



# Network Stack Commands

This chapter describes the Cisco IOS XR software to configure and monitor features related to IP Version 4 (IPv4) and IP Version 6 (IPv6).

For detailed information about network stack concepts, configuration tasks, and examples, refer to the *IP Addresses and Services Configuration Guide for Cisco 8000 Series Routers*.

- [clear ipv6 neighbors, on page 3](#)
- [clear ipv6 path-mtu, on page 5](#)
- [hw-module profile route scale ipv6-unicast connected-prefix high, on page 6](#)
- [hw-module local-station-mac, on page 7](#)
- [icmp ipv4 rate-limit unreachable, on page 8](#)
- [ipv4 address \(network\), on page 9](#)
- [ipv4 assembler max-packets, on page 11](#)
- [ipv4 assembler timeout, on page 12](#)
- [ipv4 conflict-policy, on page 13](#)
- [ipv4 directed-broadcast, on page 14](#)
- [ipv4 helper-address, on page 15](#)
- [ipv4 mask-reply, on page 17](#)
- [ipv4 mtu, on page 18](#)
- [ipv4 redirects, on page 20](#)
- [ipv4 source-route, on page 21](#)
- [ipv4 tcp-mss-adjust, on page 22](#)
- [ipv4 unnumbered \(point-to-point\), on page 24](#)
- [ipv4 unreachable disable, on page 26](#)
- [ipv4 virtual address, on page 27](#)
- [ipv6 address, on page 29](#)
- [ipv6 address link-local, on page 31](#)
- [ipv6 assembler, on page 33](#)
- [ipv6 conflict-policy, on page 34](#)
- [ipv6 enable , on page 35](#)
- [ipv6 hop-limit, on page 36](#)
- [ipv6 icmp error-interval, on page 37](#)
- [ipv6 mtu, on page 39](#)
- [ipv6 nd dad attempts , on page 41](#)
- [ipv6 nd managed-config-flag, on page 44](#)

- [ipv6 nd ns-interval](#), on page 46
- [ipv6 nd other-config-flag](#) , on page 48
- [ipv6 nd prefix](#), on page 50
- [ipv6 nd ra dns server](#), on page 52
- [ipv6 nd ra dns search list](#), on page 54
- [ipv6 nd ra-interval](#), on page 56
- [ipv6 nd ra-lifetime](#), on page 58
- [ipv6 nd ra specific route](#), on page 59
- [ipv6 nd reachable-time](#), on page 61
- [ipv6 nd redirects](#), on page 63
- [ipv6 nd router-preference](#), on page 64
- [ipv6 nd scavenge-timeout](#), on page 66
- [ipv6 nd suppress-ra](#), on page 67
- [ipv6 neighbor](#), on page 68
- [ipv6 path-mtu enable](#), on page 70
- [ipv6 path-mtu timeout](#), on page 71
- [ipv6 source-route](#), on page 72
- [ipv6 tcp-mss-adjust](#), on page 73
- [ipv6 unreachable disable](#), on page 74
- [ipv6 virtual address](#), on page 75
- [local pool](#), on page 77
- [show arm conflicts](#), on page 80
- [show arm registrations producers](#), on page 82
- [show arm database](#), on page 84
- [show arm router-ids](#), on page 87
- [show arm summary](#), on page 88
- [show arm vrf-summary](#), on page 90
- [show clns statistics](#), on page 91
- [show hw-module local-station-mac](#), on page 93
- [show ipv4 interface](#), on page 94
- [show ipv4 traffic](#), on page 97
- [show ipv6 interface](#) , on page 99
- [show ipv6 neighbors](#), on page 104
- [show ipv6 neighbors summary](#) , on page 108
- [show ipv6 path-mtu](#), on page 109
- [show ipv6 traffic](#), on page 110
- [show linux networking interfaces address-only](#), on page 113
- [show local pool](#), on page 114
- [show mpa client](#), on page 116
- [show mpa groups](#), on page 117
- [show mpa ipv4](#), on page 119
- [show mpa ipv6](#), on page 121
- [show hw-module profile route-scale](#), on page 123

# clear ipv6 neighbors

To delete all entries in the IPv6 neighbor discovery cache, except static entries, use the **clear ipv6 neighbors** command in XR EXEC mode.

**clear ipv6 neighbors** [**location** *node-id*]

<b>Syntax Description</b>	<b>location</b> <i>node-id</i> (Optional) The designated node. The <i>node-id</i> argument is entered in the <i>rack/slot/module</i> notation.						
<b>Command Default</b>	None						
<b>Command Modes</b>	XR EXEC mode						
<b>Command History</b>	<p><b>Release</b></p> <p>Release 7.0.12 This command was introduced.</p>						
<b>Usage Guidelines</b>	If the location option is specified, only the neighbor entries specified in the <b>location</b> <i>node-id</i> keyword and argument are cleared.						
<b>Task ID</b>	<table> <tr> <th>Task ID</th><th>Operations</th></tr> <tr> <td>network</td><td>read, write</td></tr> <tr> <td>IPv6</td><td>execute</td></tr> </table>	Task ID	Operations	network	read, write	IPv6	execute
Task ID	Operations						
network	read, write						
IPv6	execute						

## Examples

In the following example, only the highlighted entry is deleted:

```
RP/0/RP0/CPU0:router# clear ipv6 neighbors ?
location specify a node name

RP/0/RP0/CPU0:router# show ipv6 neighbor

IPv6 Address Age Link-layer Addr State Interface
8888::3 - 1234.2345.9877 REACH HundredGigE0/0/0/0
8888::8 - 1234.2345.9877 REACH HundredGigE0/0/0/0
fe80::205:1ff:fe9f:6400 1335 0005.019f.6400 STALE HundredGigE0/0/0/0
fe80::206:d6ff:fece:3808 1482 0006.d6ce.3808 STALE HundredGigE0/0/0/0
fe80::200:11ff:fe11:1112 1533 0000.1111.1112 STALE HundredGigE0/0/0/2

RP/0/RP0/CPU0:router# clear ipv6 neighbors location 0/2/0
RP/0/RP0/CPU0:router# show ipv6 neighbor

IPv6 Address Age Link-layer Addr State Interface
8888::3 - 1234.2345.9877 REACH HundredGigE0/0/0/0
8888::8 - 1234.2345.9877 REACH HundredGigE0/0/0/0
fe80::205:1ff:fe9f:6400 1387 0005.019f.6400 STALE HundredGigE0/0/0/0
```

```
clear ipv6 neighbors
```

```
fe80::206:d6ff:fece:3808 1534 0006.d6ce.3808 STALE HundredGigE0/0/0/0
```

# clear ipv6 path-mtu

To clear the learnt path maximum transmission unit (MTU) values of IPv6 packets, use the **clear ipv6 path-mtu** command in the XR EXEC mode.

**clear ipv6 path-mtu** [**vrf** {*vrf-name* | **all**}] [**location** *node-id* ] ] [ **address** { *ipv6-address* } ] [**location** *node-id* ] ]

<b>Syntax Description</b>	<b>location</b> <i>node-id</i>	(Optional) The designated node. The node-id argument is entered in the <i>rack/slot/module</i> notation.
	<i>ipv6-address</i>	(Optional) Specific IPv6 address.

**Command Default** None.

**Command Modes** XR EXEC mode

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines** If the location option is specified, only the entries of the node specified in the **location** *node-id* keyword and argument are cleared. Path MTU discovery for IPv6 packets is supported only for applications using TCP and Ping protocol.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write
	network	read, write
	config-services	read, write

**Examples** This example shows how to clear learnt values of path MTU values of IPv6 packets:

```
Router# clear ipv6 path-mtu vrf all location all
```

# hw-module profile route scale ipv6-unicast connected-prefix high

To enable the IPv6 prefix scale expansion for inserting /126 and /127 IPv6 prefixes in the CEM memory instead of the LPM memory, and increase the scalability of these prefixes, use the **hw-module profile route scale ipv6-unicast connected-prefix high** command in System Admin Config mode.

Use the **no** form of the command to disable the feature.

**hw-module profile route scale ipv6-unicast connected-prefix high**

<b>Syntax Description</b>	This command has no keywords or arguments.
---------------------------	--

<b>Command Default</b>	This command is disabled by default.
------------------------	--------------------------------------

<b>Command Modes</b>	System Admin Config mode
----------------------	--------------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.5.3	This command was introduced.

<b>Usage Guidelines</b>	The chassis must be reloaded for the <b>hw-module</b> command to be functional.
-------------------------	---

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write

```
Router# configure
Router(config)# hw-module profile route scale ipv6-unicast connected-prefix high
Tue Aug 23 18:26:42.473 UTC
In order to activate/deactivate this Route Scale IPv6-ucast connected-prefix profile, you
must manually reload the chassis/all line cards
Router(config)# commit
Tue Aug 23 18:26:57.018 UTC
Router(config)# end
```

After configuring, you must reload the router for the feature to take effect.

# hw-module local-station-mac

To configure the local station MAC address for the router, use the **hw-module local-station-mac** command in the configuration mode.

**hw-module local-station-mac** *mac-address*

<b>Syntax Description</b>	<i>mac-address</i> Specify the 12-digit local station MAC address for router.
---------------------------	---

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	XR Config mode
----------------------	----------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.9.1	This command was introduced.

<b>Usage Guidelines</b>	No specific guidelines impact the use of this command.
-------------------------	--

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	network	read, write
	config-services	read, write

## Examples

This example configures the local station MAC address for the router:

```
Router# config
Router(config)# hw-module local-station-mac B03F.C98C.B948
Router(config)# commit
```

# icmp ipv4 rate-limit unreachable

To limit the rate that IPv4 Internet Control Message Protocol (ICMP) destination unreachable messages are generated, use the **icmp ipv4 rate-limit unreachable** command in XR Config mode. To remove the rate limit, use the **no** form of this command.

**icmp ipv4 rate-limit unreachable** [**DF**] *milliseconds*  
**no icmp ipv4 rate-limit unreachable** [**DF**] *milliseconds*

<b>Syntax Description</b>	<b>DF</b>	(Optional) Limits the rate at which ICMP destination unreachable messages are sent when code 4 fragmentation is needed and data fragmentation is (DF) set, as specified in the IP header of the ICMP destination unreachable message.
	<i>milliseconds</i>	Time period (in milliseconds) between the sending of ICMP destination unreachable messages. Range is 1 to 4294967295.

**Command Default** The default value is one ICMP destination unreachable message every 500 milliseconds.

**Command Modes** XR Config mode

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines** The Cisco IOS XR software maintains two timers: one for general destination unreachable messages and one for DF destination unreachable messages. Both share the same time limits and defaults. If the **DF** option is not configured, the **icmp ipv4 rate-limit unreachable** command sets the time values for DF destination unreachable messages. If the **DF** option is configured, its time values remain independent from those of general destination unreachable messages.

<b>Task ID</b>	<b>Task ID Operations</b>	
	ipv4	read, write
	network	read, write

**Examples** The following example shows how to set the time interval for the ICMP destination unreachable message to be generated at a minimum interval of 10 ms:

```
RP/0/RP0/CPU0:router(config)# icmp ipv4 rate-limit unreachable 10
```



# ipv4 address (network)

To set a primary or secondary IPv4 address for an interface, use the **ipv4 address** command in interface configuration mode. To remove an IPv4 address, use the **no** form of this command.

**ipv4 address** *ipv4-address mask* [**secondary**] [**route-tag** *route-tag value*]  
**no ipv4 address** *ipv4-address mask* [**secondary**] [**route-tag** *route-tag value*]

Syntax Description	<b>ipv4-address</b>	IPv4 address.
	<i>mask</i>	Mask for the associated IP subnet. The network mask can be specified in either of two ways: <ul style="list-style-type: none"> <li>The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means the corresponding address bit belongs to the network address.</li> <li>The network mask can be indicated as a slash (/) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address.</li> </ul>
	<b>secondary</b>	(Optional) Specifies that the configured address is a secondary IPv4 address. If this keyword is omitted, the configured address is the primary IPv4 address.
	<b>route-tag</b>	(Optional) Specifies that the configured address has a route tag to be associated with it.
	<i>route-tag value</i>	(Optional) Value of the route tag. Range is 1 to 4294967295.

**Command Default** No IPv4 address is defined for the interface.

**Command Modes** Interface configuration

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** An interface can have one primary IPv4 address and multiple secondary IPv4 addresses. Packets generated by the software always use the primary IPv4 address. Therefore, all networking devices on a segment should share the same primary network number.



**Note** The same IPv4 address configured on two different interfaces causes an error message to display that indicates the conflict. The interface located in the highest rack, slot, module, instance, and port is disabled.

Hosts can determine subnet masks using the IPv4 Internet Control Message Protocol (ICMP) mask request message. Networking devices respond to this request with an ICMP mask reply message.

You can disable IPv4 processing on a particular interface by removing its IPv4 address with the **no ipv4 address** command. If the software detects another host using one of its IPv4 addresses, it will display an error message on the console.

The optional **secondary** keyword allows you to specify an unlimited number of secondary addresses. Secondary addresses are treated like primary addresses, except that the system never generates datagrams other than routing updates with secondary source addresses. IPv4 broadcasts and Address Resolution Protocol (ARP) requests are handled properly, as are interface routes in the IP routing table.

Secondary IPv4 addresses can be used in a variety of situations. The following are the most common applications:

- There may not be enough host addresses for a particular network segment. For example, your subnetting allows up to 254 hosts per logical subnet, but on one physical subnet you need to have 300 host addresses. Using secondary IPv4 addresses on the networking devices allows you to have two logical subnets using one physical subnet.
- Many older networks were built using Level 2 bridges. The judicious use of secondary addresses can aid in the transition to a subnetted, router-based network. Routers on an older, bridged segment can be easily made aware that there are many subnets on that segment.

The route-tag feature attaches a tag to all IPv4 addresses. The tag is propagated from the Management Agents (MA) to the Address Repository Managers (RPM) to routing protocols, thus enabling the user to control the redistribution of connected routes by looking at the route tags via RPL scripts.

## Task ID

### Task ID Operations

ipv4	read, write
------	----------------

network	read, write
---------	----------------

## Examples

The following example shows how to set 192.168.1.27 as the primary address and 192.168.7.17 and 192.168.8.17 as the secondary addresses on hundredGigE interface 0/0/0/1:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/0/0/1
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.1.27 255.255.255.0
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.7.17 255.255.255.0 secondary
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.8.17 255.255.255.0 secondary
```

# ipv4 assembler max-packets

To configure the maximum number of packets that are allowed in assembly queues, use the **ipv4 assembler max-packets** command in XR Config mode. To disable this feature, use the **no** form of this command.

**ipv4 assembler max-packets** *percentage value*  
**no ipv4 assembler max-packets** *percentage value*

<b>Syntax Description</b>	<i>percentage value</i> Percentage of total packets available in the system. The range is from 1 to 50.	
<b>Command Default</b>	None	
<b>Command Modes</b>	XR Config mode	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	No specific guidelines impact the use of this command.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv4	read, write
	network	read, write
<b>Examples</b>	The following example shows how to configure the maximum number of packets for the assembly queue:  Router(config)# <b>ipv4 assembler max-packets 35</b>	

# ipv4 assembler timeout

To configure the number of seconds an assembly queue can hold before a timeout occurs, use the **ipv4 assembler timeout** command in XR Config mode. To disable this feature, use the **no** form of this command.

**ipv4 assembler timeout** *seconds*  
**no ipv4 assembler timeout** *seconds*

<b>Syntax Description</b>	<i>seconds</i> Number of seconds an assembly queue can hold before a timeout occurs. The range is from 1 to 120.	
<b>Command Default</b>	None	
<b>Command Modes</b>	XR Config mode	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	No specific guidelines impact the use of this command.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv4	read, write
	network	read, write
<b>Examples</b>	The following example shows how to configure an assembly queue before a timeout occurs:	

```
RP/0/RP0/CPU0:router(config)# ipv4 assembler timeout 88
```

# ipv4 conflict-policy

To enable IP Address Repository Manager (IPARM) conflict resolution, use the **ipv4 conflict-policy** command in XR Config mode. To disable the IPARM conflict resolution, use the **no** form of the command.

**ipv4 conflict-policy** {highest-ip | longest-prefix | static}  
**no ipv4 conflict-policy** {highest-ip | longest-prefix | static}

Syntax Description	<b>highest-ip</b>	Keeps the highest ip address in the conflict set.
	<b>longest-prefix</b>	Keeps the longest prefix match in the conflict set.
	<b>static</b>	Keeps the existing interface running across new address configurations.
Command Default	The precedence rule adopted is loopback > physical > other virtual interfaces. Within virtual interfaces, there is an alphabetical preference, for example, loopback1 > loopback2 > tunnel. Among physical interfaces, the lower rack or slot takes control.	
Command Modes	XR Config mode	
Command History	Release	Modification
	Release 7.0.12	This command was introduced.
Usage Guidelines	Use <b>ipv4 conflict-policy</b> command to set an IPARM policy that resolves a conflict in the configured addresses. The policy tells IPARM what address to select from the addresses in conflict. The policy then forces the address in conflict to become inactive.	
Task ID	Task ID	Operations
	ipv4	read, write
	ip-services	read, write
Examples	The following example shows how to enable the static policy for conflict resolution:  RP/0/RP0/CPU0:router(config)# <b>ipv4 conflict-policy static</b>	

# ipv4 directed-broadcast

To enable forwarding of IPv4 directed broadcasts on an interface, use the **ipv4 directed-broadcast** command in interface configuration mode. To disable forwarding of IPv4 directed broadcast on an interface, use the **no** form of this command.

**ipv4 directed-broadcast**  
**no ipv4 directed-broadcast**

**Syntax Description** This command has no keywords or arguments.

**Command Default** By default, directed broadcasts are dropped.

**Command Modes** Interface configuration

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** A directed broadcast is a packet sent to a specific network. IPv4 directed broadcasts are dropped and not forwarded. Dropping IPv4 directed broadcasts makes routers less susceptible to denial-of-service (DoS) attacks.

Task ID	Task ID	Operations
	ipv4	read, write
	network	read, write

**Examples** The following example shows how to enable the forwarding of IPv4 directed broadcasts on interface 0/1/1/0:

```
RP/0/RP0/CPU0:router(config)# interface 0/1/1/0
RP/0/RP0/CPU0:router(config-if)# ipv4 directed-broadcast
```

# ipv4 helper-address

To configure the address to which the software forwards User Datagram Protocol (UDP) broadcasts, including BOOTP, received on an interface, use the **ipv4 helper-address** command in interface configuration mode. To remove an IPv4 helper address, use the **no** form of this command.

