u32
                  route-map text
               }
            }
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
metric	Optional. The metric to be applied to redistributed RIPng routes.

map-name	Optional. The name of a configured route map to be used for redistributing RIPng routes.

Default

When this command has not been set, routes learned from RIPng are not distributed into BGP.

Usage Guidelines

Use this command to redistribute Routing Information Protocol next generation (RIPng) routes into BGP.

Use the **set** form of this command to direct the router to redistribute routes learned from RIPng into BGP.

Use the **delete** form of this command to prevent redistribution of routes learned from RIPng into BGP.

Use the **show** form of this command to view route redistribution configuration settings.

protocols bgp <asn> address-family ipv6-unicast redistribute static

Redistributes static routes into BGP.

Syntax

set protocols bgp asn **address-family ipv6-unicast redistribute static** [**metric** *metric* | **route-map** *map-name*]

delete protocols bgp *asn* **address-family ipv6-unicast redistribute static** [**metric** | **route-map**]

show protocols bgp *asn* **address-family ipv6-unicast redistribute**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      address-family {
         ipv6-unicast {
            redistribute {
               static {
                  metric u32
                  route-map text
               }
            }
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 65535. The subrange 64512 to 65535 is reserved for private autonomous systems.
metric	Optional. The metric to be applied to redistributed static routes.

map-name	Optional. The name of a configured route map to be used for redistributing static routes.

Default

When this command has not been set static routes are not distributed into BGP.

Usage Guidelines

Use this command to redistribute static routes into BGP.

Use the **set** form of this command to direct the router to redistribute static routes into BGP.

Use the **delete** form of this command to prevent redistribution of static routes into BGP.

Use the **show** form of this command to view route redistribution configuration settings.

protocols bgp <asn> neighbor <ipv6> address-family ipv6-unicast distribute-list export <acl-num>

Applies an access list to filter outbound routing updates to this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *ipv6* **address-family ipv6-unicast distribute-list export** *acl-num*

delete protocols bgp *asn* **neighbor** *ipv6* **address-family ipv6-unicast distribute-list**

show protocols bgp *asn* **neighbor** *ipv6* **address-family ipv6-unicast distribute-list**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor ipv6 {
         address-family {
            ipv6-unicast {
               distribute-list {
                  export 1-199
               }
            }
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 4294967294. The subrange 64512 to 65535 is reserved for private autonomous systems.
ipv6	Mandatory. The IPv6 address of a BGP neighbor.
acl-num	Optional. The number of a standard or extended access list. The range for a standard access list is 1 to 99. The range for an extended access list is 100 to 199.

Default

None.

Usage Guidelines

Use the **set** form of this command to apply an access list to filter outbound routing updates to a BGP neighbor.

Use the **delete** form of this command to disable outbound distribute list filtering.

Use the **show** form of this command to view BGP neighbor distribute list configuration settings.

NOTE *A neighbor distribute list cannot be used together with a neighbor prefix list in the same direction. These two lists are mutually exclusive, and only one list may be applied to a given direction.*

protocols bgp <asn> neighbor <ipv6> address-family ipv6-unicast distribute-list import <acl-num>

Applies an access list to filter inbound routing updates from this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *ipv6* **address-family ipv6-unicast distribute-list import** *acl-num*

delete protocols bgp *asn* **neighbor** *ipv6* **address-family ipv6-unicast distribute-list**

show protocols bgp *asn* **neighbor** *ipv6* **address-family ipv6-unicast distribute-list**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor ipv6 {
         address-family {
            ipv6-unicast {
               distribute-list {
                  import 1-199
               }
            }
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 4294967294. The subrange 64512 to 65535 is reserved for private autonomous systems.
ipv6	Mandatory. The IPv6 address of a BGP neighbor.
acl-num	Optional. The number of a standard or extended access list. The range for a standard access list is 1 to 99. The range for an extended access list is 100 to 199.

Default

None.

Usage Guidelines

Use the **set** form of this command to apply an access list to filter inbound routing updates from a BGP neighbor.

Use the **delete** form of this command to disable inbound distribute list filtering.

Use the **show** form of this command to view BGP neighbor distribute list configuration settings.

NOTE *A neighbor distribute list cannot be used together with a neighbor prefix list in the same direction. These two lists are mutually exclusive, and only one list may be applied to the specified direction.*

protocols bgp <asn> neighbor <ipv6> address-family ipv6-unicast prefix-list export <list-name>

Applies an prefix list to filter updates to this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *ipv6* **address-family ipv6-unicast prefix-list export** *list-name*

delete protocols bgp *asn* **neighbor** *ipv6* **address-family ipv6-unicast prefix-list export** *list-name*

show protocols bgp *asn* **neighbor** *ipv6* **address-family ipv6-unicast prefix-list**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor ipv6 {
         address-family {
            ipv6-unicast {
               prefix-list {
                  export text
               }
            }
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 4294967294. The subrange 64512 to 65535 is reserved for private autonomous systems.
ipv6	Mandatory. The IPv6 address of a BGP neighbor.
list-name	Mandatory. Name of a configured prefix list.

Default

None.

Usage Guidelines

Use the **set** form of this command to restrict distribution of outbound BGP neighbor information by filtering with a prefix list.

Use the **delete** form of this command to remove an outbound prefix list filter.

Use the **show** form of this command to view BGP neighbor prefix list configuration settings.

protocols bgp <asn> neighbor <ipv6> address-family ipv6-unicast prefix-list import <list-name>

Applies an prefix list to filter updates from this neighbor.

Syntax

set protocols bgp *asn* **neighbor** *ipv6* **address-family ipv6-unicast prefix-list import** *list-name*

delete protocols bgp *asn* **neighbor** *ipv6* **address-family ipv6-unicast prefix-list import** *list-name*

show protocols bgp *asn* **neighbor** *ipv6* **address-family ipv6-unicast prefix-list**

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   bgp asn {
      neighbor ipv6 {
         address-family {
            ipv6-unicast {
               prefix-list {
                  import text
               }
            }
         }
      }
   }
}
```

Parameters

asn	Mandatory. The number for the AS in which this router resides. The range of values is 1 to 4294967294. The subrange 64512 to 65535 is reserved for private autonomous systems.
ipv6	Mandatory. The IP address of a BGP neighbor.
list-name	Mandatory. Name of a configured prefix list.

Default

None.

Usage Guidelines

Use the **set** form of this command to restrict distribution of inbound BGP neighbor information by filtering with a prefix list.

Use the **delete** form of this command to remove an inbound prefix list filter.

Use the **show** form of this command to view BGP neighbor prefix list configuration settings.

protocols ripng aggregate-address <ipv6net>

Specifies an aggregate RIPng route announcement.

Syntax

set protocols ripng aggregate-address *ipv6net*

delete protocols ripng aggregate-address *ipv6net*

show protocols ripng aggregate-address [*ipv6net*]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   ripng {
      aggregate-address ipv6net
   }
}
```

Parameters

ipv6net	Mandatory. The IPv6 network from which routes are to be aggregated. The format is *ipv6-address*/*prefix*.