```
{ipv4 helper-address [vrf vrf-name][destination-address]}
{no ipv4 helper-address [vrf vrf-name][destination-address]}
```

<b>Syntax Description</b>	<b>vrf</b>	(Optional) Displays VPN routing and forwarding (VRF) instance information.
	<i>vrf-name</i>	(Optional) Name of a VRF.
	<i>destination-address</i>	Destination broadcast or host address to be used when UDP broadcasts are forwarded. There can be more than one helper address per interface.

**Command Default** IPv4 helper addresses are disabled. Default VRF is assumed if the VRF is not specified.

**Command Modes** Interface configuration

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use this command with the **forward-protocol udp** command in `mode`, which specifies by port number the broadcast packets that are forwarded. UDP is enabled by default for well-known ports. The **ipv4 helper-address** command specifies the destination to which the UDP packets are forwarded.

One common application that requires IPv4 helper addresses is Dynamic Host Configuration Protocol (DHCP), which is defined in RFC 1531. DHCP protocol information is carried inside of BOOTP packets. To enable BOOTP broadcast forwarding for a set of clients, configure an IPv4 helper address on the networking device interface physically closest to the client. The IPv4 helper address should specify the address of the DHCP server. If you have multiple servers, you can configure one IPv4 helper address for each server. Because BOOTP packets are forwarded by default, DHCP information can now be forwarded by the networking device. The DHCP server now receives broadcasts from the DHCP clients.

A DHCP relay profile must be configured to perform DHCP Relay. The **ip helper-address** command is used to forward broadcast UDP (non-DHCP) packets.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv4	read, write
	network	read, write

---

**Examples**

The following example shows how to specify that all UDP broadcast packets received on HundredGigEinterface 0/1/0/0 are forwarded to 192.168.1.0:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/0  
RP/0/RP0/CPU0:router(config-if)# ipv4 helper-address 192.168.1.0
```



# ipv4 mask-reply

To enable the software to respond to IPv4 Internet Control Message Protocol (ICMP) mask requests by sending ICMP mask reply messages, use the **ipv4 mask-reply** command in interface configuration mode. To restore the default, use the **no** form of this command.

**ipv4 mask-reply**  
**no ipv4 mask-reply**

**Syntax Description** This command has no keywords or arguments.

**Command Default** IPv4 mask replies are not sent.

**Command Modes** Interface configuration

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** This command enables the software to respond to IPv4 ICMP mask requests by sending ICMP mask reply messages.

Task ID	Task ID	Operations
	ipv4	read, write
	network	read, write

**Examples** The following example enables the sending of ICMP mask reply messages on HundredGigEinterface 0/1/0/0:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/0
RP/0/RP0/CPU0:router(config-if)# ipv4 mask-reply
```

# ipv4 mtu

To set the maximum transmission unit (MTU) size of IPv4 packets sent on an interface, use the **ipv4 mtu** command in interface configuration mode. To restore the default MTU size, use the **no** form of this command.

**ipv4 mtu** *bytes*

**no ipv4 mtu**

<b>Syntax Description</b>	<i>bytes</i> MTU in bytes. Range is 68 to 65535 bytes for IPv4 packets. The maximum MTU size that can be set on an interface depends on the interface medium.				
<b>Command Default</b>	If no MTU size is configured for IPv4 packets sent on an interface, the interface derives the MTU from the Layer 2 MTU.				
<b>Command Modes</b>	Interface configuration				
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				
<b>Usage Guidelines</b>	<p>The router punts the packets that needs fragmentation; whereas the software path drops the subscriber traffic that needs fragmentation.</p> <p>The maximum MTU size that can be set on an interface depends on the interface medium. If the Layer 2 MTU is smaller than the Layer 3 MTU, the Cisco IOS XR software uses the Layer 2 MTU value for the Layer 3 MTU. Conversely, if the Layer 3 MTU is smaller than the Layer 2 MTU, the software uses Layer 3 MTU value. In other words the Cisco IOS XR software uses the lower of the two values for the MTU.</p> <p>All devices on a physical medium must have the same protocol MTU to operate.</p>				



**Note** Changing the MTU value (with the **mtu** interface configuration command) can affect the IPv4 MTU value. If the current IPv4 MTU value is the same as the MTU value, and you change the MTU value, the IPv4 MTU value will be modified automatically to match the new MTU. However, the reverse is not true; changing the IPv4 MTU value has no effect on the value for the **mtu** command.

Task ID	Task ID	Operations
	ipv4	read, write
	network	read, write
	config-services	read, write

---

**Examples**

This example shows how to set the maximum IPv4 packet size for HundredGigE interface 0/0/0/1 to 1500 bytes:

```
RP/0/(config)# interface HundredGigE0/0/0/1  
RP/0/(config-if)# ipv4 mtu 1500
```

# ipv4 redirects

To enable the sending of IPv4 Internet Control Message Protocol (ICMP) redirect messages if the software is forced to resend a packet through the same interface on which it was received, use the **ipv4 redirects** command in interface configuration mode. To restore the default, use the **no** form of this command.

**ipv4 redirects**  
**no ipv4 redirects**

<b>Syntax Description</b>	This command has no keywords or arguments.				
<b>Command Default</b>	ICMP redirect messages are disabled by default on the interface.				
<b>Command Modes</b>	Interface configuration				
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				
<b>Usage Guidelines</b>	ICMP redirect messages are disabled by default on the interface.				

Task ID	Task ID	Operations
	ipv4	read, write
	network	read, write

## Examples

The following example shows how to disable the sending of ICMP IPv4 redirect messages on &HundredGigE interface 0/1/1/0:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/1/0
RP/0/RP0/CPU0:router(config-if)# ipv4 redirects
```

# ipv4 source-route

To allow the processing of any IPv4 datagrams containing a source-route header option, use the **ipv4 source-route** command in XR EXEC mode. To have the software discard any IP datagram that contains a source-route option, use the **no** form of this command.

**ipv4 source-route**  
**no ipv4 source-route**

<b>Syntax Description</b>	This command has no keywords or arguments.	
<b>Command Default</b>	The software discards any IPv4 datagrams containing a source-route header option.	
<b>Command Modes</b>	XR EXEC mode	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	By default, any IPv4 datagram which contains a source-route header option is discarded.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv4	read, write
	network	read, write
<b>Examples</b>	The following example shows how to allow the processing of any IPv4 datagrams containing a source-route header option:	

```
RP/0/RP0/CPU0:router(config)# ipv4 source-route
```

# ipv4 tcp-mss-adjust

To enable the transit traffic of TCP flows to be a Maximum Segment Size (MSS) below the GRE tunnel interface or VLAN sub-interface MTU so that traffic fragmentation is prevented when a session is established for IPv4 packets, use the **ipv4 tcp-mss-adjust** command in the interface configuration submode. To disable the transit traffic of TCP flows to be a Maximum Segment Size (MSS) below the GRE tunnel interface or VLAN sub-interface MTU, use the **no** form of this command.

**ipv4 tcp-mss-adjust enable**  
**no ipv4 tcp-mss-adjust enable**

<b>Syntax Description</b>	<b>enable</b> Enables Maximum Segment Size (MSS) adjustment for tcp flows on the interface.
---------------------------	---

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	Interface Configuration
----------------------	-------------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

## Usage Guidelines

Task ID	Task ID	Operation
	mpls-te	read, write
	ipv4	read, write
	network	read, write
	acl	read, write

## Example

This example shows how to enable the transit traffic of TCP flows for IPv4 packets using the **ipv4 tcp-mss-adjust** command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/0/0/4.20
RP/0/RP0/CPU0:router(config-if)# ipv4 tcp-mss-adjust enable
```

**Related Commands**

Command	Description
<a href="#">ipv6 tcp-mss-adjust, on page 73</a>	Enables the transit traffic of TCP flows to be a Maximum Segment Size (MSS) below the GRE tunnel interface or VLAN sub-interface MTU so that traffic fragmentation is prevented when a session is established for IPv6 packets.

# ipv4 unnumbered (point-to-point)

To enable IPv4 processing on a point-to-point interface without assigning an explicit IPv4 address to that interface, use the **ipv4 unnumbered** command in an interface configuration mode. To disable this feature, use the **no** form of this command.

**ipv4 unnumbered** *interface-type interface-instance*  
**no ipv4 unnumbered** *interface-type interface-instance*

## Syntax Description

*interface-type* Interface type. For more information, use the question mark (?) online help function.

*interface-instance* Either a physical interface instance or a virtual interface instance as follows:

- Physical interface instance. Naming notation is *rack/slot/module/port* and a slash between values is required as part of the notation.
  - *rack*: Chassis number of the rack.
  - *slot*: Physical slot number of the modular services card or line card.
  - *module*: Module number. A physical layer interface module (PLIM) is always 0.
  - *port*: Physical port number of the interface.

**Note** In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RSP0 ) and the module is CPU0. Example: interface MgmtEth0/RSP0 /CPU0/0.

- Virtual interface instance. Number range varies depending on interface type.

For more information about the syntax for the router, use the question mark (?) online help function.

## Command Default

IPv4 processing on a point-to-point interface is disabled unless an IPv4 address is assigned explicitly to that interface.

## Command Modes

Interface configuration

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

## Usage Guidelines

For release Release 4.2.0 onward, to enter the dynamic template configuration mode, run the **dynamic-template** command in the XR Config mode.

Whenever the unnumbered interface generates a packet (for example, for a routing update), it uses the address of the specified interface as the source address of the IPv4 packet. It also uses the IPv4 address of the specified interface in determining which routing processes are sending updates over the unnumbered interface.

Restrictions include the following:



- You cannot use the **ping** EXEC command to determine whether the interface is up because the interface has no address. Simple Network Management Protocol (SNMP) can be used to remotely monitor interface status.

The interface you specify by the *interface-type* and *interface-number* arguments must be enabled (listed as “up” in the **show interfaces** command display).

Task ID	Task ID	Operations
	ipv4	read, write
	network	read, write
	config-services	read, write

### Examples

This example shows how the HundredGigE interface 0/0/0/1 is assigned the loopback interface address 5:

```
RP/0/RP0/CPU0:router(config)# interface loopback 5
RP/0/RP0/CPU0:router(config-if)# ipv4 address 192.168.6.6 255.255.255.0
RP/0/RP0/CPU0:router(config)# interface HundredGigE 0/0/0/1
RP/0/RP0/CPU0:router(config-if)# ipv4 unnumbered loopback 5
```

# ipv4 unreachable disable

To disable the generation of IPv4 Internet Control Message Protocol (ICMP) unreachable messages, use the **ipv4 unreachable disable** command in an interface configuration mode. To re-enable the generation of ICMP unreachable messages, use the **no** form of this command.

**ipv4 unreachable disable**  
**no ipv4 unreachable disable**

<b>Syntax Description</b>	This command has no keywords or arguments.	
<b>Command Default</b>	IPv4 ICMP unreachable messages are generated.	
<b>Command Modes</b>	Interface configuration	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	If the software receives a nonbroadcast packet destined for itself that uses a protocol it does not recognize, it sends an ICMP protocol unreachable message to the source.	
	If the software receives a datagram that it cannot deliver to its ultimate destination because it knows of no route to the destination address, it replies to the originator of that datagram with an ICMP host unreachable message.	
	This command affects a number of ICMP unreachable messages.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv4	read, write
	network	read, write
	config-services	read, write
<b>Examples</b>	This example shows how to disable the generation of ICMP unreachable messages on HundredGigE interface 0/0/0/1:	
	<pre>RP/0/RP0/CPU0:router(config)# interface HundredGigE0/0/0/1 RP/0/RP0/CPU0:router(config-if)# ipv4 unreachable disable</pre>	

# ipv4 virtual address

To define an IPv4 virtual address for a network of management Ethernet interfaces, use the **ipv4 virtual interface** command in XR Config mode. To remove an IPv4 virtual address from the configuration, use the **no** form of this command.

```
ipv4 virtual address {[vrf vrf-name] ipv4-address/mask | use-as-src-addr}
no ipv4 virtual address {[vrf vrf-name] ipv4-address/mask | use-as-src-addr}
```

Syntax Description	
<b>vrf</b> <i>vrf-name</i>	(Optional) Configures the virtual address on a per VPN routing and forwarding (VRF) basis for the management interfaces. The <i>vrf-name</i> argument specifies the name of the VRF.
<i>ipv4 address</i>	Virtual IPv4 address and the mask that is to be unconfigured.
<i>mask</i>	Mask for the associated IP subnet. The network mask can be specified in either of two ways: <ul style="list-style-type: none"> <li>The network mask can be a four-part dotted-decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means the corresponding address bit belongs to the network address.</li> <li>The network mask can be indicated as a slash ( / ) and number. For example, /8 indicates that the first 8 bits of the mask are ones, and the corresponding bits of the address are network address. A slash between numbers is required as part of the notation.</li> </ul>
<b>use-as-src-addr</b>	Enables the virtual address to be used as the default SRC address on sourced packets.

**Command Default** No IPv4 virtual address is defined for the configuration.

**Command Modes** XR Config mode

Command History	Release	Modification
	Release 7.5.2	This release supports virtual addresses for the hosted Linux networking stack.
	Release 7.0.12	This command was introduced.

**Usage Guidelines**

Configuring an IPv4 virtual address enables you to access the router from a single virtual address with a management network. An IPv4 virtual address persists across route processor (RP) failover situations.

Configuring an IPv4 virtual address enables you to access a dual RP router from a single address without prior knowledge of which RP is active. An IPv4 virtual address persists across RP failovers. For this to happen, the virtual IPv4 address must share a common IPv4 subnet with a Management Ethernet interface on both RPs.

If you disable the **ipv4 virtual address** command with the **vrf** keyword, the virtual IP address is unconfigured for the corresponding VRF or for the default if no VRF is specified. This results in the removal of the entry for the virtual IP address in the VRF table and in the ARP cache.

The default VRF is chosen when no VRF is specified. The virtual IP address is activated on a management interface that is attached to a default VRF.

The **use-as-src-addr** keyword eliminates the need for configuring a loopback interface as the source interface (that is, update source) for management applications. When an update source is not configured, management applications allow the transport processes (TCP, UDP, raw\_ip) to pick a suitable source address. The transport processes, in turn, consult the FIB to do so. If a Management Ethernet's IP address is picked as the source address and if the **use-as-src-addr keyword** is configured, then the transport processes replace the Management Ethernet's IP address with a relevant virtual IP address. This functionality works across RP switchovers.

Cisco IOS XR Software Release 7.5.2 and later also supports virtual addresses for the hosted Linux networking stack.

## Task ID

### Task ID Operations

ipv4	read, write
------	----------------

network	read, write
---------	----------------

## Examples

The following example shows how to define an IPv4 virtual address:

```
Router(config)# ipv4 virtual address 10.3.32.154/8
```

The following example show how to configure the virtual IP addresses for management interfaces on a per VRF basis:

```
Router(config)# ipv4 virtual address vrf ppp 10.26.3.4/16
```

# ipv6 address

To configure an IPv6 address for an interface and enable IPv6 processing on the interface using an EUI-64 interface ID in the low-order 64 bits of the address, use the **ipv6 address** command in interface configuration mode. To remove the address from the interface, use the **no** form of this command.

**ipv6 address** *ipv6-prefix/prefix-length* [**eui-64**] [**route-tag** *route-tag value*]  
**no ipv6 address** *ipv6-prefix/prefix-length* [**eui-64**] [**route-tag** *route-tag value*]

<b>Syntax Description</b>	<i>ipv6-prefix</i>	The IPv6 network assigned to the interface.  This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
	<i>/prefix-length</i>	The length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address compose the prefix (the network portion of the address). A slash (/) must precede the decimal value.
	<b>eui-64</b>	(Optional) Specifies an interface ID in the low-order 64 bits of the IPv6 address.
	<b>route-tag</b>	(Optional) Specifies that the configured address has a route tag to be associated with it.
	<i>route-tag value</i>	(Optional) Value of the route tag. Range is 1 to 4294967295.
<b>Command Default</b>	No IPv6 address is defined for the interface.	
<b>Command Modes</b>	Interface configuration	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	If the value specified for the <i>/prefix-length</i> argument is greater than 64 bits, the prefix bits have precedence over the interface ID.	
	Using the <b>no ipv6 address</b> command without arguments removes all manually configured IPv6 addresses from an interface.	
	If the Cisco IOS XR software detects another host using one of its IPv6 addresses, it displays an error message on the console.	
	The route-tag feature attaches a tag to all IPv6 addresses. The tag is propagated from the Management Agents (MA) to the Address Repository Managers (RPM) to routing protocols, thus enabling the user to control the redistribution of connected routes by looking at the route tags via RPL scripts.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write

---

**Task ID   Operations**

---

network   read,  
            write

---



---

**Examples**

The following example assigns IPv6 address 2001:0DB8:0:1::/64 to HundredGigE interface 0/0/0/1 and specifies an EUI-64 interface ID in the low-order 64 bits of the address:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/0/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 address 2001:0DB8:0:1::/64 eui-64
```

# ipv6 address link-local

To configure an IPv6 link-local address for an interface and enable IPv6 processing on the interface, use the **ipv6 address link-local** command in interface configuration mode. The *ipv6-address* value specified with this command overrides the link-local address that is automatically generated for the interface. To remove the address from the interface, use the **no** form of this command.

**ipv6 address** *ipv6-address* **link-local** [**route-tag** *route-tag* *value*]  
**no ipv6 address** *ipv6-address* **link-local** [**route-tag** *route-tag* *value*]

Syntax Description	ipv6-address	The IPv6 address assigned to the interface.  This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.	
	link-local	Specifies a link-local address. The <i>ipv6-address</i> value specified with this command overrides the link-local address that is automatically generated for the interface.	
	route-tag	(Optional) Specifies that the configured address has a route-tag to be associated with it.	
	route-tag value	(Optional) Displays the route-tag value. Range is 1 to 4294967295.	
Command Default	No IPv6 address is defined for the interface.		
Command Modes	Interface configuration		
Command History	Release	Modification	
	Release 7.0.12	This command was introduced.	
Usage Guidelines	If the Cisco IOS XR software detects another host using one of its IPv6 addresses, the software displays an error message on the console.		
	The system automatically generates a link-local address for an interface when IPv6 processing is enabled on the interface, typically when an IPv6 address is configured on the interface. To manually specify a link-local address to be used by an interface, use the <b>ipv6 address link-local</b> command.		
	A double colon may be used as part of the <i>ipv6-address</i> argument when consecutive 16-bit values are denoted as zero. You can configure multiple IPv6 addresses per interfaces, but only one link-local address.		
Task ID	Task ID	Operations	
	ipv6	read, write	
	network	read, write	

---

**Examples**

The following example shows how to assign FE80::260:3EFF:FE11:6770 as the link-local address for HundredGigE interface 0/0/0/1:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/0/0/1  
RP/0/RP0/CPU0:router(config-if)# ipv6 address FE80::260:3EFF:FE11:6770 link-local
```



# ipv6 assembler

To configure the maximum number of packets that are allowed in assembly queues or to configure the number of seconds an assembly queue will hold before timeout, use the **ipv6 assembler** command in the appropriate configuration mode. To disable this feature, use the **no** form of this command.

**ipv6 assembler** { **max-packets** *value* | **timeout** *seconds* }  
**no ipv6 assembler** { **max-packets** *value* | **timeout** *seconds* }

<b>Syntax Description</b>	<b>max-packets</b>	Maximum packets allowed in assembly queues.
	<b>timeout</b>	Number of seconds an assembly queue will hold before timeout.
<b>Command Default</b>	None	
<b>Command Modes</b>	XR Config mode	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	No specific guidelines impact the use of this command.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operation</b>
	ipv6	read, write

## Example

The following example shows how to configure the maximum number of packets that are allowed in assembly queues:

```
Router# config
Router(config)# ipv6 assembler max-packets 35
```

# ipv6 conflict-policy

To enable IP Address Repository Manager (IPARM) conflict resolution, use the **ipv6 conflict-policy** command in XR Config mode. To disable the IPARM conflict resolution, use the **no** form of the command.

**ipv6 conflict-policy** {highest-ip | longest-prefix | static}  
**no ipv6 conflict-policy** {highest-ip | longest-prefix | static}

Syntax Description	<b>highest-ip</b>	Keeps the highest IP address in the conflict set.
	<b>longest-prefix</b>	Keeps the longest prefix match in the conflict set.
	<b>static</b>	Keeps the existing interface running across new address configurations.

**Command Default** Default is the lowest rack/slot if no conflict policy is configured.

**Command Modes** XR Config mode

Command History	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines** No specific guidelines impact the use of this command.

Task ID	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write
	ip-services	read, write

**Examples** The following example shows how to enable the longest prefix policy for conflict resolution:

```
RP/0/RP0/CPU0:router(config)# ipv6 conflict-policy longest-prefix
```

# ipv6 enable

To enable IPv6 processing on an interface that has not been configured with an explicit IPv6 address, use the **ipv6 enable** command in an appropriate configuration mode. To disable IPv6 processing on an interface that has not been configured with an explicit IPv6 address, use the **no** form of this command.

**ipv6 enable**  
**no ipv6 enable**

<b>Syntax Description</b>	None				
<b>Command Default</b>	IPv6 is disabled.				
<b>Command Modes</b>	Interface configuration (not applicable for BNG) Dynamic template configuration (for BNG)				
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				

**Usage Guidelines**

The **ipv6 enable** command automatically configures an IPv6 link-local unicast address on the interface while also enabling the interface for IPv6 processing. The **no ipv6 enable** command does not disable IPv6 processing on an interface that is configured with an explicit IPv6 address.

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the XR EXEC mode.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write
	network	read, write
	config-services	read, write

## Examples

This example (not applicable for BNG) shows how to enable IPv6 processing on HundredGigE interface 0/0/0/1:

```
Router(config)# interface HundredGigE0/0/0/1
Router(config-if)# ipv6 enable
```

# ipv6 hop-limit

To configure the maximum number of hops used in router advertisements and all IPv6 packets that are originated by the router, use the **ipv6 hop-limit** command in XR Config mode mode. To return the hop limit to its default value, use the **no** form of this command.

**ipv6 hop-limit** *hops*  
**no ipv6 hop-limit** *hops*

<b>Syntax Description</b>	<i>hops</i> Maximum number of hops. Range is 1 to 255.	
<b>Command Default</b>	<i>hops</i> : 64 hops	
<b>Command Modes</b>	XR Config mode	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	No specific guidelines impact the use of this command.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write
	network	read, write
<b>Examples</b>	<p>The following example shows how to configure a maximum number of 15 hops for router advertisements and all IPv6 packets that are originated from the router:</p> <pre>RP/0/RP0/CPU0:router(config)# <b>ipv6 hop-limit 15</b></pre>	

# ipv6 icmp error-interval

To configure the interval and bucket size for IPv6 Internet Control Message Protocol (ICMP) error messages on all nodes, use the **ipv6 icmp error-interval** command in XR Config mode. To return the interval to its default setting, use the **no** form of this command.

**ipv6 icmp error-interval** *milliseconds* [*bucketsize*]  
**no ipv6 icmp error-interval**

<b>Syntax Description</b>	<div> <div><i>milliseconds</i></div> <div>Time interval (in milliseconds) between tokens being placed in the bucket. Range is 0 to 2147483647.</div> </div> <div> <div><i>bucketsize</i></div> <div>(Optional) The maximum number of tokens stored in the bucket. The acceptable range is 1 to 200 with a default of 10 tokens.</div> </div>						
<b>Command Default</b>	<p>ICMP rate limiting is enabled by default. To disable ICMP rate limiting, set the interval to zero.</p> <p><i>milliseconds</i> : 100 milliseconds</p> <p><i>bucketsize</i> : 10 tokens</p>						
<b>Command Modes</b>	XR Config mode						
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.		
Release	Modification						
Release 7.0.12	This command was introduced.						
<b>Usage Guidelines</b>	<p>Use the <b>ipv6 icmp error-interval</b> command in XR Config mode to limit the rate at which IPv6 ICMP error messages are sent for each node. A token bucket algorithm is used with one token representing one IPv6 ICMP error message. Tokens are placed in the virtual bucket at a specified interval until the maximum number of tokens allowed in the bucket is reached.</p> <p>The <i>milliseconds</i> argument specifies the time interval between tokens being placed in the bucket. The optional <i>bucketsize</i> argument is used to define the maximum number of tokens stored in the bucket. Tokens are removed from the bucket when IPv6 ICMP error messages are sent, which means that if the <i>bucketsize</i> argument is set to 20, a rapid succession of 20 IPv6 ICMP error messages can be sent. When the bucket is empty of tokens, IPv6 ICMP error messages are not sent until a new token is placed in the bucket.</p> <p>Use the <b>show ipv6 traffic EXEC</b> command to display IPv6 ICMP rate-limited counters.</p>						
<b>Task ID</b>	<table> <tr> <th>Task ID</th><th>Operations</th></tr> <tr> <td>ipv6</td><td>read, write</td></tr> <tr> <td>network</td><td>read, write</td></tr> </table>	Task ID	Operations	ipv6	read, write	network	read, write
Task ID	Operations						
ipv6	read, write						
network	read, write						

---

**Examples**

The following example shows an interval of 50 milliseconds and a bucket size of 20 tokens being configured for IPv6 ICMP error messages:

```
RP/0/RP0/CPU0:router(config)# ipv6 icmp error-interval 50 20
```

# ipv6 mtu

To set the maximum transmission unit (MTU) size of IPv6 packets sent on an interface, use the **ipv6 mtu** command in an appropriate configuration mode. To restore the default MTU size, use the **no** form of this command.

**ipv6 mtu** *bytes*  
**no ipv6 mtu**

<b>Syntax Description</b>	<i>bytes</i> MTU in bytes. Range is 1280 to 65535 for IPv6 packets. The maximum MTU size that can be set on an interface depends on the interface medium.				
<b>Command Default</b>	If no MTU size is configured for IPv6 packets sent on an interface, the interface derives the MTU from the Layer 2 MTU.				
<b>Command Modes</b>	Interface configuration (not applicable for BNG) Dynamic template configuration (for BNG)				
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				
<b>Usage Guidelines</b>	<p>If an IPv6 packet exceeds the MTU set for the interface, only the source router of the packet can fragment it. The maximum MTU size that can be set on an interface depends on the interface medium. If the Layer 2 MTU is smaller than the Layer 3 MTU, the Cisco IOS XR software uses the Layer 2 MTU value for the Layer 3 MTU. Conversely, If the Layer 3 MTU is smaller than the Layer 2 MTU, the software uses Layer 3 MTU value. In other words the Cisco IOS XR software uses the lower of the two values for the MTU.</p> <p>All devices on a physical medium must have the same protocol MTU to operate.</p> <p>For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run <b>dynamic-template</b> command in the XR Config mode.</p>				



**Note** Changing the MTU value (with the **mtu** interface configuration command) can affect the IPv6 MTU value. If the current IPv6 MTU value is the same as the MTU value, and you change the MTU value, the IPv6 MTU value will be modified automatically to match the new MTU. However, the reverse is not true; changing the IPv6 MTU value has no effect on the value for the **mtu** command.

Task ID	Task ID	Operations
	ipv6	read, write
	network	read, write

Task ID	Operations
config-services	read, write

## Examples

This example (not applicable for BNG) shows how to set the maximum IPv6 packet size for HundredGigE interface 0/0/0/1 to 1350 bytes:

```
Router(config)# interface HundredGigE0/0/0/1  
Roputer(config-if)# ipv6 mtu 1350
```



# ipv6 nd dad attempts

To configure the number of consecutive neighbor solicitation messages that are sent on an interface while duplicate address detection is performed on the unicast IPv6 addresses of the interface, use the **ipv6 nd dad attempts** command in an appropriate configuration mode. To return the number of messages to the default value, use the **no** form of this command.

**ipv6 nd dad attempts** *value*  
**no ipv6 nd dad attempts** *value*

<b>Syntax Description</b>	<i>value</i> Number of neighbor solicitation messages. Range is 0 to 600. Configuring a value of 0 disables duplicate address detection processing on the specified interface; a value of 1 configures a single transmission without follow-up transmissions.				
<b>Command Default</b>	Duplicate address detection on unicast IPv6 addresses with the sending of one neighbor solicitation message is enabled. The default is one message.				
<b>Command Modes</b>	Interface configuration (not applicable for BNG) Dynamic template configuration (for BNG)				
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 6.0</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 6.0	This command was introduced.
Release	Modification				
Release 6.0	This command was introduced.				
<b>Usage Guidelines</b>	<p>Duplicate address detection verifies the uniqueness of new unicast IPv6 addresses before the addresses are assigned to interfaces (the new addresses remain in a tentative state while duplicate address detection is performed). Duplicate address detection uses neighbor solicitation messages to verify the uniqueness of unicast IPv6 addresses.</p> <p>The DupAddrDetectTransmits node configuration variable (as specified in RFC 2462, <i>IPv6 Stateless Address Autoconfiguration</i>) is used to automatically determine the number of consecutive neighbor solicitation messages that are sent on an interface while duplicate address detection is performed on a tentative unicast IPv6 address.</p> <p>The interval between the sending of duplicate address detection neighbor solicitation messages (the duplicate address detection timeout interval) is specified by the neighbor discovery-related variable RetransTimer (as specified in RFC 2461, <i>Neighbor Discovery for IP Version 6 [IPv6]</i>), which is used to determine the time between retransmissions of neighbor solicitation messages to a neighbor when the address is being resolved or when the reachability of a neighbor is being probed. This is the same management variable used to specify the interval for neighbor solicitation messages during address resolution and neighbor unreachability detection. Use the <b>ipv6 nd ns-interval</b> command to configure the interval between neighbor solicitation messages that are sent during duplicate address detection.</p> <p>Duplicate address detection is suspended on interfaces that are administratively down. While an interface is administratively down, the unicast IPv6 addresses assigned to the interface are set to a pending state. Duplicate address detection is automatically restarted on an interface when the interface returns to being administratively up.</p> <p>For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run <b>dynamic-template</b> command in the .</p>				



**Note** An interface returning to administratively up restarts duplicate address detection for all of the unicast IPv6 addresses on the interface. While duplicate address detection is performed on the link-local address of an interface, the state for the other IPv6 addresses is still set to tentative. When duplicate address detection is completed on the link-local address, duplicate address detection is performed on the remaining IPv6 addresses.

When duplicate address detection identifies a duplicate address, the state of the address is set to duplicate and the address is not used. If the duplicate address is the link-local address of the interface, the processing of IPv6 packets is disabled on the interface and an error message similar to the following is issued:

```
ipv6_nd[145]: %IPV6_ND-3-ADDRESS_DUPLICATE : Duplicate address 111::1 has been detected
```

If the duplicate address is a global address of the interface, the address is not used and an error message similar to the following is issued:

```
%IPV6-4-DUPLICATE: Duplicate address 3000::4 on &HundredGigE;
```

All configuration commands associated with the duplicate address remain as configured while the state of the address is set to duplicate.

If the link-local address for an interface changes, duplicate address detection is performed on the new link-local address and all of the other IPv6 address associated with the interface are regenerated (duplicate address detection is performed only on the new link-local address).

Task ID	Task ID	Operations
	ipv6	read, write
	config-services	read, write

## Examples

This example (not applicable for BNG) shows how to set the number of consecutive neighbor solicitation messages for interface 0/2/0/1 to 1 and then display the state (tentative or duplicate) of the unicast IPv6 address configured for an interface:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/2/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 nd dad attempts 1
RP/0/RP0/CPU0:router(config-if)# Uncommitted changes found, commit them before
exiting(yes/no/cancel)? [cancel]:y
```

```
RP/0/RP0/CPU0:router# show ipv6 interface
HundredGigE/2/0/0 is Up, line protocol is Up
  IPv6 is disabled, link-local address unassigned
  No global unicast address is configured
HundredGigE/2/0/1 is Up, line protocol is Up
  IPv6 is enabled, link-local address is fe80::203:fdff:fe1b:4501
  Global unicast address(es):
    1:4::1, subnet is 1:4::/64 [DUPLICATE]
  MTU is 1514 (1500 is available to IPv6)
```

```
ICMP redirects are disabled
ND DAD is enabled, number of DAD attempts 1
ND reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
HundredGigE/2/0/2 is Shutdown, line protocol is Down
IPv6 is enabled, link-local address is fe80::200:11ff:fe11:1111 [TENTATIVE]
Global unicast address(es):
  111::2, subnet is 111::/64 [TENTATIVE]
MTU is 1514 (1500 is available to IPv6)
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts 1
ND reachable time is 0 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
```