Usage Guidelines

Use this command for IPv6 address aggregation.

Use the **set** form of this command to specify a contiguous block of IPv6 addresses to aggregate.

Use the **delete** form of this command to delete an aggregate address.

Use the **show** form of this command to view aggregate address configuration settings.

protocols ripng network <ipv6net>

Specifies a network for the RIPng.

Syntax

set protocols ripng network *ipv6net*

delete protocols ripng network *ipv6net*

show protocols ripng network

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   ripng {
      network: ipv6net
   }
}
```

Parameters

ipv6net	Mandatory. Multi-node. The IPv6 network address of the RIPng network. You can identify more than one RIPng network by creating multiple **protocols ripng network** configuration nodes.

Default

None.

Usage Guidelines

Use this command to identify Routing Information Protocol next generation (RIPng) networks.

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

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

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

protocols ripng route <ipv6net>

Specifies a RIPng static route.

Syntax

set protocols ripng route *ipv6net*

delete protocols ripng route *ipv6net*

show protocols ripng route

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   ripng {
      route ipv6net
   }
}
```

Parameters

ipv6net	Mandatory. The IPv6 network address defining the RIPng static route.

Default

None.

Usage Guidelines

Use this command for setting static routes in Routing Information Protocol next generation (RIPng).

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

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

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

protocols static interface-route6 <subnet> next-hop-interface <ethx>

Allows you to configure the next hop interface for an interface-based IPv6 static route.

Syntax

set protocols static interface-route6 *subnet* **next-hop-interface** *ethx* [**disable** | **distance** *distance*]

delete protocols static interface-route6 *subnet* **next-hop-interface** *ethx* [**disable** | **distance**]

show protocols static interface-route6 *subnet* **next-hop-interface** *ethx* [**disable** | **distance**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   static {
      interface-route6 ipv6net {
         next-hop-interface eth0..eth23 {
            disable
            distance 1-255
         }
      }
   }
}
```

Parameters

subnet	Mandatory. Multi-node. Defines an interface-based static route. The format is a destination subnet of the form IPv6-*address*/*prefix*. You can define multiple interface-based routes by creating multiple **interface-route6** configuration nodes.
ethx	Mandatory. The next hop Ethernet interface.
disable	Disables the interface-based IPv6 static route.

distance	Optional. Defines the next-hop distance for this route. Routes with a smaller distance are selected before those with a larger distance.

Default

None.

Usage Guidelines

Use this command to configure interface-based IPv6 static routes on the system.

Use the **set** form of this command to specify the next hop interface for the route.

Use the **delete** form of this command to remove the next hop interface.

Use the **show** form of this command to view the next hop interface for the route.

protocols static route6 <subnet> blackhole

Allows you to configure a blackhole IPv6 static route.

Syntax

set protocols static route6 *subnet* **blackhole** [**distance** *distance*]

delete protocols static route6 *subnet* **blackhole** [**distance**]

show protocols static route6 *subnet* **blackhole** [**distance**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   static {
      route6 ipv6net {
         blackhole {
            distance 1-255
         }
      }
   }
}
```

Parameters

subnet	Mandatory. Multi-node. Defines an IPv6 static route. The format is a destination subnet of the form IPv6-*address*/*prefix*. You can define multiple static routes by creating multiple **route** configuration nodes.
distance	Optional. Defines the blackhole distance for this route. Routes with a smaller distance will be selected before those with a larger distance.

Default

None.

Usage Guidelines

Use this command to configure a blackhole IPv6 static route. A blackhole route silently discards packets that are matched.

Use the **set** form of this command to specify a blackhole IPv6 static route.

Use the **delete** form of this command to remove a blackhole IPv6 static route.

Use the **show** form of this command to view blackhole IPv6 static route configuration.

protocols static route6 <subnet> next-hop <address>

Allows you to configure the next hop for an IPv6 static route.

Syntax

set protocols static route6 *subnet* **next-hop** *address* [**disable** | **distance** *distance* | **interface** *interface*]

delete protocols static route6 *subnet* **next-hop** *address* [**disable** | **distance** | **interface**]

show protocols static route6 *subnet* **next-hop** *address* [**disable** | **distance** | **interface**]

Command Mode

Configuration mode.

Configuration Statement

```
protocols {
   static {
      route6 ipv6net {
         next-hop ipv6 {
            disable
            distance 1-255
            interface interface
         }
      }
   }
}
```

Parameters

subnet	Mandatory. Multi-node. Defines an IPv6 static route. The format is a destination subnet of the form IPv6-*address*/*prefix*. You can define multiple static routes by creating multiple **route6** configuration nodes.
address	Mandatory. The IPv6 address of the next hop router.
disable	Disable the IPv6 static route.
distance	Optional. Defines the next-hop distance for this route. Routes with a smaller distance will be selected before those with a larger distance.

interface	Optional. The outgoing interface used to reach the next-hop address. This is necessary when the next-hop address is a link-local address (that is, it has a fe80::/64 prefix).

Default

None.

Usage Guidelines

Use this command to configure IPv6 static routes on the system.

Use the **set** form of this command to specify the next hop for the route.

Use the **delete** form of this command to remove the static route next hop.

Use the **show** form of this command to view static route next hop configuration.

show ipv6 bgp

Displays BGP routes.

Syntax

show ipv6 bgp [*ipv6* | *ipv6net* [**longer-prefixes**] | **summary**]

Command Mode

Operational mode.

Parameters

ipv6	Optional. Displays routes for the neighbor at the specified IPv6 address.
ipv6net	Optional. Displays routes for the specified IPv6 network.
longer-prefixes	Optional. Displays any routes more specific than the one specified.
summary	Optional. Shows summary BGP route information.

Default

Displays all BGP routes by default.

Usage Guidelines

Use this command to display the BGP routing table.

show ipv6 bgp community <community>

Displays BGP routes belonging to the specified BGP community.

Syntax

show ipv6 bgp community *community* [exact-match]

Command Mode

Operational mode.

Parameters

community	Mandatory. A BGP community identifier in the form AA:NN (where AA and NN are in the range of 0-65535), one of the well-known BGP communities **local-AS**, **no-export**, or **no-advertise**, or a space-separated list of up to four community identifiers.
exact-match	Optional. Displays only routes that have an exact match.

Default

None.

Usage Guidelines

Use this command to display the BGP routes belonging to up to four BGP communities.

show ipv6 bgp community-list <list-name>

Displays BGP routes permitted by the specified community list.

Syntax

show ipv6 bgp community-list *list-name* [exact-match]

Command Mode

Operational mode.

Parameters

list-name	Mandatory. A preconfigured list of BGP communities.
exact-match	Optional. Displays only route that have an exact match.

Default

None.

Usage Guidelines

Use this command to display the BGP routes permitted by the specified community list.

show ipv6 bgp filter-list <list-num>

Displays routes matching a list of autonomous system paths.

Syntax

show ipv6 bgp filter-list *list-num*

Command Mode

Operational mode.