For BNG, this example shows how to display the state (tentative or duplicate) of the unicast IPv6 address on the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd dad attempts 1
```

# ipv6 nd managed-config-flag

To set the managed address configuration flag in IPv6 router advertisements, use the **ipv6 nd managed-config-flag** command in an appropriate configuration mode. To clear the flag from IPv6 router advertisements, use the **no** form of this command.

**ipv6 nd managed-config-flag**  
**no ipv6 nd managed-config-flag**

<b>Syntax Description</b>	This command has no keywords or arguments.				
<b>Command Default</b>	The managed address configuration flag is not set in IPv6 router advertisements.				
<b>Command Modes</b>	Interface configuration (not applicable for BNG) Dynamic template configuration (for BNG)				
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				
<b>Usage Guidelines</b>	<p>Setting the managed address configuration flag in IPv6 router advertisements indicates to attached hosts whether they should use stateful autoconfiguration to obtain addresses. If the flag is set, the attached hosts should use stateful autoconfiguration to obtain addresses. If the flag is not set, the attached hosts should not use stateful autoconfiguration to obtain addresses.</p> <p>Hosts may use stateful and stateless address autoconfiguration simultaneously.</p> <p>For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run <b>dynamic-template</b> command in the XR EXEC mode.</p>				

Task ID	Task ID	Operations
	ipv6	read, write
	network	read, write
	config-services	read, write

## Examples

This example (not applicable for BNG) shows how to configure the managed address configuration flag in IPv6 router advertisements on HundredGigE interface 0/1/0/1:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 nd managed-config-flag
```

For BNG, this example shows how to configure the managed address configuration flag in IPv6 router advertisements on dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1  
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd managed-config-flag
```

# ipv6 nd ns-interval

To configure the interval between IPv6 neighbor solicitation retransmissions on an interface, use the **ipv6 nd ns-interval** command in an appropriate configuration mode. To restore the default interval, use the **no** form of this command.

**ipv6 nd ns-interval** *milliseconds*  
**no ipv6 nd ns-interval**

<b>Syntax Description</b>	<i>milliseconds</i> Interval (in milliseconds) between IPv6 neighbor solicit transmissions. Range is 1000 to 3600000.				
<b>Command Default</b>	0 milliseconds (unspecified) is advertised in router advertisements, and the value 1000 is used for the neighbor discovery activity of the router itself.				
<b>Command Modes</b>	Interface configuration (not applicable for BNG) Dynamic template configuration (for BNG)				
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				
<b>Usage Guidelines</b>	<p>This value is included in all IPv6 router advertisements sent out from this interface. Very short intervals are not recommended in normal IPv6 operation. When a nondefault value is configured, the configured time is both advertised and used by the router itself.</p> <p>For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run <b>dynamic-template</b> command in the XR Config mode.</p>				

Task ID	Task ID	Operations
	ipv6	read, write
	network	read, write
	config-services	read, write

## Examples

This example (not applicable for BNG) configures an IPv6 neighbor solicit transmission interval of 9000 milliseconds for HundredGigE interface 0/1/0/1:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 nd ns-interval 9000
```

For BNG, this example configures an IPv6 neighbor solicit transmission interval of 9000 milliseconds in the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1  
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd ns-interval 9000
```

# ipv6 nd other-config-flag

To set the other stateful configuration flag in IPv6 router advertisements, use the **ipv6 nd other-config-flag** command in an appropriate configuration mode. To clear the flag from IPv6 router advertisements, use the **no** form of this command.

**ipv6 nd other-config-flag**  
**no ipv6 nd other-config-flag**

<b>Syntax Description</b>	This command has no keywords or arguments.				
<b>Command Default</b>	The other stateful configuration flag is not set in IPv6 router advertisements.				
<b>Command Modes</b>	Interface configuration (not applicable for BNG) Dynamic template configuration (for BNG)				
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				

**Usage Guidelines** The setting of the other stateful configuration flag in IPv6 router advertisements indicates to attached hosts how they can obtain autoconfiguration information other than addresses. If the flag is set, the attached hosts should use stateful autoconfiguration to obtain the other (nonaddress) information.



**Note** If the managed address configuration flag is set using the **ipv6 nd managed-config-flag** command, then an attached host can use stateful autoconfiguration to obtain the other (nonaddress) information regardless of the setting of the other stateful configuration flag.

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the XR Config mode.

Task ID	Task ID	Operations
	ipv6	read, write
	network	read, write
	config-services	read, write

**Examples** This example (not applicable for BNG) configures the “other stateful configuration” flag in IPv6 router advertisements on HundredGigE interface 0/1/0/1:



```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/1  
RP/0/RP0/CPU0:router(config-if)# ipv6 nd other-config-flag
```

For BNG, this example configures the “other stateful configuration” flag for IPv6 router advertisements in the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1  
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd other-config-flag
```

# ipv6 nd prefix

To configure how IPv6 prefixes are advertised in IPv6 router advertisements, use the **ipv6 nd prefix** command in interface configuration mode. To advertise a prefix with default parameter values, use the **no** form of this command. To prevent a prefix (or prefixes) from being advertised, use the **no-adv** keyword.

**ipv6 nd prefix** {*ipv6prefix/prefix-length* | **default** [{*valid-lifetime* | **at** | **infinite** | **no-adv** | **no-autoconfig** | **off-link**}]}

**no ipv6 nd prefix** {*ipv6prefix/prefix-length* | **default** [{*valid-lifetime* | **at** | **infinite** | **no-adv** | **no-autoconfig** | **off-link**}]}

Syntax Description	ipv6-prefix	The IPv6 network number to include in router advertisements.  This keyword must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
	/prefix-length	The length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address compose the prefix (the network portion of the address). A slash (/) must precede the decimal value.
	default	(Optional) Specifies all prefixes.
	valid-lifetime	(Optional) The amount of time (in seconds) that the specified IPv6 prefix is advertised as being valid. The range of values is 0 to 4294967295 seconds.
	at	(Optional) The date and time at which the lifetime and preference expire. The prefix is valid until this specified date and time are reached. Dates are expressed in the form <i>date-valid-expire month-valid-expire hh:mm-valid-expire date-prefer-expire month-prefer-expire hh:mm-prefer-expire</i> .
	infinite	(Optional) The valid lifetime does not expire.
	no-adv	(Optional) The prefix is not advertised.
	no-autoconfig	(Optional) Indicates to hosts on the local link that the specified prefix cannot be used for IPv6 autoconfiguration.
	off-link	(Optional) Indicates that the specified prefix is assigned to the link. Nodes sending traffic to such addresses that contain the specified prefix consider the destination to be locally reachable on the link. This prefix should not be used for <i>onlink</i> determination.
Command Default	All prefixes configured on interfaces that originate IPv6 router advertisements are advertised with a valid lifetime of 2592000 seconds (30 days) and a preferred lifetime of 604800 seconds (7 days), and with both the “onlink” and “autoconfig” flags set.	
Command Modes	Interface configuration	
Command History	Release	Modification
	Release 7.0.12	This command was introduced.

### Usage Guidelines

This command allows control over the individual parameters per prefix, including whether or not the prefix should be advertised.

To control how prefixes are advertised, use the **ipv6 nd prefix** command. By default, prefixes configured as addresses on an interface using the **ipv6 address** command are advertised with default values. If you configure prefixes for advertisement using the **ipv6 nd prefix** command, only the specified prefixes are advertised with the configured values, all other prefixes are advertised with default values.

The default keyword can be used to set default parameters for all prefixes.

A date can be set to specify the expiration of a prefix. The valid and preferred lifetimes are counted down in real time. When the expiration date is reached, the prefix is no longer advertised.

When onlink is “on” (by default), the specified prefix is assigned to the link. Nodes sending traffic to such addresses that contain the specified prefix consider the destination to be locally reachable on the link.

When autoconfig is “on” (by default), it indicates to hosts on the local link that the specified prefix can be used for IPv6 autoconfiguration.

### Task ID

Task ID	Operations
ipv6	read, write
network	read, write

### Examples

The following example includes the IPv6 prefix 2001:0DB8::/35 in router advertisements sent out HundredGigE interface 0/1/0/0 with a valid lifetime of 1000 seconds and a preferred lifetime of 900 seconds:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/0
RP/0/RP0/CPU0:router(config-if)# ipv6 nd prefix 2001:0DB8::/35 1000 900
```

# ipv6 nd ra dns server

To configure the IPv6 router advertisement of DNS server addresses on an interface, use the **ipv6 nd ra dns server** command in interface configuration mode. To remove the IPv6 router advertisement of DNS server addresses, use the **no** form of this command.

**ipv6 nd ra dns server** *ipv6-address* {*seconds* | **infinite-lifetime** | **zero-lifetime** }  
**no ipv6 nd ra dns server** *ipv6-address*  
**no ipv6 nd ra dns server**

Syntax Description	<b>server</b> <i>ipv6-address</i>	Specify the DNS server address to be advertised in an IPv6 router advertisement (RA).
	<i>seconds</i>   <b>infinite-lifetime</b>   <b>zero-lifetime</b>	The amount of time that the DNS server is advertised in an IPv6 RA. The range for seconds is from 200 to 4294967295. The lifetime can also be specified as infinite or zero.

**Command Default** The DNS server is not advertised in an IPv6 RA.

**Command Modes** Interface configuration

Command History	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines**

This configuration is not allowed for management interfaces.

You can use the **ipv6 nd ra dns server** command to configure up to five DNS server addresses in an RA.

If you configure a seconds value of zero, the DNS server will no longer be used.

Use the **no ipv6 nd ra dns server** *ipv6-address* command to delete a single DNS server under an interface.

Use the **no ipv6 nd ra dns server** command to delete all DNS servers under an interface.

Task ID	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write
	network	read, write
	config-services	read, write

**Examples**

The following example configures a DNS server with an IPv6 address of 2001:DB8:1::1 to be advertised in an RA with a lifetime of 600 seconds:

```
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/2/0/0
RP/0/RP0/CPU0:router(config-if)# ipv6 nd ra dns server 2001:DB8:1::1 600
```

The following example configures a DNS server with an IPv6 address of 4::4 to be advertised in an RA with an infinite lifetime:

```
RP/0/RP0/CPU0:router(config)# interface GigabitEthernet 0/2/0/0
RP/0/RP0/CPU0:router(config-if)# ipv6 nd ra dns server 4::4 infinite-lifetime
```

**Related Commands**

Command	Description
<a href="#">ipv6 nd ra-lifetime, on page 58</a>	Configures the lifetime of an IPv6 router advertisement.
<a href="#">show ipv6 interface , on page 99</a>	Displays the usability status of interfaces configured for IPv6.

# ipv6 nd ra dns search list

To configure the IPv6 router advertisement of DNS search list on an interface, use the **ipv6 nd ra dns search list** command in interface configuration mode. To remove the IPv6 router advertisement of DNS search list, use the **no** form of this command.

**ipv6 nd ra dns search list** *name* {*seconds* | **infinite-lifetime** | **zero-lifetime** }  
**no ipv6 nd ra dns search list** *name*  
**no ipv6 nd ra dns search list**

Syntax Description	<i>name</i>	Specify the DNS search list to be advertised in an IPv6 router advertisement (RA).
	<i>seconds</i>   <b>infinite-lifetime</b>   <b>zero-lifetime</b>	The amount of time that the DNS search list is advertised in an IPv6 RA. The range for seconds is from 200 to 4294967295. The lifetime can also be specified as infinite or zero.

**Command Default** The DNS search list is not advertised in an IPv6 RA.

**Command Modes** Interface configuration

Command History	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines** This configuration is not allowed for management interfaces.

You can use the **ipv6 nd ra dns search list** command to configure up to 50 DNS search lists in an RA.

If you configure a seconds value of zero, the DNS server will no longer be used.

Use the **no ipv6 nd ra dns search list** *name* command to delete a single DNS search list under an interface.

Use the **no ipv6 nd ra dns search list** command to delete all DNS search lists under an interface.

Task ID	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write
	network	read, write
	config-services	read, write

**Examples** The following example configures a DNS search list with a name of aaa.cc.com to be advertised in an RA with an infinite lifetime:

```
Router(config)# interface GigabitEthernet 0/2/0/0
Router(config-if)# ipv6 nd ra dns search list aaa.cc.com infinite-lifetime
```

**Related Commands**

Command	Description
<a href="#">ipv6 nd ra-lifetime, on page 58</a>	Configures the lifetime of an IPv6 router advertisement.
<a href="#">ipv6 nd ra-lifetime, on page 58</a>	Displays the usability status of interfaces configured for IPv6.

# ipv6 nd ra-interval

To configure the interval between IPv6 router advertisement transmissions on an interface, use the **ipv6 nd ra-interval** command in an appropriate configuration mode. To restore the default interval, use the **no** form of this command.

```
ipv6 nd ra-interval seconds
no ipv6 nd ra-interval seconds
```

<b>Syntax Description</b>	<i>seconds</i> The interval (in seconds) between IPv6 router advertisement transmissions.								
<b>Command Default</b>	<i>seconds</i> : 200 seconds								
<b>Command Modes</b>	Interface configuration (not applicable for BNG) Dynamic template configuration (for BNG)								
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.				
Release	Modification								
Release 7.0.12	This command was introduced.								
<b>Usage Guidelines</b>	<p>The interval between transmissions should be less than or equal to the IPv6 router advertisement lifetime if the router is configured as a default router by using the <b>ipv6 nd ra-lifetime</b> command. To prevent synchronization with other IPv6 nodes, randomly adjust the actual value used to within 20 percent of the specified value.</p> <p>For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run <b>dynamic-template</b> command in the XR Config mode.</p>								
<b>Task ID</b>	<table> <tr> <th>Task ID</th><th>Operations</th></tr> <tr> <td>ipv6</td><td>read, write</td></tr> <tr> <td>network</td><td>read, write</td></tr> <tr> <td>config-services</td><td>read, write</td></tr> </table>	Task ID	Operations	ipv6	read, write	network	read, write	config-services	read, write
Task ID	Operations								
ipv6	read, write								
network	read, write								
config-services	read, write								
<b>Examples</b>	<p>This example (not applicable for BNG) configures an IPv6 router advertisement interval of 201 seconds on HundredGigE interface 0/1/0/1:</p> <pre>Router(config)# interface HundredGigE0/1/0/1 Router(config-if)# ipv6 nd ra-interval 201</pre> <p>For BNG, this example configures an IPv6 router advertisement interval of 201 seconds in the dynamic template configuration mode:</p>								



```
Router(config)# dynamic-template type ppp p1  
Router(config-dynamic-template-type)# ipv6 nd ra-interval 201
```

# ipv6 nd ra-lifetime

To configure the router lifetime value in IPv6 router advertisements on an interface, use the **ipv6 nd ra-lifetime** command in an appropriate configuration mode. To restore the default lifetime, use the **no** form of this command.

**ipv6 nd ra-lifetime** *seconds*  
**no ipv6 nd ra-lifetime**

<b>Syntax Description</b>	<i>seconds</i> The validity (in seconds) of this router as a default router on this interface.
---------------------------	--

<b>Command Default</b>	<i>seconds</i> : 1800 seconds
------------------------	-------------------------------

<b>Command Modes</b>	Interface configuration (not applicable for BNG) Dynamic template configuration (for BNG)
----------------------	--

<b>Usage Guidelines</b>	The router lifetime value is included in all IPv6 router advertisements sent out the interface. The value indicates the usefulness of the router as a default router on this interface. Setting the value to 0 indicates that the router should not be considered a default router on this interface. The router lifetime value can be set to a nonzero value to indicate that it should be considered a default router on this interface. The nonzero value for the router lifetime value should not be less than the router advertisement interval.
-------------------------	---

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the XR Config mode.

Task ID	Task ID	Operations
	ipv6	read, write
	network	read, write
	config-services	read, write

## Examples

This example (not applicable for BNG) configures an IPv6 router advertisement lifetime of 1801 seconds on HundredGigE interface 0/1/0/1:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 nd ra-lifetime 1801
```

For BNG, this example configures an IPv6 router advertisement lifetime of 1801 seconds in the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd ra-lifetime 1801
```

# ipv6 nd ra specific route

To configure specific route for a router on a specific interface, use the **ipv6 nd ra specific route** command in interface configuration mode. To delete a single or all specific routes, use the **no** form of this command.

```

ipv6 nd ra specific route prefix Lifetime {seconds | infinite-lifetime | zero-lifetime } [preference { high
| medium | low }]
no ipv6 nd ra specific route prefix
no ipv6 nd ra specific route

```

<b>Syntax Description</b>	<b>route</b> <i>prefix</i>	Variable-length field containing an IP address or a prefix of an IP address to identify a route.
	<b>Lifetime</b> { <i>seconds</i>   <b>infinite-lifetime</b>   <b>zero-lifetime</b> }	The length of time the route prefix is valid for route determination specified as seconds, infinite, or zero.
	[ <b>preference</b> { <b>high</b>   <b>medium</b>   <b>low</b> } ]	(Optional) Preference for the router specified on an interface specified as high, medium, or low.

**Command Default** Router advertisements (RAs) are sent with the medium preference.

**Command Modes** Interface configuration

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines**

This configuration is not allowed for management interfaces.

If the Lifetime is set to zero, then the host will no longer use the router for route aspect of the route information option.

If no preference is specified, then the default value for preference (medium) is used.

Use the **no ipv6 nd ra specific route** *prefix* command to delete a single specific route under an interface. Use the **no ipv6 nd ra specific route** command to delete all specific routes under an interface.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write
	network	read, write
	config-services	read, write

**Examples**

The following example configures a specific route for the router on gigabit Ethernet interface 0/2/0/0:

```
Router(config)# interface GigabitEthernet 0/2/0/0
Router(config-if)# ipv6 nd ra specific route 3::3/116 Lifetime 1112 preference low
```

**Related Commands**

Command	Description
<a href="#">ipv6 nd ra-lifetime, on page 58</a>	Configures the lifetime of an IPv6 router advertisement.
<a href="#">ipv6 nd ra-lifetime, on page 58</a>	Displays the usability status of interfaces configured for IPv6.

# ipv6 nd reachable-time

To configure the amount of time that a remote IPv6 node is considered reachable after some reachability confirmation event has occurred, use the **ipv6 nd reachable-time** command in an appropriate configuration mode. To restore the default time, use the **no** form of this command.

**ipv6 nd reachable-time** *milliseconds*  
**no ipv6 nd reachable-time**

<b>Syntax Description</b>	<i>milliseconds</i> The amount of time (in milliseconds) that a remote IPv6 node is considered reachable. The range is from 0 to 3600000.								
<b>Command Default</b>	0 milliseconds (unspecified) is advertised in router advertisements and 30000 (30 seconds) is used for the neighbor discovery activity of the router itself.								
<b>Command Modes</b>	Interface configuration (not applicable for BNG) Dynamic template configuration (for BNG)								
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.				
Release	Modification								
Release 7.0.12	This command was introduced.								
<b>Usage Guidelines</b>	<p>The configured time enables the router to detect unavailable neighbors. Shorter configured times enable the router to detect unavailable neighbors more quickly; however, shorter times consume more IPv6 network bandwidth and processing resources in all IPv6 network devices. Very short configured times are not recommended in normal IPv6 operation.</p> <p>The configured time is included in all router advertisements sent out of an interface so that nodes on the same link use the same time value. A value of 0 indicates that the configured time is unspecified by this router.</p> <p>For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run <b>dynamic-template</b> command in the XR Config mode.</p>								
<b>Task ID</b>	<table> <tr> <th>Task ID</th><th>Operations</th></tr> <tr> <td>ipv6</td><td>read, write</td></tr> <tr> <td>network</td><td>read, write</td></tr> <tr> <td>config-services</td><td>read, write</td></tr> </table>	Task ID	Operations	ipv6	read, write	network	read, write	config-services	read, write
Task ID	Operations								
ipv6	read, write								
network	read, write								
config-services	read, write								
<b>Examples</b>	<p>This example (not applicable for BNG) shows how to configure an IPv6 reachable time of 1,700,000 milliseconds for HundredGigE interface 0/1/0/1:</p> <pre>RP/0/RP0/CPU0:router(config)# <b>interface HundredGigE0/1/0/1</b></pre>								

```
RP/0/RP0/CPU0:router(config-if)# ipv6 nd reachable-time 1700000
```

For BNG, this example shows how to configure an IPv6 reachable time of 1,700,000 milliseconds in the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1  
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd reachable-time 1700000
```

# ipv6 nd redirects

To send Internet Control Message Protocol (ICMP) redirect messages, use the **ipv6 nd redirects** command in interface configuration mode. To restore the system default, use the **no** form of this command.

**ipv6 nd redirects**  
**no ipv6 nd redirects**

<b>Syntax Description</b>	This command has no keywords or arguments.
---------------------------	--

<b>Command Default</b>	The default value is disabled.
------------------------	--------------------------------

<b>Command Modes</b>	Interface configuration
----------------------	-------------------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

<b>Usage Guidelines</b>	This command has no keywords or arguments.
-------------------------	--

<b>Task ID</b>	<b>Task ID    Operations</b>	
	ipv6	read, write
	network	read, write

<b>Examples</b>	The following example shows how to redirect IPv6 nd-directed broadcasts on HundredGigE interface 0/2/0/2:
-----------------	---

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/2/0/2  
RP/0/RP0/CPU0:router(config-if)# ipv6 nd redirects
```

# ipv6 nd router-preference

To configure a default router preference (DRP) for the router on a specific interface, use the **ipv6 nd router-preference** command in interface configuration mode. To return to the default DRP, use the **no** form of this command.

**ipv6 nd router-preference {high | medium | low }**  
**no ipv6 nd router-preference**

Syntax Description	<b>high</b>	Preference for the router specified on an interface is high.
	<b>medium</b>	Preference for the router specified on an interface is medium.
	<b>low</b>	Preference for the router specified on an interface is low.

**Command Default** Router advertisements (RAs) are sent with the medium preference.

**Command Modes** Interface configuration

Command History	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines**

This configuration is not allowed for management interfaces.

RA messages are sent with the DRP configured by the `ipv6 nd router-preference` command. If no DRP is configured, RAs are sent with a medium preference.

A DRP is useful when, for example, two routers on a link may provide equivalent, but not equal-cost, routing, and policy may dictate that hosts should prefer one of the routers.

Task ID	Task ID	Operations
	ipv6	read, write
	network	read, write
	config-services	read, write

**Examples**

The following example configures a DRP of high for the router on gigabit Ethernet interface 0/2/0/0:

```
Router(config)# interface GigabitEthernet 0/2/0/0
Router(config-if)# ipv6 nd router-preference high
```



**Related Commands**

Command	Description
<a href="#">ipv6 nd ra-lifetime, on page 58</a>	Configures the lifetime of an IPv6 router advertisement.
<a href="#">ipv6 nd ra-lifetime, on page 58</a>	Displays the usability status of interfaces configured for IPv6.

# ipv6 nd scavenge-timeout

To set the lifetime for neighbor entries in the stale state, use the **ipv6 nd scavenge-timeout** command in XR Config mode mode. To disable this feature, use the **no** form of this command.

**ipv6 nd scavenge-timeout** *seconds*  
**no ipv6 nd scavenge-timeout** *seconds*

<b>Syntax Description</b>	seconds RA lifetime in seconds. The range is from 0 to 43200.	
<b>Command Default</b>	None	
<b>Command Modes</b>	XR Config mode	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	When the scavenge-timer for a neighbor entry expires, the entry is cleared.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write
	network	read, write
<b>Examples</b>	The following example shows how to set the lifetime for the neighbor entry:	
	RP/0/RP0/CPU0:router(config)# <b>ipv6 nd scavenge-timeout 3000</b>	

# ipv6 nd suppress-ra

To suppress IPv6 router advertisement transmissions on a LAN interface, use the **ipv6 nd suppress-ra** command in an appropriate configuration mode. To reenable the sending of IPv6 router advertisement transmissions on a LAN interface, use the **no** form of this command.

**ipv6 nd suppress-ra**  
**no ipv6 nd suppress-ra**

<b>Syntax Description</b>	This command has no keywords or arguments.	
<b>Command Default</b>	IPv6 router advertisements are automatically sent on other types of interfaces if IPv6 unicast routing is enabled on the interfaces. IPv6 router advertisements are not sent on other types of interfaces.	
<b>Command Modes</b>	Interface configuration (not applicable for BNG) Dynamic template configuration (for BNG)	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	<p>Use the <b>no ipv6 nd suppress-ra</b> command to enable the sending of IPv6 router advertisement transmissions on non-LAN interface types (for example, serial or tunnel interfaces).</p> <p>For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run <b>dynamic-template</b> command in the XR Config mode.</p>	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write
	network	read, write
	config-services	read, write

## Examples

This example (not applicable for BNG) shows how to suppress IPv6 router advertisements on HundredGigE interface 0/1/0/1:

```
RP/0/RP0/CPU0:router(config)# interface HundredGigE0/1/0/1
RP/0/RP0/CPU0:router(config-if)# ipv6 nd suppress-ra
```

For BNG, this example shows how to suppress IPv6 router advertisements in the dynamic template configuration mode:

```
RP/0/RP0/CPU0:router(config)# dynamic-template type ppp p1
RP/0/RP0/CPU0:router(config-dynamic-template-type)# ipv6 nd suppress-ra
```

# ipv6 neighbor

To configure a static entry in the IPv6 neighbor discovery cache, use the **ipv6 neighbor** command in XR Config mode. To remove a static IPv6 entry from the IPv6 neighbors discovery cache, use the **no** form of this command.

**ipv6 neighbor** *ipv6-address interface-type interface-instance hardware-address*  
**no ipv6 neighbor** *ipv6-address interface-type interface-instance hardware-address*

Syntax Description	<i>ipv6-address</i>	The IPv6 address that corresponds to the local data-link address.  This argument must be in the form documented in RFC 2373 where the address is specified in hexadecimal using 16-bit values between colons.
	<i>interface-type</i>	Interface type. For more information, use the question mark (?) online help function.
	<i>interface-instance</i>	Either a physical interface instance or a virtual interface instance as follows: <ul style="list-style-type: none"> <li>Physical interface instance. Naming notation is <i>rack/slot/module/port</i> and a slash between values is required as part of the notation.               <ul style="list-style-type: none"> <li><i>rack</i>: Chassis number of the rack.</li> <li><i>slot</i>: Physical slot number of the modular services card or line card.</li> <li><i>module</i>: Module number. A physical layer interface module (PLIM) is always 0.</li> <li><i>port</i>: Physical port number of the interface.</li> </ul> </li> </ul> <p><b>Note</b> In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RSP0 ) and the module is CPU0. Example: interface MgmtEth0/RSP0 /CPU0/0.</p> <ul style="list-style-type: none"> <li>Virtual interface instance. Number range varies depending on interface type.</li> </ul> <p>For more information about the syntax for the router, use the question mark (?) online help function.</p>
	<i>hardware-address</i>	The local data-link address (a 48-bit address).

**Command Default** Static entries are not configured in the IPv6 neighbor discovery cache.

**Command Modes** XR Config mode

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines**

The **ipv6 neighbor** command is similar to the **arp** (global) command.

If an entry for the specified IPv6 address already exists in the neighbor discovery cache—learned through the IPv6 neighbor discovery process—the entry is automatically converted to a static entry.

Use the **show ipv6 neighbors** command to display static entries in the IPv6 neighbors discovery cache. A static entry in the IPv6 neighbor discovery cache has one state: reach (reachable)—The interface for this entry is up. If the interface for the entry is down, the **show ipv6 neighbors** command does not show the entry.



**Note** Reachability detection is not applied to static entries in the IPv6 neighbor discovery cache; therefore, the descriptions for the reach (reachable) state are different for dynamic and static cache entries. See the **show ipv6 neighbors** command for a description of the reach (reachable) state for dynamic cache entries.

The **clear ipv6 neighbors** command deletes all entries in the IPv6 neighbors discovery cache, except static entries. The **no ipv6 neighbor** command deletes a specified static entry from the neighbor discovery cache; the command does not remove dynamic entries—learned from the IPv6 neighbor discovery process—from the cache. Disabling IPv6 on an interface by using the **no ipv6 enable** or the **no ipv6 unnumbered** command deletes all IPv6 neighbor discovery cache entries configured for that interface, except static entries (the state of the entry changes to reach [reachable]).

Static entries in the IPv6 neighbor discovery cache are not modified by the neighbor discovery process.



**Note** Static entries for IPv6 neighbors can be configured only on IPv6-enabled LAN and ATM LAN Emulation interfaces.

**Task ID****Task ID Operations**

ipv6	read, write
------	----------------

network	read, write
---------	----------------

**Examples**

The following example shows how to configure a static entry in the IPv6 neighbor discovery cache for a neighbor with the IPv6 address 2001:0DB8::45A and link-layer address 0002.7D1A.9472 on ethernet interface 0/RSP0/CPU0:

```
RP/0/RP0/CPU0:router(config)# ipv6 neighbor 2001:0DB8::45A 0002.7D1A.9472
```

# ipv6 path-mtu enable

To enable the command to configure path maximum transmission unit (MTU) discovery of IPv6 packets, use the **ipv6 path-mtu enable** command in the XR Config mode.

**ipv6 path-mtu enable**

<b>Command Default</b>	None.
------------------------	-------

<b>Command Modes</b>	XR Config mode
----------------------	----------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

<b>Usage Guidelines</b>	Path MTU discovery for IPv6 packets is supported only for applications using TCP and Ping protocol.
-------------------------	---

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write
	network	read, write
	config-services	read, write

<b>Examples</b>	This example shows how to enable path MTU discovery command of IPv6 packets:
-----------------	--

```
RP/0/RP0/CPU0:router(config)# ipv6 path-mtu enable
```

# ipv6 path-mtu timeout

To set the maximum transmission unit (MTU) timeout value of IPv6 packets, use the **ipv6 path-mtu timeout** command in the XR Config mode.

**ipv6 path-mtu timeout** *minutes*

<b>Syntax Description</b>	<i>minutes</i> MTU timeout in minutes. Range is 1 to 15 minutes. Default timeout value is 10 minutes.	
<b>Command Default</b>	None.	
<b>Command Modes</b>	XR Config mode	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	Path MTU discovery for IPv6 packets is supported only for applications using TCP and Ping protocol.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write
	network	read, write
	config-services	read, write
<b>Examples</b>	This example shows how to set path MTU timeout of IPv6 packets:	

```
RP/0/RP0/CPU0:router(config)# ipv6 path-mtu timeout 15
```

# ipv6 source-route

To enable processing of the IPv6 type source (type 0) routing header, use the **ipv6 source-route** command in XR EXEC mode. To disable the processing of this IPv6 extension header, use the **no** form of this command.

**ipv6 source-route**  
**no ipv6 source-route**

<b>Syntax Description</b>	This command has no keywords or arguments.	
<b>Command Default</b>	The <b>no</b> version of the <b>ipv6 source-route</b> command is the default.	
<b>Command Modes</b>	XR EXEC mode	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	The <b>no ipv6 source-route</b> command (which is the default) prevents hosts from performing source routing using your routers. When the <b>no ipv6 source-route</b> command is configured and the router receives a packet with a type 0 source routing header, the router drops the packet and sends an IPv6 ICMP error message back to the source and logs an appropriate debug message.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operation</b>
	network	read, write
	ipv6	read, write

## Example

The following example shows how to allow the processing of any IPv6 datagrams containing a source-route header option:

```
RP/0/RP0/CPU0:router# config
RP/0/RP0/CPU0:router(config)# ipv6 source-route
RP/0/RP0/CPU0:router(config)#
```



# ipv6 tcp-mss-adjust

To enable the transit traffic of TCP flows to be a Maximum Segment Size (MSS) below the GRE tunnel interface or VLAN sub-interface MTU so that traffic fragmentation is prevented when a session is established for IPv6 packets, use the **ipv6 tcp-mss-adjust** command in the interface configuration submode. To disable the transit traffic of TCP flows to be a Maximum Segment Size (MSS) below the GRE tunnel interface or VLAN sub-interface MTU, use the **no** form of this command.

**ipv6 tcp-mss-adjust enable**  
**no ipv6 tcp-mss-adjust enable**

Syntax Description	<b>enable</b> Enables Maximum Segment Size (MSS) adjustment for tcp flows on the interface..						
Command Default	None						
Command Modes	Interface Configuration						
Command History	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.		
Release	Modification						
Release 7.0.12	This command was introduced.						
Usage Guidelines	This command has no keywords or arguments.						
Task ID	<table> <tr> <th>Task ID</th><th>Operation</th></tr> <tr> <td>mpls-te</td><td>read, write</td></tr> <tr> <td>ipv6</td><td>read, write</td></tr> </table>	Task ID	Operation	mpls-te	read, write	ipv6	read, write
Task ID	Operation						
mpls-te	read, write						
ipv6	read, write						

## Example

This example shows how to enable the transit traffic of TCP flows for IPv6 packets using the **ipv6 tcp-mss-adjust** command:

```
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# interface HundredEthernet 0/0/0/4.20
RP/0/RP0/CPU0:router(config-if)# ipv6 tcp-mss-adjust enable
```

# ipv6 unreachable disable

To disable the generation of IPv6 Internet Control Message Protocol (ICMP) unreachable messages, use the **ipv6 unreachable disable** command in an appropriate configuration mode. To re-enable the generation of ICMP unreachable messages, use the **no** form of this command.

**ipv6 unreachable disable**  
**no ipv6 unreachable disable**

<b>Syntax Description</b>	This command has no keywords or arguments.				
<b>Command Default</b>	IPv6 ICMP unreachable messages are generated.				
<b>Command Modes</b>	Interface configuration (not applicable for BNG) Dynamic template configuration (for BNG)				
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				

**Usage Guidelines**

If the software receives a nonbroadcast packet destined for itself that uses a protocol it does not recognize, it sends an ICMP protocol unreachable message to the source.

If the software receives a datagram that it cannot deliver to its ultimate destination because it knows of no route to the destination address, it replies to the originator of that datagram with an ICMP host unreachable message.

This command affects a number of ICMP unreachable messages.

For BNG, ensure you run this command in the dynamic template configuration mode. To enter the dynamic template configuration mode, run **dynamic-template** command in the XR EXEC mode.

Task ID	Task ID	Operations
	ipv6	read, write
	network	read, write
	config-services	read, write

**Examples**

This example (not applicable for BNG) shows how to disable the generation of ICMP unreachable messages on HundredGigE interface 0/6/0/0:

```
Router(config)# interface HundredGigE0/6/0/0
Router(config-if)# ipv6 unreachable disable
```

# ipv6 virtual address

To define an IPv6 virtual address for a network of management Ethernet interfaces, use the **ipv6 virtual address** command in XR Config mode. To remove an IPv6 virtual address from the configuration, use the **no** form of this command.