Parameters

list-num	Mandatory. The number of a preconfigured autonomous system path access list. The range is 1 to 500.

Default

None.

Usage Guidelines

Use this command to filter displayed routes according to preconfigured access list of autonomous system paths.

BGP filter lists are defined using the the **policy as-path-list** command.

show ipv6 bgp neighbors

Displays BGP neighbor information.

Syntax

show ipv6 bgp neighbors [*ipv6*]

Command Mode

Operational mode.

Parameters

ipv6	Optional. The IPv6 address of a BGP neighbor.

Default

Information is shown for all BGP neighbors.

Usage Guidelines

Use this command to display BGP neighbor information.

show ipv6 bgp neighbors <ipv6> advertised-routes

Displays advertised routes for a BGP neighbor.

Syntax

show ipv6 bgp neighbors *ipv6* **advertised-routes**

Command Mode

Operational mode.

Parameters

ipv6	Mandatory. The IPv6 address of a BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display advertised routes for a BGP neighbor.

show ipv6 bgp neighbors <ipv6> received-routes

Displays routes received from a BGP neighbor.

Syntax

show ipv6 bgp neighbors *ipv6* **received-routes**

Command Mode

Operational mode.

Parameters

ipv6	Mandatory. The IPv6 address of a BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display routes (both accepted and rejected) received from a BGP neighbor.

show ipv6 bgp neighbors <ipv6> routes

Displays all received and accepted routes from a BGP neighbor.

Syntax

show ipv6 bgp neighbors *ipv6* **routes**

Command Mode

Operational mode.

Parameters

ipv6	Mandatory. The IPv6 address of a BGP neighbor.

Default

None.

Usage Guidelines

Use this command to display received and accepted routes from a BGP neighbor.

show ipv6 bgp prefix-list <list-name>

Displays BGP routes matching a prefix list.

Syntax

show ipv6 bgp prefix-list *list-name*

Command Mode

Operational mode.

Parameters

list-name	Mandatory. Name of a defined prefix list.

Default

None.

Usage Guidelines

Use this command to display routes that match a preconfigured prefix list.

Prefix lists are configured using the **policy prefix-list** command.

show ipv6 bgp regexp <regexp>

Displays routes matching an AS path regular expression.

Syntax

show ipv6 bgp regexp *regexp*

Command Mode

Operational mode.

Parameters

regexp	Mandatory. A POSIX-style regular expression representing a set of AS paths.

Default

None.

Usage Guidelines

Use this command to display routes matching a regular expression representing an autonomous system (AS) path list.

show ipv6 neighbors

Displays the system's IPv6 ND cache.

Syntax

show ipv6 neighbors

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display the system's IPv6 ND (Neighbor Discovery) cache.

Table 3-1 shows possible ND states.

Table 3-1 ND states

State	Description
incomplete	Address resolution is currently being performed on this neighbor entry. A neighbor solicitation message has been sent but a reply has not yet been received.
reachable	Address resolution has determined that the neighbor is reachable. Positive confirmation has been received and the path to this neighbor is operationable.
stale	More than the configured elapsed time has passed since reachability confirmation was received from this neighbor.
delay	More than the configured elapsed time has passed since reachability confirmation was received from this neighbor. This state allows TCP to confirm the neighbor. If not, a probe should be sent after the next delay time has elapsed.

Table 3-1 ND states

State	Description
probe	A solicitation has been sent and the router is waiting for a response from this neighbor.
failed	Neighbor reachability state detection failed.
noarp	The neighbor entry is valid. There will be no attempts to validate it but it can be removed from the cache when its lifetime expires.
permanent	The neighbor entry is valid indefinitely and should not be cleared from the cache.
none	No state is defined

show ipv6 ripng

Displays information for the Routing Information Protocol next generation (RIPng).

Syntax

show ipv6 ripng [status]

Command Mode

Operational mode.

Parameters

status	Optional. Displays only RIPng protocol status information.

Default

Displays all RIPng protocol information.

Usage Guidelines

Use this command to see information about the Routing Information Protocol next generation (RIPng).

show ipv6 route

Displays IPv6 routes stored in the RIB and FIB.

Syntax

show ipv6 route [*ipv6* | *ipv6net*]

Command Mode

Operational mode.

Parameters

ipv6	Optional. Displays routing information for the specified IPv6 address.
ipv6net	Optional. Displays routing information for the specified IPv6 prefix.

Default

Lists all IPv6 routes in the RIB and FIB.

Usage Guidelines

Use this command to display active IPv6 prefixes stored in the Routing Information Base (RIB), as well as those stored in the Forwarding Information Base (FIB).

The routes shown in the FIB can also be seen using the **show ipv6 route forward** command (see page 86).

show ipv6 route <ipv6net> longer-prefixes

Displays IPv6 prefixes longer than a specified prefix.

Syntax

show ipv6 route *ipv6net* longer-prefixes

Command Mode

Operational mode.

Parameters

ipv6net	Mandatory. Displays all prefixes longer than the specified IPv6 prefix.

Default

None.

Usage Guidelines

Use this command to display all prefixes in the Routing Information Base (RIB) that are longer than a given IPv6 address or prefix.

show ipv6 route bgp

Displays IPv6 BGP routes.

Syntax

show ipv6 route bgp

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display IPv6 BGP routes.

show ipv6 route cache

Displays the kernel IPv6 route cache.

Syntax

show ipv6 route cache [*ipv6net*]

Command Mode

Operational mode.

Parameters

ipv6net	Optional. Displays kernel IPv6 route cache information for the specified route.

Default

Lists routes in the kernel IPv6 route cache.

Usage Guidelines

Use this command to display information about routes stored in the kernel IPv6 route cache. The route cache contains all paths currently in use by the cache. Multiple equal-cost paths are necessary before equal-cost-multi-path (ECMP) routing can be performed.

show ipv6 route connected

Displays IPv6 connected routes.

Syntax

show ipv6 route connected

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display IPv6 routes directly connected to the local system.

show ipv6 route forward

Displays IPv6 routes stored in the FIB.

Syntax

show ipv6 route forward [*ipv6net*]

Command Mode

Operational mode.

Parameters

ipv6net	Optional. Displays information from the kernel forwarding table for the specified IPv6 route.

Default

Lists IPv6 routes in the FIB.

Usage Guidelines

Use this command to display the FIB.

The FIB contains multiple equal-cost paths if existed. Multiple equal-cost paths are necessary before equal-cost multi-path (ECMP) routing or WAN load balancing can be performed.

show ipv6 route kernel

Displays IPv6 kernel routes.

Syntax

show ipv6 route kernel

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display IPv6 kernel routes. Kernel routes are routes that have been added through means other than by using the Vyatta CLI.

show ipv6 route ripng

Displays IPv6 RIPng routes.

Syntax

show ipv6 route ripng

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display IPv6 RIPng routes.

show ipv6 route static

Displays IPv6 static routes.