```
ipv6 virtual address {vrf vrf-name ipv6-address / prefix-length | use-as-src-addr}
no ipv6 virtual address {[vrf vrf-name] ipv6-address / prefix-length | use-as-src-addr}
```

<b>Syntax Description</b>	<table> <tr> <td data-bbox="386 573 581 600"><b>vrf vrf-name</b></td><td data-bbox="589 573 1528 667">(Optional) Configures the virtual address on a per VPN routing and forwarding (VRF) basis for the management interfaces. The <i>vrf-name</i> argument specifies the name of the VRF.</td></tr> <tr> <td data-bbox="386 695 581 722"><i>ipv6 address</i></td><td data-bbox="589 695 1528 722">The virtual IPv6 address to be used.</td></tr> <tr> <td data-bbox="386 749 581 777"><i>prefix-length</i></td><td data-bbox="589 749 1528 844">The length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.</td></tr> <tr> <td data-bbox="386 871 581 898"><b>use-as-src-addr</b></td><td data-bbox="589 871 1528 898">Enables the virtual address to be used as the default SRC address on sourced packets.</td></tr> </table>	<b>vrf vrf-name</b>	(Optional) Configures the virtual address on a per VPN routing and forwarding (VRF) basis for the management interfaces. The <i>vrf-name</i> argument specifies the name of the VRF.	<i>ipv6 address</i>	The virtual IPv6 address to be used.	<i>prefix-length</i>	The length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.	<b>use-as-src-addr</b>	Enables the virtual address to be used as the default SRC address on sourced packets.
<b>vrf vrf-name</b>	(Optional) Configures the virtual address on a per VPN routing and forwarding (VRF) basis for the management interfaces. The <i>vrf-name</i> argument specifies the name of the VRF.								
<i>ipv6 address</i>	The virtual IPv6 address to be used.								
<i>prefix-length</i>	The length of the IPv6 prefix. A decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash mark must precede the decimal value.								
<b>use-as-src-addr</b>	Enables the virtual address to be used as the default SRC address on sourced packets.								
<b>Command Default</b>	No IPv6 virtual address is defined for the configuration.								
<b>Command Modes</b>	XR Config mode								
<b>Command History</b>	<table> <tr> <th data-bbox="386 1094 548 1121">Release</th><th data-bbox="557 1094 1528 1121">Modification</th></tr> <tr> <td data-bbox="386 1148 548 1176">Release 7.5.2</td><td data-bbox="557 1148 1528 1176">This release supports virtual addresses for the hosted Linux networking stack.</td></tr> <tr> <td data-bbox="386 1203 548 1230">Release 7.0.12</td><td data-bbox="557 1203 1528 1230">This command was introduced.</td></tr> </table>	Release	Modification	Release 7.5.2	This release supports virtual addresses for the hosted Linux networking stack.	Release 7.0.12	This command was introduced.		
Release	Modification								
Release 7.5.2	This release supports virtual addresses for the hosted Linux networking stack.								
Release 7.0.12	This command was introduced.								
<b>Usage Guidelines</b>	<p>Configuring an IPv6 virtual address enables you to access the router from a single virtual address with a management network. An IPv6 virtual address persists across route processor (RP) failover situations.</p> <p>Configuring an IPv6 virtual address enables you to access a dual RP router from a single address without prior knowledge of which RP is active. An IPv6 virtual address persists across RP failovers. For this to happen, the virtual IPv6 address must share a common IPv6 subnet with a Management Ethernet interface on both RPs.</p> <p>If you disable the <b>ipv6 virtual address</b> command with the <b>vrf</b> keyword, the virtual IP address is unconfigured for the corresponding VRF or for the default if no VRF is specified. This results in the removal of the entry for the virtual IP address in the VRF table and in the ARP cache.</p> <p>The default VRF is chosen when no VRF is specified. The virtual IP address is activated on a management interface that is attached to a default VRF.</p> <p>The <b>use-as-src-addr</b> keyword eliminates the need for configuring a loopback interface as the source interface (that is, update source) for management applications. When an update source is not configured, management applications allow the transport processes (TCP, UDP, raw_ip) to pick a suitable source address. The transport processes, in turn, consult the FIB to do so. If a Management Ethernet's IP address is picked as the source</p>								

address and if the **use-as-src-addr keyword** is configured, then the transport processes replace the Management Ethernet's IP address with a relevant virtual IP address. This functionality works across RP switchovers.

Cisco IOS XR Software Release 7.5.2 and later also supports virtual addresses for the hosted Linux networking stack.

## Task ID

### Task ID Operations

ipv6	read, write
------	----------------

network	read, write
---------	----------------

## Examples

The following example shows how to define an IPv6 virtual address:

```
RP/0/RP0/CPU0:router(config)# ipv6 virtual address 0:0:0:7272::72/64
```

The following example shows how to configure the virtual IP addresses for management interfaces on a per VRF basis:

```
RP/0/RP0/CPU0:router(config)# ipv6 virtual address vrf ppp 0:0:0:7272::72/64
```

# local pool

To create one or more local address pools from which IP addresses are assigned when a peer connects, use the **local pool** command in XR Config mode. To restore the default behavior, use the **no** form of this command.

```
local pool [ipv4] [vrf vrf_name] {poolname | default} first-ip-address [last-ip-address]
no local pool [ipv4] [vrf vrf_name] {poolname | default} first-ip-address [last-ip-address]
```

Syntax Description		
<b>vrf</b>		Specifies that a VRF name will be given. If is parameter is missing, the default VRF is assumed.
<i>vrf_name</i>		Specifies the name of the VRF to which the addresses of the pool belongs. If no name is given, the default VRF is assumed.
<b>default</b>		Creates a default local IPv4 address pool that is used if no other pool is named.
<i>poolname</i>		Specifies the name of the local IPv4 address pool.
<i>first-ip-address</i>		Specifies the first address in an IPv4 address range. If high-IP-address is not specified, the address range is considered to have only one address.
<i>last-ip-address</i>		(Optional) Specifies the last address in an IPv4 address range. If high-IP-address is not specified, the address range is considered to have only one address.

**Command Default** Special default pool if VRF is not specified. By default, this functionality is disabled.

**Command Modes** XR Config mode

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use this command to create local address pools to use in assigning IP addresses when a peer connects. You can also add range of IP addresses to an existing pool. If no pool name is specified, the pool with the name "default" is used.

The optional **vrf** keyword and associated *vrf\_name* allows the association of an IPv4 address pool with a named VRF. Any IPv4 address pool created without the **vrf** keyword automatically becomes a member of a default VRF. An IPv4 address pool name can be associated with only one VRF. Subsequent use of the same pool name, within a pool group, is treated as an extension of that pool, and any attempt to associate an existing local IPv4 address pool name with a different VRF is rejected. Therefore, each use of a pool name is an implicit selection of the associated VRF.



**Note** To reduce the chances of inadvertent generation of duplicate addresses, the system allows creation of the default pool only in the default VRF.

All IPv4 address pools within a VRF are checked to prevent overlapping addresses; however, addresses may overlap across different VRFs.

Task ID	Task ID	Operations
	ipv4	read, write
	ipv6	read, write
	network	read, write

## Examples

The following example creates a local IPv4 address pool named “pool2,” which contains all IPv4 addresses in the range 172.16.23.0 to 172.16.23.255:

```
RP/0/RP0/CPU0:router(config)# local pool ipv4 pool2 172.16.23.0 172.16.23.255
```

The following example configures a pool of 1024 IP addresses:

```
RP/0/RP0/CPU0:router(config)#no local pool ipv4 default
RP/0/RP0/CPU0:router(config)#local pool ipv4 default 10.1.1.0 10.1.4.255
```



**Note** It is good practice to precede local pool definitions with a **no** form of the command to remove any existing pool, because the specification of an existing pool name is taken as a request to extend that pool with the new IPv4 addresses. To extend the pool, the **no** form of the command is not applicable.

The following example configures multiple ranges of IPv4 addresses into one pool:

```
RP/0/RP0/CPU0:router(config)#local pool ipv4 default 10.1.1.0 10.1.9.255
RP/0/RP0/CPU0:router(config)#local pool ipv4 default 10.2.1.0 10.2.9.255
```

The following examples show how to configure two pool groups and IPv4 address pools in the base system group:

```
RP/0/RP0/CPU0:router(config)#local pool vrf grp1 ipv4 p1_g1 10.1.1.1 10.1.1.50
RP/0/RP0/CPU0:router(config)#local pool vrf grp1 ipv4 p2_g1 10.1.1.100 10.1.1.110
RP/0/RP0/CPU0:router(config)#local pool vrf grp2 ipv4 p1_g2 10.1.1.1 10.1.1.40
RP/0/RP0/CPU0:router(config)#local pool ipv4 lp1 10.1.1.1 10.1.1.10
RP/0/RP0/CPU0:router(config)#local pool vrf grp1 ipv4 p3_g1 10.1.2.1 10.1.2.30
RP/0/RP0/CPU0:router(config)#local pool vrf grp2 ipv4 p2_g2 10.1.1.50 10.1.1.70
RP/0/RP0/CPU0:router(config)#local pool ipv4 lp2 10.1.2.1 10.1.2.10
```

In this example:

- VRF grp1 consists of pools p1\_g1, p2\_g1, and p3\_g1.
- VRF grp2 consists of pools p1\_g2 and p2\_g2.
- Pools lp1 and lp2 are not explicitly associated with a vrf and are therefore members of the default vrf.




---

**Note** IPv4 address 10.1.1.1 overlaps in vrfs grp1, grp2 and the default vrf . There is no overlap within any vrf that includes the default vrf.

---

The following examples shows the configurations of IP address pools and groups for use by a VPN and VRF:

```
RP/0/RP0/CPU0:router(config)# local pool vrf vpn1 ipv4 p1_vpn1 10.1.1.1 10.1.1.50
RP/0/RP0/CPU0:router(config)# local pool vrf vpn1 ipv4 p2_vpn1 10.1.1.100 10.1.1.110
RP/0/RP0/CPU0:router(config)# local pool vrf vpn2 ipv4 p1_vpn2 10.1.1.1 10.1.1.40
RP/0/RP0/CPU0:router(config)# local pool ipv4 lp1 10.1.1.1 10.1.1.10
RP/0/RP0/CPU0:router(config)# local pool vrf vpn1 ipv4 p3_vpn1 10.1.2.1 10.1.2.30
RP/0/RP0/CPU0:router(config)# local pool vrf vpn2 ipv4 p2_vpn2 10.1.1.50 10.1.1.70 group
vpn2
RP/0/RP0/CPU0:router(config)# local pool ipv4 lp2 10.1.2.1 10.1.2.10
```

These examples show configuration of pools in two VRFs and the default VRF:

- VRF vpn1 consists of pools p1\_vpn1, p2\_vpn1, and p3\_vpn1.
- VRF vpn2 consists of pools p1\_vpn2 and p2\_vpn2.
- Pools lp1 and lp2 are not associated with a VRF and therefore belong to the default VRF.




---

**Note** IPv4 address 10.1.1.1 overlaps across VRFs vpn1, vpn2 and the default VRF . There is no overlap within any VRF.

---

The VPN requires a configuration that selects the proper vrf by selecting the proper pool based on remote user data. Each user in a given VPN can select an address space using the pool and associated vrf appropriate for that VPN. Duplicate addresses in other VPNs (other vrfs) are not a concern, because the address space of a VPN is specific to that VPN. In the example, a user in VRF vpn1 is associated with a combination of the pools p1\_vpn1, p2\_vpn1, and p3\_vpn1, and is allocated addresses from that address space. Addresses are returned to the same pool from which they were allocated.

# show arm conflicts

To display IPv4 or IPv6 address conflict information identified by the Address Repository Manager (ARM), use the **show arm conflicts** command in XR EXEC mode.

**show arm** {**ipv4** | **ipv6**} [**vrf** *vrf-name*] **conflicts** [{**address** | **override** | **unnumbered**}]

<b>Syntax Description</b>	<b>ipv4</b>	Displays IPv4 address conflicts.
	<b>ipv6</b>	Displays IPv6 address conflicts.
	<b>vrf</b>	(Optional) Displays VPN routing and forwarding (VRF) instance information. Available for IPv4 only.
	<i>vrf-name</i>	(Optional) Name of a VRF.
	<b>address</b>	(Optional) Displays address conflict information.
	<b>override</b>	(Optional) Displays address conflict override information.
	<b>unnumbered</b>	(Optional) Displays unnumbered interface conflict information.
<b>Command Default</b>	None	
<b>Command Modes</b>	XR EXEC mode	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	Use the <b>show arm conflicts</b> command to display information about IPv4 or IPv6 address conflicts. You can use address conflict information to identify misconfigured IPv4 or IPv6 addresses.	
	Conflict information is displayed for interfaces that are forced down and for interfaces that are up.	
	Issuing the <b>show arm conflicts</b> command without specifying any optional keywords displays the output generated from both the <b>address</b> and <b>unnumbered</b> keywords.	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	network	read

## Examples

The following sample output is from the **show arm ipv4 conflicts** command:

```
RP/0/RP0/CPU0:router# show arm ipv4 conflicts

F Forced down
| Down interface & addr                               Up interface & addr

F Lo2 10.1.1.2/24                                       Lo1 10.1.1.1/24
```



```

Forced down interface      Up interface
tu2->tu1                  tu1->Lo1

```

The following is sample output from the **show arm ipv4 conflicts** command with the **address** keyword:

```

RP/0/RP0/CPU0:router# show arm ipv4 conflicts address

F Forced down
| Down interface & addr          Up interface & addr
F Lo2 10.1.1.2/24                Lo1 10.1.1.1/24

```

The following is sample output from the **show arm ipv4 conflicts** command with the **unnumbered** keyword:

```

RP/0/RP0/CPU0:router# show arm ipv4 conflicts unnumbered

Forced down interface      Up interface      VRF
tu2->tu1                  tu1->Lo1

```

This table describes the significant fields shown in the display.

**Table 1: show arm conflicts Command Field Descriptions**

Field	Description
Forced down	Legend defining a symbol that may appear in the output for this command.
Down interface & addr	Forced down interface name, type, and address.
Up interface & addr	List of interfaces that are up.
Forced down interface	Unnumbered interfaces that are in conflict and forced down.
Up interface	Unnumbered interfaces that are in conflict and are up.

# show arm registrations producers

To display producer registration information for the Address Repository Manager (ARM), use the **show arm registrations producers** command in XR EXEC mode.

**show arm {ipv4 | ipv6} registrations producers**

<b>Syntax Description</b>	<b>ipv4</b> Displays IPv4 producer registration information.
	<b>ipv6</b> Displays IPv6 producer registration information.

**Command Default** None

**Command Modes** XR EXEC mode

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines** Use the **show arm registrations producers** command to display information on producers of IP ARM registrations. Registration information is displayed with the ID.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	network	read

**Examples** The following is sample output from the **show arm ipv4 registrations producers** command:

```
Routers# show arm ipv4 registrations producers
```

Id	Node	Producer Id	IPC Version	Connected?
0	0/0/0	ipv4_io	1.1	Y
4	0/1/0	ipv4_io	1.1	Y
3	0/2/0	ipv4_io	1.1	Y
2	0/4/0	ipv4_io	1.1	Y
1	0/6/0	ipv4_io	1.1	Y

This table describes the significant fields shown in the display.

**Table 2: show arm registrations producers Command Field Descriptions**

Field	Description
Id	An identifier used by the IP Address ARM (IP ARM) to keep track of the producer of the IP address.
Node	The physical node (RP/LC CPU) where the producer is running.
Producer Id	The string used by the producer when registering with IP ARM.

Field	Description
IPC Version	Version of the apis used by the producer to communicate with IP ARM.
Connected?	Status of whether the producer is connected or not.

# show arm database

To display IPv4 or IPv6 address information stored in the Address Repository Manager (ARM) database, use the **show arm database** command in XR EXEC mode.

**show arm** {**ipv4** | **ipv6**} [**vrf** {*vrf-name*}] **database** [{**interface** *type interface-path-id* | **network** *prefix / length*}]

<b>Syntax Description</b>	<b>ipv4</b>	Displays IPv4 address information.
	<b>ipv6</b>	Displays IPv6 address information.
	<b>vrf</b>	(Optional) Displays VPN routing and forwarding (VRF) instance information.
	<i>vrf-name</i>	(Optional) Name of a VRF.
	<b>interface</b>	(Optional) Displays the IPv4 or IPv6 address configured on the specified interface.
	<i>type</i>	(Optional) Interface type. For more information, use the question mark (?) online help function.
	<i>interface-path-id</i>	Physical interface or virtual interface.
	<b>Note</b>	Use the show interfaces command to see a list of all interfaces currently configured on the router.
		For more information about the syntax for the router, use the question mark (?) online help function.
	<b>network</b>	(Optional) Displays addresses that match a prefix.
	<i>prefix / length</i>	(Optional) Network prefix and mask. A slash (/) must precede the specified mask. The range is from 0 to 128.
<b>Command Default</b>	None	
<b>Command Modes</b>	XR EXEC mode	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.
<b>Usage Guidelines</b>	The <b>show arm database</b> command should be used to display information in the IP ARM database. Database information is displayed with the IPv4 or IPv6 address, interface type and name, and producer information.	
<b>Task ID</b>	<b>Task ID Operations</b>	
	network	read

## Examples

The following is sample output from the **show arm database** command:

```
RP/0/RP0/CPU0:router# show arm
ipv4 database interface loopback database
Fri Jul 25 10:54:52.304 PST DST

P = Primary, S = Secondary address
|U = Unnumbered
|| Address          Interface          Producer

VRF: default
P 172.29.52.75/24    MgmtEth0/RP0/CPU0/0    ipv4_ma 0/RP0/CPU0    100
P 10.2.2.2/32        Loopback0               ipv4_ma 0/RP1/CPU0
P 10.12.24.2/24      Bundle-POS24            ipv4_ma 0/RP1/CPU0
P 10.12.28.2/24      Bundle-Ether28          ipv4_ma 0/RP1/CPU0
P 10.12.29.2/24      Bundle-Ether28.1        ipv4_ma 0/RP1/CPU0
P 10.12.30.2/24      Bundle-Ether28.2        ipv4_ma 0/RP1/CPU0
P 10.12.31.2/24      Bundle-Ether28.3        ipv4_ma 0/RP1/CPU0
P 10
.1
.1
.s
/24 Loopback1 ipv4_io 0/0/0P 10.1
.1
.1
/24 Loopback1  ipv4_io 0/0/0

| Address          Interface          Producer
P 10.12.16.2/24     GigabitEthernet0/1/5/0    ipv4_ma 0/1/CPU0    1001
P 10.23.4.2/24      GigabitEthernet0/1/5/1    ipv4_ma 0/1/CPU0    1002
P 10.27.4.2/24      GigabitEthernet0/1/5/2    ipv4_ma 0/1/CPU0
P 10.12.8.2/24      POS0/1/0/1               ipv4_ma 0/1/CPU0
P 10.112.4.2/24     POS0/1/0/2               ipv4_ma 0/1/CPU0
P 10.112.8.2/24     POS0/1/0/3               ipv4_ma 0/1/CPU0
P 10.12.32.2/24     POS0/1/4/2               ipv4_ma 0/1/CPU0
P 10.12.32.2/24     POS0/1/4/3               ipv4_ma 0/1/CPU0
P 172.29.52.28/24    MgmtEth0/4/CPU1/0        ipv4_ma 0/4/CPU1
P 172.29.52.27/24    MgmtEth0/4/CPU0/0        ipv4_ma 0/4/CPU0
P 10.12.20.2/24      GigabitEthernet0/6/5/1    ipv4_ma 0/6/CPU0
P 10.4
.1
.4
/24 gigabitethernet 10/0 ipv4_io 1 10
S 10.4.2.4/24        gigabitethernet 10/0    ipv4_io 1 10
S 10.4.3.4/24        gigabitethernet 10/1    ipv4_io 1 10

P = Primary, S = Secondary address

|U = Unnumbered

|| Address          Interface          Producer
VRF: default
P 10.12.12.2/24     POS0/6/0/1             ipv4_ma 0/6/CPU0
P 10.23.8.2/24      POS0/6/4/4             ipv4_ma 0/6/CPU0
P 10.12.4.2/24      POS0/6/4/5             ipv4_ma 0/6/CPU0
P 10.24.4.2/24      POS0/6/4/6             ipv4_ma 0/6/CPU0
P 12
.25.12
.10/16 MgmtEth0/RSP0/CPU0/0  ipv4_ma 0/RSP0/CPU0
```

This table describes the significant fields shown in the display.

**Table 3: show arm database Command Field Descriptions**

Field	Description
Primary	Primary IP address.
Secondary	Secondary IP address.
Unnumbered Address	Interface is unnumbered and the address displayed is that of the referenced interface.
Interface	Interface that has this IP address.
Producer	Process that provides the IP address to the ARM.

# show arm router-ids

To display the router identification information with virtual routing and forwarding table information for the Address Repository Manager (ARM), use the **show arm router-ids** command in XR EXEC mode.

**show arm [ipv4] router-ids**

<b>Syntax Description</b>	<b>ipv4</b> (Optional) Displays IPv4 router information.
---------------------------	--

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	XR EXEC mode
----------------------	--------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

<b>Usage Guidelines</b>	Use the <b>show arm router-ids</b> command with the <b>ipv4</b> keyword to display the selected router ID information for the router.
-------------------------	---

<b>Task ID</b>	<b>Task ID</b> <b>Operations</b>
	network read

## Examples

The following is sample output from the **show arm router-ids** command:

```
RP/0/RP0/CPU0:router# show arm router-ids

Router-ID          Interface
10.10.10.10        Loopback0
```

This table describes the significant fields shown in the display.

**Table 4: show arm router-ids Command Field Descriptions**

Field	Description
Router-ID	Router identification.
Interface	Interface identification.

# show arm summary

To display summary information for the IP Address Repository Manager (ARM), use the **show arm summary** command in XR EXEC mode.

**show arm {ipv4 | ipv6} summary**

<b>Syntax Description</b>	<p><b>ipv4</b> Displays IPv4 summary information.</p> <p><b>ipv6</b> Displays IPv6 summary information.</p>								
<b>Command Default</b>	None								
<b>Command Modes</b>	XR EXEC mode								
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.				
Release	Modification								
Release 7.0.12	This command was introduced.								
<b>Usage Guidelines</b>	Use the <b>show arm summary</b> command to display a summary of the number of producers, address conflicts, and unnumbered interface conflicts in the router.								
<b>Task ID</b>	<table> <tr> <th>Task ID</th><th>Operations</th></tr> <tr> <td>network</td><td>read</td></tr> </table>	Task ID	Operations	network	read				
Task ID	Operations								
network	read								
<b>Examples</b>	<p>The following is sample output from the <b>show arm summary</b> command:</p> <pre>Router# show arm ipv4 summary  IPv4 Producers                :          1 IPv4 address conflicts        :          0 IPv4 unnumbered interface conflicts :          0 IPv4 VRF known                :          0 IPv4 DB Master version        : 0x00000000</pre> <p>This table describes the significant fields shown in the display.</p> <p><b>Table 5: show arm summary Command Field Descriptions</b></p> <table> <tr> <th>Field</th><th>Description</th></tr> <tr> <td>IPv4 Producers</td><td>Number of IPv4 producers on the router.</td></tr> <tr> <td>IPv4 address conflicts</td><td>Number of IPv4 address conflicts on the router.</td></tr> <tr> <td>IPv4 unnumbered interface conflicts</td><td>Number of IPv4 conflicts on unnumbered interfaces.</td></tr> </table>	Field	Description	IPv4 Producers	Number of IPv4 producers on the router.	IPv4 address conflicts	Number of IPv4 address conflicts on the router.	IPv4 unnumbered interface conflicts	Number of IPv4 conflicts on unnumbered interfaces.
Field	Description								
IPv4 Producers	Number of IPv4 producers on the router.								
IPv4 address conflicts	Number of IPv4 address conflicts on the router.								
IPv4 unnumbered interface conflicts	Number of IPv4 conflicts on unnumbered interfaces.								



Field	Description
IPv4 DB Master version	IPv4 DB Master version

# show arm vrf-summary

To display a summary of VPN routing and forwarding (VRF) instance information identified by the Address Repository Manager (ARM), use the **show arm vrf-summary** command in XR EXEC mode.

**show arm {ipv4 | ipv6} vrf-summary**

## Syntax Description

**ipv4** Displays IPv4 address information.

**ipv6** Displays IPv6 address information.

## Command Default

None

## Command Modes

XR EXEC mode

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

## Usage Guidelines

Use the **show arm vrf-summary** command to display information about an IPv4 VPN routing and forwarding instance.

## Task ID

Task ID	Operations
network	read

## Examples

The following example is output from the **show arm vrf-summary** command:

```
RP/0/RP0/CPU0:router# show arm vrf-summary
```

VRF IDs:	VRF-Names:
0x60000000	default
0x60000001	vrf1
0x60000002	vrf2

This table describes the significant fields shown in the display.

**Table 6: show arm vrf-summary Command Field Descriptions**

Field	Description
VRF IDs	VPN routing and forwarding (VRF) identification (vrfid) number.
VRF-Names	Name given to the VRF.

# show clns statistics

To display Connectionless Network Service (CLNS) protocol statistics, use the **show clns statistics** command in XR EXEC mode.

**show clns statistics**

<b>Syntax Description</b>	This command has no keywords or arguments.
---------------------------	--

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	XR EXEC mode
----------------------	--------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

<b>Usage Guidelines</b>	Use this command to display CLNS statistics.
-------------------------	--

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	isis	read

**Examples** The following is sample output from the **show clns statistics** command:

```
RP/0/RP0/CPU0:router# show clns statistics

CLNS Statistics:
Last counter clear:                2868 seconds ago
Total number of packets sent:      0
Total number of packets received:  0
Send packets dropped, buffer overflow: 0
Send packets dropped, out of memory: 0
Send packets dropped, other:       0
Receive socket max queue size:     0
Class      Overflow/Max    Rate Limit/Max
IIH         0/0           0/0
LSP         0/0           0/0
SNP         0/0           0/0
OTHER       0/0           0/0
Total       0             0
```

This table describes the significant fields shown in the display.

**Table 7: show clns traffic Command Field Descriptions**

Field	Description
Class	Indicates the packet type. Packets types are as follows: <ul style="list-style-type: none"><li>• IIH—Intermediate System-to-Intermediate-System hello packets</li><li>• lsp—Link state packets</li><li>• snp—Sequence number packets</li><li>• other</li></ul>
Overflow/Max	Indicates the number of packet drops due to the socket queue being overflowed. The count displays in an <i>x/y</i> format where <i>x</i> indicates the total number of packet drops and <i>y</i> indicates the maximum number of drops in a row.
Rate Limit/Max	Indicates the number of packet drops due to rate limitation. The count displays in an <i>x/y</i> format where <i>x</i> indicates the total number of packet drops and <i>y</i> indicates the maximum number of drops in a row.

# show hw-module local-station-mac

To display status of local station MAC address in the router, use the **show hw-module local-station-mac** command in XR EXEC mode.

**show hw-module local-station-mac**

<b>Syntax Description</b>	This command has no keywords or arguments.
---------------------------	--

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	XR EXEC mode
----------------------	--------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.9.1	This command was introduced.

<b>Usage Guidelines</b>	Use the <b>show hw-module local-station-mac</b> command to display status of the local station MAC address in the router..
-------------------------	--

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	network	read

## Examples

The following example is output from the **show hw-module local-station-mac** command:

```
Router# show hw-module local-station-mac
-----
Knob                               Status      Applied    Action
-----
Local-Station-MAC                 Configured   Yes         None
```

# show ipv4 interface

To display the usability status of interfaces configured for IPv4, use the **show ipv4 interface** command in the XR EXEC mode.

**show ipv4** [**vrf** *vrf-name*] **interface** [{*type interface-path-id* | **brief** | **summary**}]

## Syntax Description

<b>vrf</b>	(Optional) Displays VPN routing and forwarding (VRF) instance information.
<i>vrf-name</i>	(Optional) Name of a VRF.
<i>type</i>	Interface type. For more information, use the question mark (?) online help function.
<i>interface-path-id</i>	<p>Either a physical interface instance or a virtual interface instance as follows:</p> <ul style="list-style-type: none"> <li>Physical interface instance. Naming notation is <i>rack/slot/module/port</i> and a slash between values is required as part of the notation. <ul style="list-style-type: none"> <li><i>rack</i>: Chassis number of the rack.</li> <li><i>slot</i>: Physical slot number of the modular services card or line card.</li> <li><i>module</i>: Module number. A physical layer interface module (PLIM) is always 0.</li> <li><i>port</i>: Physical port number of the interface.</li> </ul> </li> </ul> <p><b>Note</b> In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RSP0 ) and the module is CPU0. Example: interface MgmtEth0/RSP0 /CPU0/0.</p> <ul style="list-style-type: none"> <li>Virtual interface instance. Number range varies depending on interface type.</li> </ul> <p>For more information about the syntax for the router, use the question mark (?) online help function.</p>
<b>brief</b>	(Optional) Displays the primary IPv4 addresses configured on the router's interfaces and their protocol and line states.
<b>summary</b>	(Optional) Displays the number of interfaces on the router that are assigned, unassigned, or unnumbered.

## Command Default

If VRF is not specified, the software displays the default VRF.

## Command Modes

XR EXEC mode

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

**Usage Guidelines**

The **show ipv4 interface** command provides output similar to the **show ipv6 interface** command, except that it is IPv4-specific.

The interface name will be displayed only if the name belongs to the VRF instance. If the *vrf-name* is not specified then the interface instance will be displayed only if the interface belongs to the default VRF.

**Task ID****Task ID Operations**

ipv4	read
------	------

network	read
---------	------

**Examples**

This is the sample output of the **show ipv4 interface** command:

```
RP/0/RP0/CPU0:router# show ipv4 interface

Bundle-Ether1 is Down, ipv4 protocol is Down
  Vrf is default (vrfid 0x60000000)
  Internet address is 40.30.1.2/24
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound common access list is not set, access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachable are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
Bundle-Ether2 is Down, ipv4 protocol is Down
  Vrf is default (vrfid 0x60000000)
  Internet address is 40.30.2.2/24
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound common access list is not set, access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachable are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
Bundle-Ether10 is Shutdown, ipv4 protocol is Down
  Vrf is default (vrfid 0x60000000)
  Internet protocol processing disabled
Bundle-Ether54 is Up, ipv4 protocol is Up
  Vrf is default (vrfid 0x60000000)
  Internet address is 10.0.9.0/31
  MTU is 1514 (1500 is available to IP)
  Helper address is not set
  Multicast reserved groups joined: 224.0.0.2 224.0.0.1 224.0.0.2
  Directed broadcast forwarding is disabled
  Outgoing access list is not set
  Inbound common access list is not set, access list is not set
  Proxy ARP is disabled
  ICMP redirects are never sent
  ICMP unreachable are always sent
  ICMP mask replies are never sent
  Table Id is 0xe0000000
Bundle-Ether1900 is Down, ipv4 protocol is Down
```