Syntax

show ipv6 route static

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display IPv6 static routes in the Routing Information Base (RIB).

show ipv6 route summary

Displays IPv6 routes summary.

Syntax

show ipv6 route summary

Command Mode

Operational mode.

Parameters

None.

Default

None.

Usage Guidelines

Use this command to display a summary of the various IPv6 routes by route source.

Examples

Example 3-1 shows a summary of IPv6 routes.

Example 3-1 "show ipv6 route summary": Displaying a summary of IPv6 routes

```
vyatta@vyatta:~$ show ipv6 route summary
Route Source         Routes               FIB
connected            4                    4
static               2                    2
------
Totals               6                    6
[edit]
vyatta@vyatta:~$
```

system ipv6 blacklist

Disables IPv6 on the system.

Syntax

set system ipv6 blacklist

delete system ipv6 blacklist

show system ipv6

Command Mode

Configuration mode.

Configuration Statement

```
system {
   ipv6 {
      blacklist
   }
}
```

Parameters

None.

Default

IPv6 is enabled on the system.

Usage Guidelines

Use this command to disable IPv6 on the system.

Use the **set** form of this command to disable IPv6 on the system.

Use the **delete** form of this command to enable IPv6 on the system.

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

system ipv6 disable

Disables assignment of IPv6 addresses on all interfaces.

Syntax

set system ipv6 disable

delete system ipv6 disable

show system ipv6 disable

Command Mode

Configuration mode.

Configuration Statement

```
system {
   ipv6 {
      disable
   }
}
```

Parameters

None.

Default

IPv6 addresses are assigned on all interfaces.

Usage Guidelines

Use this command to disable the assignment of IPv6 addresses on all interfaces.

Use the **set** form of this command to disable IPv6 address assignment on all interfaces.

Use the **delete** form of this command to enable IPv6 address assignmenton all interfaces.

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

system ipv6 disable-forwarding

Disables IPv6 forwarding on all interfaces.

Syntax

set system ipv6 disable-forwarding

delete system ipv6 disable-forwarding

show system ipv6 disable-forwarding

Command Mode

Configuration mode.

Configuration Statement

```
system {
   ipv6 {
      disable-forwarding
   }
}
```

Parameters

None.

Default

IPv6 packets are forwarded.

Usage Guidelines

Use this command to disable IPv6 forwarding on all interfaces. IPv6 forwarding can also be disabled on a per interface basis using the **interfaces <interface> ipv6 disable-forwarding** command (see page 104).

Use the **set** form of this command to disable IPv6 packet forwarding on all interfaces.

Use the **delete** form of this command to enable IPv6 packet forwarding on all interfaces.

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

system ipv6 strict-dad

Disables IPv6 operation on an interface when DAD fails for a link-local address.

Syntax

set system ipv6 strict-dad

delete system ipv6 strict-dad

show system ipv6 strict-dad

Command Mode

Configuration mode.

Configuration Statement

```
system {
   ipv6 {
      strict-dad
   }
}
```

Parameters

None.

Default

IPv6 operation is not disabled on an interface where Duplicate Address Detection (DAD) fails for a link-local address.

Usage Guidelines

Use this command to disable IPv6 operation on an interface where Duplicate Address Detection fails for a link-local address. Link-local addresses are formed from an interface identifier based on the hardware address, which is supposed to be uniquely assigned. By default the duplicate address is not assigned to the interface but IPv6 continues to operate. This command disables IPv6 on the interface when a duplicate of the link-local address is detected.

Use the **set** form of this command to disable IPv6 operation on an interface when DAD fails for a link-local address.

Use the **delete** form of this command to leave IPv6 operational on an interface when DAD fails for a link-local address.

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

traceroute6 <host>

Displays the route packets take to an IPv6 network host.

Syntax

traceroute6 *host*

Command Mode

Operational mode

Parameters

host	The host that is the destination for the packets. Can be specified either as name (if DNS is being used on the network) or as an IPv6 address.

Usage Guidelines

Traceroute utilizes the IP protocol time to live (“ttl”) field and attempts to elicit an ICMP TIME_EXCEEDED response from each gateway along the path to some host to track the route a set of packets follows.It attempts to trace the route an IP packet would follow to some internet host by launching UDP probe packets with a small ttl then listening for an ICMP “time exceeded” reply from a gateway.

Chapter 4: IPv6 Interface Commands

This chapter describes commands for configuring IPv6 on various interfaces.

This chapter presents the following topics:

- IPv6 Interface Commands

IPv6 Interface Commands

This chapter contains the following commands.

Configuration Commands	
interfaces <interface> ipv6 address	Assigns an IPv6 address to an interface.
interfaces <interface> ipv6 disable-forwarding	Disables IPv6 forwarding on an interface.
interfaces <interface> ipv6 dup-addr-detect-transmits <num>	Specifies the number of times to transmit NS packets as part of the DAD process.
interfaces <interface> ipv6 router-advert	Specifies Router Advertisement (RA) settings on an interface.
interfaces <interface> ipv6 router-advert cur-hop-limit <limit>	Specifies the Hop Count field of the IP header for outgoing (unicast) IP packets.
interfaces <interface> ipv6 router-advert default-lifetime <lifetime>	Specifies the lifetime associated with the default router.
interfaces <interface> ipv6 router-advert link-mtu <mtu>	Specifies the MTU used in router advertisements.
interfaces <interface> ipv6 router-advert managed-flag <state>	Specifies whether hosts use the administered protocol for address autoconfiguration.
interfaces <interface> ipv6 router-advert max-interval <interval>	Specifies the maximum time allowed between sending unsolicited multicast router advertisements.
interfaces <interface> ipv6 router-advert min-interval <interval>	Specifies the minimum time allowed between sending unsolicited multicast router advertisements.
interfaces <interface> ipv6 router-advert other-config-flag <state>	Specifies whether hosts use the administered protocol for autoconfiguration of non-address information.
interfaces <interface> ipv6 router-advert prefix <ipv6net>	Specifies the IPv6 prefix that is to be advertised on the interface.
interfaces <interface> ipv6 router-advert prefix <ipv6net> autonomous-flag <state>	Specifies that the prefix can be used for autonomous address configuration.
interfaces <interface> ipv6 router-advert prefix <ipv6net> on-link-flag <state>	Specifies that the prefix can be used for on-link determination.
interfaces <interface> ipv6 router-advert prefix <ipv6net> preferred-lifetime <lifetime>	Specifies the length of time addresses generated from the prefix via SLAAC remain preferred.
interfaces <interface> ipv6 router-advert prefix <ipv6net> valid-lifetime <lifetime>	Specifies the length of time the prefix remains valid for on-link determination.
interfaces <interface> ipv6 router-advert reachable-time <time>	Specifies the length of time a node assumes a neighbor is reachable after receiving a reachability confirmation.

interfaces <interface> ipv6 router-advert retrans-timer <time>	Specifies the length of time between retransmitted Neighbor Solicitation messages.
interfaces <interface> ipv6 router-advert send-advert <state>	Specifies whether or not router advertisements are sent from this interface.
Operational Commands	
None.	

interfaces <interface> ipv6 address

Assigns an IPv6 address to an interface.

Syntax

set interfaces *interface* ipv6 address [autoconf | eui64 *ipv6prefix*]

delete interfaces *interface* **ipv6 address** [**autoconf | eui64** *ipv6prefix*]

show interfaces *interface* **ipv6 address** [**autoconf | eui64**]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      address {
         autoconf
         eui64 ipv6prefix
      }
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
autoconf	Specifies that the interface will generate an IPv6 address using the SLAAC (Stateless Address Auto-Configuration) protocol as defined in RFC 4862. This parameter is set if the interface is performing a “host” function rather than a “router” function. In addition to this parameter being set, IPv6 forwarding must be disabled either globally or on this interface in order for the system to acquire addresses on this interface using SLAAC. This parameter can be specified in addition to static IPv6, static IPv4, and IPv4 DHCP addresses on the interface.

ipv6prefix	The 64-bit IPv6 address prefix used to configure an IPv6 address using EUI-64 format. The system concatenates this prefix with a 64-bit EUI-64 value that is formed based on the 48-bit MAC address of the interface.

Default

None.

Usage Guidelines

Use this command to specify an IPv6 address on an interface.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.

Interface Type	Syntax	Parameters
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.

Interface Type	Syntax	Parameters
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify an IPv6 address for the interface.

Use the **delete** form of this command to delete an IPv6 address from the interface.

Use the **show** form of this command to view the IPv6 address configuration settings.

interfaces <interface> ipv6 disable-forwarding

Disables IPv6 forwarding on an interface.

Syntax

set interfaces *interface* **ipv6 disable-forwarding**

delete interfaces *interface* **ipv6 disable-forwarding**

show interfaces *interface* **ipv6 disable-forwarding**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      disable-forwarding
   }
}
```

Parameters

None.

Default

IPv6 packets are forwarded.

Usage Guidelines

Use this command to disable IPv6 packet forwarding on an interface. IPv6 forwarding can also be disabled globally (for all interfaces) using the **system ipv6 disable-forwarding** command (see page 93).

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.

Interface Type	Syntax	Parameters
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.

Interface Type	Syntax	Parameters
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to disable IPv6 packet forwarding on an interface.

Use the **delete** form of this command to enable IPv6 packet forwarding on an interface.

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

interfaces <interface> ipv6 dup-addr-detect-transmits <num>

Specifies the number of times to transmit NS packets as part of the DAD process.

Syntax

set interfaces *interface* ipv6 dup-addr-detect-transmits *num*

delete interfaces *interface* **ipv6 dup-addr-detect-transmits**

show interfaces *interface* **ipv6 dup-addr-detect-transmits**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      dup-addr-detect-transmits u32
   }
}
```

Parameters

interface	Mandatory. The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
num	The number of times to transmit Neighbor Solicitation (NS) packets as part of the Duplicate Address Detection (DAD) process. The default is 1.

Default

One NS packet is transmitted as part of the DAD process.

Usage Guidelines

Use this command to specify the number of times to transmit Neighbor Solicitation (NS) packets as part of the Duplicate Address Detection (DAD) process.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.

Interface Type	Syntax	Parameters
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.

Interface Type	Syntax	Parameters
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify the number of times to transmit Neighbor Solicitation (NS) packets as part of the Duplicate Address Detection (DAD) process.

Use the **delete** form of this command to delete the parameter from the interface and use the default value.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert

Specifies Router Advertisement (RA) settings on an interface.

Syntax

set interfaces *interface* ipv6 router-advert

delete interfaces *interface* **ipv6 router-advert**

show interfaces *interface* **ipv6 router-advert**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.

Default

Router Advertisements are not sent on an interface.

Usage Guidelines

Use this command to specify that Router Advertisements (RA) are to be sent out the interface. Router Advertisements are sent out by IPv6 routers in order to advertise their existance to hosts on the network. IPv6 hosts do not send out Router Advertisements. If the **router-advert** node of the configuration tree is missing then Router Advertisements are not sent out. If IPv6 forwarding is disabled either globally (see the **system ipv6**

disable-forwarding command (see page 93)) or on the interface (the **interfaces <interface> ipv6 disable-forwarding** command (see page 104)) then Router Advertisements are not sent out.

Most of the Router Advertisement parameters are required by either the Neighbor Discover (ND) protocol or the Stateless Address Auto-Configuration (SLAAC) protocol. These parameters are used both locally for the IPv6 implementation and become part of the RA messages sent to hosts on the network so that they can be configured appropriately.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.

Interface Type	Syntax	Parameters
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.

Interface Type	Syntax	Parameters
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to create the **router-advert** configuration node and begin to send router advertisements.

Use the **delete** form of this command to remove **router-advert** configuration node and stop sending router advertisements.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert cur-hop-limit <limit>

Specifies the Hop Count field of the IP header for outgoing (unicast) IP packets.

Syntax

set interfaces *interface* ipv6 router-advert cur-hop-limit *limit*

delete interfaces *interface* **ipv6 router-advert cur-hop-limit**

show interfaces *interface* **ipv6 router-advert cur-hop-limit**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         cur-hop-limit [0-255]
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
limit	The value to be placed in the Hop Count field of the IP header for outgoing (unicast) IP packets. The range of values is 0 to 255. The default value is 64. A value of 0 means unspecified by the router.

Default

The Hop Count field is set to 64.

Usage Guidelines

Use this command to specify the value of the Hop Count field of the IP header for outgoing (unicast) IP packets. The value should be set to the current diameter of the Internet. Specifying a value of 0 indicates that it is unspecified by the router.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Interface Type	Syntax	Parameters
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.

Interface Type	Syntax	Parameters
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify the value of the Hop Count field of the IP header for outgoing (unicast) IP packets.

Use the **delete** form of this command to return the Hop Count field to its default value.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert default-lifetime <lifetime>

Specifies the lifetime associated with the default router.

Syntax

set interfaces *interface* ipv6 router-advert default-lifetime *lifetime*

delete interfaces *interface* **ipv6 router-advert default-lifetime**

show interfaces *interface* **ipv6 router-advert default-lifetime**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         default-lifetime [0,(3*max-interval-9000)]
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
lifetime	The lifetime associated with the default router (in seconds). The range of values is 0, and **max-interval** to 9000 (18.2 hours). 0 indicates the router is not a default router. The default value is 3 * **max-interval**.

Default

The default router lifetime is 3 * **max-interval**.

Usage Guidelines

Use this command to specify the lifetime associated with the default router. Specifying a value of 0 indicates that the router is not a default router and should not appear on the default router list. This option applies only to the router's usefulness as a default router; it does not apply to information contained in other message fields or options. Options that need time limits for their information include their own lifetime fields.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.

Interface Type	Syntax	Parameters
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.

Interface Type	Syntax	Parameters
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify the value of the lifetime associated with the default router.

Use the **delete** form of this command to return the lifetime option to its default value.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert link-mtu <mtu>

Specifies the MTU used in router advertisements.

Syntax

set interfaces *interface* ipv6 router-advert link-mtu *mtu*

delete interfaces *interface* **ipv6 router-advert link-mtu**

show interfaces *interface* **ipv6 router-advert link-mtu**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         link-mtu [0, 1280-MaxLinkMTU]
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
mtu	The MTU value to be advertised for the link. The range of values is 0, or 1280 to the maximum MTU for the link (see RFC 2464). The default value is 0 (meaning unspecfied).

Default

The MTU is not specified in the router advertisement message.

Usage Guidelines

Use this command to specify the MTU value that will be specified in router advertisement messages. By default, the MTU is configured on the interface itself and not here. This option is used in cases where the link MTU is not well known. The system will issue a warning, but not fail, if the value set here does not match the MTU configured on the interface.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.

Interface Type	Syntax	Parameters
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.

Interface Type	Syntax	Parameters
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify the MTU to be sent in router advertisement messages

Use the **delete** form of this command to return it to its default and cease sending the MTU in router advertisements.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert managed-flag <state>

Specifies whether hosts use the administered protocol for address autoconfiguration.

Syntax

set interfaces *interface* ipv6 router-advert managed-flag *state*

delete interfaces *interface* **ipv6 router-advert managed-flag**

show interfaces *interface* **ipv6 router-advert managed-flag**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         managed-flag [true | false]
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
state	Allows you to enable the administered protocol for address autoconfiguration. Supported values are as follows: **true**: Hosts use the administered (stateful) protocol for address autoconfiguration in addition to any addresses autoconfigured using stateless address autoconfiguration. **false**: Hosts use only stateless address autoconfiguration The default value is **false**.

Default

Hosts use stateless address autoconfiguration.

Usage Guidelines

Use this command to specify whether or not hosts use the administered (stateful) protocol for address autoconfiguration. When set (true), hosts use the administered (stateful) protocol for address autoconfiguration in addition to any addresses autoconfigured using stateless address autoconfiguration. When not set (false), host only use stateless address autoconfiguration (see RFC 4862).

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.

Interface Type	Syntax	Parameters
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.

Interface Type	Syntax	Parameters
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify whether or not hosts use the administered (stateful) protocol for address autoconfiguration.

Use the **delete** form of this command to return it to its default and only use stateless address autoconfiguration

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert max-interval <interval>

Specifies the maximum time allowed between sending unsolicited multicast router advertisements.

Syntax

set interfaces *interface* ipv6 router-advert max-interval *interval*

delete interfaces *interface* **ipv6 router-advert max-interval**

show interfaces *interface* **ipv6 router-advert max-interval**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         max-interval [4-1600]
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
interval	The maximum time (in seconds) allowed between sending unsolicited multicast router advertisements from the interface. The range of supported values is 4 to 1800. The default value is 600 (10 minutes).

Default

The maximum time between unsolicited multicast router messages is 600 seconds.

Usage Guidelines

Use this command to specify the maximum time (in seconds) allowed between sending unsolicited multicast router advertisements from the interface.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Interface Type	Syntax	Parameters
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.

Interface Type	Syntax	Parameters
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify the maximum time (in seconds) allowed between sending unsolicited multicast router advertisements from the interface.

Use the **delete** form of this command to return it to its default value.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert min-interval <interval>

Specifies the minimum time allowed between sending unsolicited multicast router advertisements.

Syntax

set interfaces *interface* ipv6 router-advert min-interval *interval*

delete interfaces *interface* **ipv6 router-advert min-interval**

show interfaces *interface* **ipv6 router-advert min-interval**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         min-interval [3-(0.75*max-interval)]
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
interval	The minimum time (in seconds) allowed between sending unsolicited multicast router advertisements from the interface. The range of supported values is 3 to 0.75 * max-interval. The default value is 0.33 * max-interval.

Default

The minimum time between unsolicited multicast router messages is 0.33 * **max-interval**.

Usage Guidelines

Use this command to specify the minimum time (in seconds) allowed between sending unsolicited multicast router advertisements from the interface.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Interface Type	Syntax	Parameters
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.

Interface Type	Syntax	Parameters
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify the minimum time (in seconds) allowed between sending unsolicited multicast router advertisements from the interface.

Use the **delete** form of this command to return it to its default value.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert other-config-flag <state>

Specifies whether hosts use the administered protocol for autoconfiguration of non-address information.

Syntax

set interfaces *interface* ipv6 router-advert other-config-flag *state*

delete interfaces *interface* **ipv6 router-advert other-config-flag**

show interfaces *interface* **ipv6 router-advert other-config-flag**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         other-config-flag [true | false]
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
state	Allows you to enable the administered protocol for autoconfiguration of non-address information (see RFC 4862). Supported values are as follows: **true**: Hosts use the administered (stateful) protocol for autoconfiguration of non-address information. **false**: Hosts use stateless autoconfiguration of non-address information. The default value is **false**.

Default

Hosts use stateless autoconfiguration of non-address information.

Usage Guidelines

Use this command to specify whether hosts use the administered (stateful) protocol for autoconfiguration of non-address information.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.

Interface Type	Syntax	Parameters
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.

Interface Type	Syntax	Parameters
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify whether hosts use the administered protocol for autoconfiguration of non-address information.

Use the **delete** form of this command to return it to its default value.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert prefix <ipv6net>

Specifies the IPv6 prefix that is to be advertised on the interface.

Syntax

set interfaces *interface* ipv6 router-advert prefix *ipv6net*

delete interfaces *interface* **ipv6 router-advert prefix** *ipv6net*

show interfaces *interface* **ipv6 router-advert prefix** [*ipv6net*]

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         prefix ipv6net {}
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
ipv6net	Multi-node. The IPv6 prefix to be advertised on the IPv6 interface. It is in the format ipv6-address/prefix.

Default

None.

Usage Guidelines

Use this command to specify the IPv6 prefix that is to be advertised on the interface.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.

Interface Type	Syntax	Parameters
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.

Interface Type	Syntax	Parameters
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify the IPv6 prefix that is to be advertised on the interface.

Use the **delete** form of this command to delete the specified IPv6 prefix.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert prefix <ipv6net> autonomous-flag <state>

Specifies that the prefix can be used for autonomous address configuration.

Syntax

set interfaces *interface* ipv6 router-advert prefix *ipv6net* autonomous-flag *state*

delete interfaces *interface* **ipv6 router-advert prefix** *ipv6net* **autonomous-flag**

show interfaces *interface* **ipv6 router-advert prefix** *ipv6net* **autonomous-flag**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         prefix ipv6net {
            autonomous-flag [true | false]
         }
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
ipv6net	Multi-node. The IPv6 prefix to be advertised on the IPv6 interface. It is in the format ipv6-address/prefix.
state	Allows you to enable the prefix to be used for autonomous address configuration (see RFC 4862). Supported values are as follows: **true**: The prefix can be used for autonomous address configuration. **false**: The prefix cannot be used for autonomous address configuration. The default value is **true**.

Default

The prefix can be used for autonomous address configuration.

Usage Guidelines

Use this command to specify whether or not the prefix can be used for autonomous address configuration.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.

Interface Type	Syntax	Parameters
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.

Interface Type	Syntax	Parameters
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify whether or not the prefix can be used for autonomous address configuration.

Use the **delete** form of this command to return the option to its default state.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert prefix <ipv6net> on-link-flag <state>

Specifies that the prefix can be used for on-link determination.

Syntax

set interfaces *interface* ipv6 router-advert prefix *ipv6net* on-link-flag *state*

delete interfaces *interface* **ipv6 router-advert prefix** *ipv6net* **on-link-flag**

show interfaces *interface* **ipv6 router-advert prefix** *ipv6net* **on-link-flag**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         prefix ipv6net {
            on-link-flag [true | false]
         }
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
ipv6net	Multi-node. The IPv6 prefix to be advertised on the IPv6 interface. It is in the format ipv6-address/prefix.

state	Allows you to enable the prefix to be used for on-link determination (see RFC 4862). Supported values are as follows: **true**: The prefix can be used for on-link determination. **false**: The advertisement makes no statement about on-link or off-link properties of the prefix. For instance, the prefix might be used for address configuration with some addresses belonging to the prefix being on-link and others being off-link. The default value is **true**.

Default

The prefix can be used for on-link determination.

Usage Guidelines

Use this command to specify whether or not the prefix can be used for on-link determination

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.

Interface Type	Syntax	Parameters
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.

Interface Type	Syntax	Parameters
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify whether or not the prefix can be used for on-link determination

Use the **delete** form of this command to return the option to its default state.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert prefix <ipv6net> preferred-lifetime <lifetime>

Specifies the length of time addresses generated from the prefix via SLAAC remain preferred.

Syntax

set interfaces *interface* ipv6 router-advert prefix *ipv6net* preferred-lifetime *lifetime*

delete interfaces *interface* **ipv6 router-advert prefix** *ipv6net* **preferred-lifetime**

show interfaces *interface* **ipv6 router-advert prefix** *ipv6net* **preferred-lifetime**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         prefix ipv6net {
            preferred-lifetime [u32 | infinity]
         }
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
ipv6net	Multi-node. The IPv6 prefix to be advertised on the IPv6 interface. It is in the format ipv6-address/prefix.

lifetime	The length of time (in seconds) (relative to the time the packet is sent) that the addresses generated from the prefix via stateless address autoconfiguration (SLAAC) remain preferred. (see RFC 4862). Supported values are as follows: 1-4294967296: The time in seconds. **infinity**: A symbolic value representing infinity (that is a value of all one bits - 0xffffffff). The default value is 604800 (7 days).

Default

Addresses generated from the prefix via SLAAC remain preferred for 604800 seconds.

Usage Guidelines

Use this command to specify the length of time addresses generated from the prefix via SLAAC remain preferred.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.

Interface Type	Syntax	Parameters
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.

Interface Type	Syntax	Parameters
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify the length of time addresses generated from the prefix via SLAAC remain preferred.

Use the **delete** form of this command to return the option to its default value.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert prefix <ipv6net> valid-lifetime <lifetime>

Specifies the length of time the prefix remains valid for on-link determination.

Syntax

set interfaces *interface* ipv6 router-advert prefix *ipv6net* valid-lifetime *lifetime*

delete interfaces *interface* **ipv6 router-advert prefix** *ipv6net* **valid-lifetime**

show interfaces *interface* **ipv6 router-advert prefix** *ipv6net* **valid-lifetime**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         prefix ipv6net {
            valid-lifetime [u32 | infinity]
         }
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
ipv6net	Multi-node. The IPv6 prefix to be advertised on the IPv6 interface. It is in the format ipv6-address/prefix.

lifetime	The length of time (in seconds) (relative to the time the packet is sent) that the prefix is valid for the purpose of on-link determination. (see RFC 4862). Supported values are as follows: 1-4294967296: The time in seconds. **infinity**: A symbolic value representing infinity (that is a value of all one bits - 0xffffffff). The default value is 2592000 (30 days).

Default

Prefixes remain valid for the purpose of on-link determination for 2592000 seconds.

Usage Guidelines

Use this command to specify the length of time prefixes remain valid for the purpose of on-link determination.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.

Interface Type	Syntax	Parameters
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.

Interface Type	Syntax	Parameters
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify the length of time prefixes remain valid for the purpose of on-link determination.

Use the **delete** form of this command to return the option to its default value.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert reachable-time <time>

Specifies the length of time a node assumes a neighbor is reachable after receiving a reachability confirmation.

Syntax

set interfaces *interface* ipv6 router-advert reachable-time *time*

delete interfaces *interface* **ipv6 router-advert reachable-time**

show interfaces *interface* **ipv6 router-advert reachable-time**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         reachable-time [0-3600000]
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
time	The length of time (in milliseconds) that a node assumes a neighbor is reachable after having received a reachability confirmation. This value is used by address resolution and the Neighbor Unreachability Detection algorithm (see Section 7.3 of RFC 2461). The range of supported values is 0 to 3600000. A value of 0 means unspecified (by this router). The default value is 0 (unspecified by this router).

Default

This option is 0 (unspecified by this router).

Usage Guidelines

Use this command to specify the length of time a node assumes a neighbor is reachable after receiving a reachability confirmation.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.

Interface Type	Syntax	Parameters
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.

Interface Type	Syntax	Parameters
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify the length of time a node assumes a neighbor is reachable after receiving a reachability confirmation.

Use the **delete** form of this command to return the option to its default value.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert retrans-timer <time>

Specifies the length of time between retransmitted Neighbor Solicitation messages.

Syntax

set interfaces *interface* ipv6 router-advert retrans-timer *time*

delete interfaces *interface* **ipv6 router-advert retrans-timer**

show interfaces *interface* **ipv6 router-advert retrans-timer**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         retrans-timer u32
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
time	The length of time (in milliseconds) between retransmitted Neighbor Solicitation messages. This value is used by address resolution and the Neighbor Unreachability Detection algorithm (see Sections 7.2 and 7.3 of RFC 2461). The range of supported values is 0 to 4294967295. A value of 0 means unspecified (by this router). The default value is 0 (unspecified by this router).

Default

This option is 0 (unspecified by this router).

Usage Guidelines

Use this command to specify the length of time between retransmitted Neighbor Solicitation messages.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Interface Type	Syntax	Parameters
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.

Interface Type	Syntax	Parameters
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify the length of time between retransmitted Neighbor Solicitation messages.

Use the **delete** form of this command to return the option to its default value.

Use the **show** form of this command to view the configuration.

interfaces <interface> ipv6 router-advert send-advert <state>

Specifies whether or not router advertisements are sent from this interface.

Syntax

set interfaces *interface* ipv6 router-advert send-advert *state*

delete interfaces *interface* **ipv6 router-advert send-advert**

show interfaces *interface* **ipv6 router-advert send-advert**

Command Mode

Configuration mode.

Configuration Statement

```
interfaces text {
   ipv6 {
      router-advert {
         send-advert [true | false]
      }
   }
}
```

Parameters

interface	The type of interface. For detailed keywords and arguments that can be specified as interface types, see the table in Usage Guidelines below.
state	Allows you to specify whether router advertisements are sent from this interface. Supported values are as follows: **true**: Send router advertisements from this interface. **false**: Do not send router advertisements from this interface (parameters in this sub-tree are still used to configure the local implementation parameters). The default value is **true**.

Default

Router advertisements are sent from this interface.

Usage Guidelines

Use this command to specify whether or not router advertisements are sent from this interface.

The following table shows the syntax and parameters for supported interface types.

Interface Type	Syntax	Parameters
ADSL Bridged Ethernet	adsl *adslx* pvc *pvc-id* bridged-ethernet	*adslx* The name of a Bridged Ethernet- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL Classical IPOA	adsl *adslx* pvc *pvc-id* classical-ipoa	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically.
ADSL PPPoA	adsl *adslx* pvc *pvc-id* pppoa *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The PPPoA unit number. This number must be unique across all PPPoA interfaces. In addition, only one PPPoA instance can be configured on a PVC. PPPoA units range from 0 to 15 and the resulting interfaces are named pppoa0 to pppoa15.
ADSL PPPoE	adsl *adslx* pvc *pvc-id* pppoe *num*	*adslx* The name of a Classical IPoA- encapsulated DSL interface. *pvc-id* The identifier for the PVC. It can either be the *vpi*/*vci* pair or the keyword **auto**, where *vpi* is a Virtual Path Index from 0 to 255, *vci* is a Virtual Circuit Index from from 0 to 65535, and **auto** directs the system to detect the Virtual Path Index and Virtual Circuit Index automatically. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Bonding	bonding *bondx*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**.
Bonding Vif	bonding *bondx* vif *vlan-id*	*bondx* The identifier for the bonding interface. Supported values are **bond0** through **bond99**. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.

Interface Type	Syntax	Parameters
Bridge	bridge *brx*	*brx* The name of a Bridge group. The range is **br0** through **br999**.
Ethernet	ethernet *ethx*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system.
Ethernet PPPoE	ethernet *ethx* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Ethernet Vif	ethernet *ethx* vif *vlan-id*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094.
Ethernet Vif PPPoE	ethernet *ethx* vif *vlan-id* pppoe *num*	*ethx* The name of an Ethernet interface. The range is **eth0** through **eth23**, depending on the physical interfaces available on your system. *vlan-id* The VLAN ID for the vif. The range is 0 to 4094. *num* The name of a defined PPPoE unit. The range is 0 to 15.
Loopback	loopback *lo*	*lo* The name of the loopback interface.
Multilink	multilink *mlx* vif *1*	*mlx* The identifier of the multilink bundle. You can create up to two multilink bundles. Supported values are **ml0** (“em ell zero”) through **ml23** (“em ell twenty-three”). *1* The identifier of the virtual interface. Currently, only one vif is supported for multilink interfaces, and the identifier must be 1. The vif must already have been defined.
OpenVPN	openvpn *vtunx*	*vtunx* The identifier for the OpenVPN interface. This may be **vtun0** to **vtun*x***, where ***x*** is a non-negative integer.
Pseudo-Ethernet	pseudo-ethernet *pethx*	*pethx* The name of a pseudo-Ethernet interface. The range is **peth0** through **peth999**.
Serial Cisco HDLC	serial *wanx* cisco-hdlc vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for Cisco HDLC interfaces, and the identifier must be 1. The vif must already have been defined.
Serial Frame Relay	serial *wanx* frame-relay vif *dlci*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *dlci* The identifier of the virtual interface. For Frame Relay interfaces, this is the DLCI number for the interface. the range is 16 to 991. The vif must already have been defined.

Interface Type	Syntax	Parameters
Serial PPP	serial *wanx* ppp vif *1*	*wanx* The serial interface you are configuring: one of **wan0** through **wan23**. The interface must already have been defined. *1* The identifier of the virtual interface. Currently, only one vif is supported for point-to-point interfaces, and the identifier must be 1. The vif must already have been defined.
Tunnel	tunnel *tunx*	*tunx* An identifier for the tunnel interface you are defining. The range is **tun0** to **tun23**.
Wireless	wireless *wlanx*	*wlanx* The identifier for the wireless interface you are using. This may be **wlan0** to **wlan999**.
Wireless Modem	wirelessmodem *wlmx*	*wlmx* The identifier for the wirelessmodem interface you are using. This may be **wlm0** to **wlm999**.

Use the **set** form of this command to specify whether or not router advertisements are sent from this interface.

Use the **delete** form of this command to return the option to its default value.

Use the **show** form of this command to view the configuration.

Glossary of Acronyms

ACL	access control list
ADSL	Asymmetric Digital Subscriber Line
API	Application Programming Interface
AS	autonomous system
ARP	Address Resolution Protocol
BGP	Border Gateway Protocol
BIOS	Basic Input Output System
BPDU	Bridge Protocol Data Unit
CA	certificate authority
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
DMZ	demilitarized zone
DN	distinguished name
DNS	Domain Name System

DSCP	Differentiated Services Code Point
DSL	Digital Subscriber Line
eBGP	external BGP
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/Ouput
ICMP	Internet Control Message Protocol
IDS	Intrusion Detection System
IEEE	Institute of Electrical and Electronics Engineers
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
ISP	Internet Service Provider
L2TP	Layer 2 Tunneling Protocol
LACP	Link Aggregation Control Protocol
LAN	local area network
LDAP	Lightweight Directory Access Protocol

MAC	medium access control
MIB	Management Information Base
MLPPP	multilink PPP
MRRU	maximum received reconstructed unit
MTU	maximum transmission unit
NAT	Network Address Translation
ND	Neighbor Discovery
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
PKI	Public Key Infrastructure
PPP	Point-to-Point Protocol
PPPoA	PPP over ATM
PPPoE	PPP over Ethernet
PPTP	Point-to-Point Tunneling Protocol
PVC	permanent virtual circuit
QoS	quality of service
RADIUS	Remote Authentication Dial-In User Service
RA	router advertisement
RIB	Routing Information Base
RIP	Routing Information Protocol

RIPng	RIP next generation
RS	router solicitation
Rx	receive
SLAAC	Stateless address auto-configuration
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SONET	Synchronous Optical Network
SSH	Secure Shell
STP	Spanning Tree Protocol
TACACS+	Terminal Access Controller Access Control System Plus
TCP	Transmission Control Protocol
ToS	Type of Service
Tx	transmit
UDP	User Datagram Protocol
vif	virtual interface
VLAN	virtual LAN
VPN	Virtual Private Network
VRRP	Virtual Router Redundancy Protocol
WAN	wide area network