```

Vrf is default (vrfid 0x60000000)
Internet address is 10.0.54.1/30
MTU is 9000 (8986 is available to IP)
Helper address is not set
Directed broadcast forwarding is disabled
Outgoing access list is not set
Inbound common access list is not set, access list is not set
Proxy ARP is disabled
ICMP redirects are never sent
ICMP unreachable are always sent
ICMP mask replies are never sent
Table Id is 0xe0000000
Bundle-Ether1901 is Down, ipv4 protocol is Down
Vrf is default (vrfid 0x60000000)
Internet address is 10.0.55.1/30
MTU is 9000 (8986 is available to IP)

```

This table describes the significant fields shown in the display.

**Table 8: show ipv4 interface Command Field Descriptions**

Field	Description
Loopback0 is Up	If the interface hardware is usable, the interface is marked “Up.” For an interface to be usable, both the interface hardware and line protocol must be up.
line protocol is Up	If the interface can provide two-way communication, the line protocol is marked “Up.” For an interface to be usable, both the interface hardware and line protocol must be up.
Internet address	IPv4 Internet address and subnet mask of the interface.
Secondary address	Displays a secondary address, if one has been set.
MTU	Displays the IPv4 MTU <sup>1</sup> value set on the interface.
Multicast reserved groups joined	Indicates the multicast groups this interface belongs to.
Directed broadcast forwarding	Indicates whether directed broadcast forwarding is enabled or disabled.
Outgoing access list	Indicates whether the interface has an outgoing access list set.
Inbound access list	Indicates whether the interface has an incoming access list set.
Proxy ARP	Indicates whether proxy ARP <sup>2</sup> is enabled or disabled on an interface.
ICMP redirects	Specifies whether ICMPv4 <sup>3</sup> redirects are sent on this interface.
ICMP unreachable	Specifies whether unreachable messages are sent on this interface.
Internet protocol processing disabled	Indicates an IPv4 address has not been configured on the interface.

<sup>1</sup> MTU = maximum transmission unit

<sup>2</sup> ARP = Address Resolution Protocol address resolution protocol

<sup>3</sup> ICMPv4 = Internet Control Message Protocol internet control message protocol version 4



# show ipv4 traffic

To display the IPv4 traffic statistics, use the **show ipv4 traffic** command in the XR EXEC mode.

**show ipv4 traffic [brief]**

<b>Syntax Description</b>	<b>brief</b> (Optional) Displays only IPv4 and Internet Control Message Protocol version 4 (ICMPv4) traffic.						
<b>Command Default</b>	None						
<b>Command Modes</b>	XR EXEC mode						
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.		
Release	Modification						
Release 7.0.12	This command was introduced.						
<b>Usage Guidelines</b>	The <b>show ipv4 traffic</b> command provides output similar to the <b>show ipv6 traffic</b> command, except that it is IPv4-specific.						
<b>Task ID</b>	<table> <tr> <th>Task ID</th><th>Operations</th></tr> <tr> <td>ipv4</td><td>read</td></tr> <tr> <td>network</td><td>read</td></tr> </table>	Task ID	Operations	ipv4	read	network	read
Task ID	Operations						
ipv4	read						
network	read						
<b>Examples</b>	<p>This is the sample output of the <b>show ipv4 traffic</b> command:</p> <pre>RP/0/RP0/CPU0:router# show ipv4 traffic  IP statistics:   Rcvd:  486522 total, 55292 local destination          0 format errors, 0 bad hop count          0 unknown protocol, 0 not a gateway          0 security failures, 0 bad source, 0 bad header          842 with options, 0 bad, 0 unknown   Opts:  0 end, 0 nop, 0 basic security, 0 extended security          0 strict source rt, 0 loose source rt, 0 record rt          0 stream ID, 0 timestamp, 842 alert, 0 cipso   Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble, 0 fragments received          0 fragmented, 0 fragment count, 0 fragment max drop   Bcast: 0 sent, 0 received   Mcast: 13042 sent, 417434 received   Lisp:  0 encapsulated in v4, 0 decapped from v4          0 encapsulated in v6, 0 decapped from v6          0 encap errors, 0 decap errors   Drop:  0 encapsulation failed, 19 no route, 0 too big   Sent:  446780 total  ICMP statistics:   Sent:  0 admin unreachable, 190147 network unreachable          0 host unreachable, 0 protocol unreachable</pre>						

```

0 port unreachable, 0 fragment unreachable
0 time to live exceeded, 0 reassembly ttl exceeded
0 echo request, 0 echo reply
0 mask request, 0 mask reply
0 parameter error, 0 redirects
190147 total
Rcvd: 0 admin unreachable, 11 network unreachable
0 host unreachable, 0 protocol unreachable
0 port unreachable, 0 fragment unreachable
0 time to live exceeded, 0 reassembly ttl exceeded
0 echo request, 0 echo reply
0 mask request, 0 mask reply
0 redirect, 0 parameter error
0 source quench, 0 timestamp, 0 timestamp reply
0 router advertisement, 0 router solicitation
11 total, 0 checksum errors, 0 unknown

UDP statistics:
424354 packets input, 10881 packets output
0 checksum errors, 13236 no port
0 forwarded broadcasts

TCP statistics:
53775 packets input, 56104 packets output
0 checksum errors, 0 no port

```

This table describes the significant fields shown in the display.

**Table 9: show ipv4 traffic Command Field Descriptions**

Field	Description
bad hop count	Occurs when a packet is discarded because its TTL <sup>4</sup> field was decremented to zero.
encapsulation failed	Usually indicates that the router had no ARP request entry and therefore did not send a datagram.
format errors	Indicates a gross error in the packet format, such as an impossible Internet header length.
IP statistics Rcvd total	Indicates the total number of local destination and other packets received in the software plane. It does not account for the IP packets forwarded or discarded in hardware.
no route	Counted when the Cisco IOS XR software discards a datagram it did not know how to route.

<sup>4</sup> TTL = time-to-live

# show ipv6 interface

To display the usability status of interfaces configured for IPv6, use the **show ipv6 interface** command in the XR EXEC mode.

**show ipv6** [**vrf** *vrf-name*] **interface** [{**summary** | [*type* *interface-path-id*][**brief** [{**link-local** | **global**}]]}]

Syntax Description	
<b>vrf</b>	(Optional) Displays VPN routing and forwarding (VRF) instance information.
<i>vrf-name</i>	(Optional) Name of a VRF.
<i>type</i>	(Optional) Interface type. For more information, use the question mark (?) online help function.
<i>interface-path-id</i>	(Optional) Either a physical interface instance or a virtual interface instance as follows: <ul style="list-style-type: none"> <li>Physical interface instance. Naming notation is <i>rack/slot/module/port</i> and a slash between values is required as part of the notation.               <ul style="list-style-type: none"> <li><i>rack</i>: Chassis number of the rack.</li> <li><i>slot</i>: Physical slot number of the modular services card or line card.</li> <li><i>module</i>: Module number. A physical layer interface module (PLIM) is always 0.</li> <li><i>port</i>: Physical port number of the interface.</li> </ul> </li> </ul> <p><b>Note</b> In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RSP0) and the module is CPU0. Example: interface MgmtEth0/RSP0 /CPU0/0.</p> <ul style="list-style-type: none"> <li>Virtual interface instance. Number range varies depending on interface type.</li> </ul> <p>For more information about the syntax for the router, use the question mark (?) online help function.</p>
<b>brief</b>	(Optional) Displays the primary IPv6 addresses configured on the router interfaces and their protocol and line states.
<b>link-local</b>	(Optional) Displays the link local IPv6 address.
<b>global</b>	(Optional) Displays the global IPv6 address.
<b>summary</b>	(Optional) Displays the number of interfaces on the router that are assigned, unassigned, or unnumbered.
<b>Command Default</b>	None
<b>Command Modes</b>	XR EXEC mode

## show ipv6 interface

## Command History

Release	Modification
Release 7.0.12	This command was introduced.

## Usage Guidelines

The **show ipv6 interface** command provides output similar to the **show ipv4 interface** command, except that it is IPv6-specific.

Use the **link-local** or **global** keywords along with the **brief** keyword to view the link local or global IPv6 addresses.

## Task ID

Task ID	Operations
ipv6	read

## Examples

This is the sample output of the **show ipv6 interface** command:

```
RP/0/RP0/CPU0:router# show ipv6 interface
Bundle-Ether1 is Down, ipv6 protocol is Down, Vrfid is default (0x60000000)
IPv6 is enabled, link-local address is fe80::e448:5cff:fe10:b484 [TENTATIVE]
Global unicast address(es):
  40:30:1:1::2, subnet is 40:30:1:1::/64 [TENTATIVE]
Joined group address(es): ff02::2 ff02::1
MTU is 1514 (1500 is available to IPv6)
ICMP redirects are disabled
ICMP unreachable are enabled
ND DAD is enabled, number of DAD attempts 1
ND reachable time is 0 milliseconds
ND cache entry limit is 1000000000
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 160 to 240 seconds
ND router advertisements live for 1800 seconds
Hosts use stateless autoconfig for addresses.
Outgoing access list is not set
Inbound common access list is not set, access list is not set
Table Id is 0xe0800000
Complete protocol adjacency: 0
Complete glean adjacency: 0
Incomplete protocol adjacency: 0
Incomplete glean adjacency: 0
Dropped protocol request: 0
Dropped glean request: 0
```

This table describes the significant fields shown in the display.

**Table 10: show ipv6 interface Command Field Descriptions**

Field	Description
Bundle-Ether1 is Down	Indicates whether the interface hardware is currently active (whether line signal is present) and whether it has been taken down by an administrator. If the interface hardware is usable, the interface is marked “Up.” For an interface to be usable, both the interface hardware and line protocol must be up.

Field	Description
line protocol is Up (or down)	Indicates whether the software processes that handle the line protocol consider the line usable (that is, whether keepalives are successful). If the interface can provide two-way communication, the line protocol is marked “Up.” For an interface to be usable, both the interface hardware and line protocol must be up.
IPv6 is enabled, stalled, disabled (stalled and disabled are not shown in sample output)	Indicates that IPv6 is enabled, stalled, or disabled on the interface. If IPv6 is enabled, the interface is marked “enabled.” If duplicate address detection processing identified the link-local address of the interface as being a duplicate address, the processing of IPv6 packets is disabled on the interface and the interface is marked “stalled.” If IPv6 is not enabled, the interface is marked “disabled.”
link-local address	Displays the link-local address assigned to the interface.
TENTATIVE	<p>The state of the address in relation to duplicate address detection. States can be any of the following:</p> <ul style="list-style-type: none"> <li>• duplicate—The address is not unique and is not being used. If the duplicate address is the link-local address of an interface, the processing of IPv6 packets is disabled on that interface.</li> <li>• tentative—Duplicate address detection is either pending or under way on this interface.</li> </ul> <p><b>Note</b> If an address does not have one of these states (the state for the address is blank), the address is unique and is being used.</p>
Global unicast addresses	Displays the global unicast addresses assigned to the interface.
ICMP redirects	State of Internet Control Message Protocol (ICMP) IPv6 redirect messages on the interface (the sending of the messages is enabled or disabled).
ND DAD	State of duplicate address detection on the interface (enabled or disabled).
number of DAD attempts	Number of consecutive neighbor solicitation messages that are sent on the interface while duplicate address detection is performed.
ND reachable time	Displays the neighbor discovery reachable time (in milliseconds) assigned to this interface.

This is the sample output of the **show ipv6 interface brief link-local** command:

```
RP/0/RP0/CPU0:router#show ipv6 interface brief link-local
```

Interface	IPv6-Address	Status	Protocol
Bundle-Ether1	fe80::e448:5cff:fe10:b484	Down	Down
Bundle-Ether2	fe80::e448:5cff:fe10:b483	Down	Down
Bundle-Ether10	unassigned	Shutdown	Down
Bundle-Ether54	fe80::e448:5cff:fe10:b481	Up	Up
Bundle-Ether1900	fe80::e448:5cff:fe10:b48a	Down	Down
Bundle-Ether1901	fe80::e448:5cff:fe10:b489	Down	Down
Bundle-Ether1902	fe80::e448:5cff:fe10:b488	Down	Down

**show ipv6 interface**

Bundle-Ether1903	fe80::e448:5cff:fe10:b487	Down	Down
Bundle-Ether1904	fe80::e448:5cff:fe10:b486	Down	Down
Bundle-Ether1905	unassigned	Shutdown	Down
Bundle-Ether1906	fe80::e448:5cff:fe10:b48e	Down	Down
Loopback0	fe80::9d4c:a5ff:fe2f:2615	Up	Up
Loopback1	fe80::9d4c:a5ff:fe2f:2615	Up	Up
tunnel-te54	unassigned	Down	Down
tunnel-te718	unassigned	Up	Up
tunnel-te720	unassigned	Up	Up
tunnel-te5454	unassigned	Up	Up
MgmtEth0/RP0/CPU0/0	unassigned	Up	Up
HundredGigE0/2/0/0	unassigned	Shutdown	Down
HundredGigE0/2/0/1	unassigned	Shutdown	Down
HundredGigE0/2/0/2	unassigned	Shutdown	Down
HundredGigE0/2/0/3	unassigned	Shutdown	Down
HundredGigE0/2/0/4	fe80::e448:5cff:fe10:b130	Shutdown	Down
HundredGigE0/2/0/5	unassigned	Shutdown	Down
HundredGigE0/2/0/6	unassigned	Shutdown	Down
HundredGigE0/2/0/7	unassigned	Shutdown	Down
HundredGigE0/2/0/8	unassigned	Down	Down
HundredGigE0/2/0/9	unassigned	Shutdown	Down
HundredGigE0/2/0/10	unassigned	Shutdown	Down
HundredGigE0/2/0/11	unassigned	Shutdown	Down
HundredGigE0/2/0/12	unassigned	Shutdown	Down
HundredGigE0/2/0/13	unassigned	Shutdown	Down
HundredGigE0/2/0/15	unassigned	Shutdown	Down
HundredGigE0/2/0/16	unassigned	Shutdown	Down
HundredGigE0/2/0/17	unassigned	Shutdown	Down
HundredGigE0/2/0/18	unassigned	Shutdown	Down
HundredGigE0/2/0/19	unassigned	Shutdown	Down
HundredGigE0/2/0/20	unassigned	Shutdown	Down
HundredGigE0/2/0/21	unassigned	Shutdown	Down
HundredGigE0/2/0/22	unassigned	Shutdown	Down
HundredGigE0/2/0/23	unassigned	Shutdown	Down
HundredGigE0/2/0/25	fe80::e448:5cff:fe10:b184	Shutdown	Down
HundredGigE0/2/0/26	unassigned	Shutdown	Down
HundredGigE0/2/0/27	unassigned	Shutdown	Down
HundredGigE0/2/0/28	unassigned	Shutdown	Down
HundredGigE0/2/0/29	unassigned	Shutdown	Down
HundredGigE0/2/0/31	unassigned	Shutdown	Down
HundredGigE0/2/0/32	unassigned	Shutdown	Down
HundredGigE0/2/0/33	unassigned	Shutdown	Down
HundredGigE0/2/0/34	unassigned	Shutdown	Down
HundredGigE0/2/0/35	unassigned	Shutdown	Down
TenGigE0/2/0/14/0	unassigned	Up	Up
TenGigE0/2/0/14/1	unassigned	Up	Up
TenGigE0/2/0/14/2	unassigned	Up	Up
TenGigE0/2/0/14/3	unassigned	Up	Up
TenGigE0/2/0/24/0	fe80::e448:5cff:fe10:b180	Up	Up

This is the sample output of the **show ipv6 interface brief global** command:

RP/0/#**show ipv6 interface brief global**

Interface	IPv6-Address	Status	Protocol
Bundle-Ether54	10:0:9::2	Up	Up
Bundle-Ether1900	10:0:54::2	Up	Up
Bundle-Ether1901	10:0:55::2	Up	Up
Bundle-Ether1902	10:0:56::2	Up	Up
Bundle-Ether1903	10:0:84::2	Up	Up
Bundle-Ether1904	10:0:85::2	Up	Up
Bundle-Ether1906	10:0:86::2	Up	Up

This is the sample output of the **show ipv6 interface type interface-path-id brief link-local** command:

```
RP/0/RP0/CPU0:router#show ipv6 interface tenGigE 0/0/0/0 brief link-local
```

Interface	IPv6-Address	Status	Protocol
HundredGigE0/0/0/0	fe80::fe:8ff:feeb:26c5	Up	Up

This is the sample output of the **show ipv6 interface *type interface-path-id* brief global** command:

```
RP/0/RP0/CPU0:router#show ipv6 interface tenGigE 0/0/0/0 brief global
```

Interface	IPv6-Address	Status	Protocol
HundredGigE0/0/0/0	2001:db8::1	Up	Up

# show ipv6 neighbors

To display the IPv6 neighbor discovery cache information, use the **show ipv6 neighbors** command in the XR EXEC mode.

**show ipv6 neighbors** [{*type interface-path-id* | **location** *node-id*}]

<b>Syntax Description</b>	<i>type</i>	(Optional) Interface type. For more information, use the question mark (?) online help function.
	<i>interface-path-id</i>	(Optional) Physical interface instance or a virtual interface.
	<b>Note</b>	Use the <b>show interfaces</b> command to see a list of all interfaces currently configured on the router.
		For more information about the syntax for the router, use the question mark (?) online help function.
	<b>location</b> <i>node-id</i>	(Optional) Designates a node. The <i>node-id</i> argument is entered in the <i>rack/slot/module</i> notation.

**Command Default** All IPv6 neighbor discovery cache information is displayed.

**Command Modes** XR EXEC mode

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Usage Guidelines** When the *interface-type* and *interface-number* arguments are not specified, cache information for all IPv6 neighbors is displayed. Specifying the *interface-type* and *interface-number* arguments displays only cache information about the specified interface.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv6	read

## Examples

This is the sample output of the **show ipv6 neighbors** command when entered with an interface type and number:

```
RP/0/RP0/CPU0:router# show ipv6 neighbors HundredGigE0/0/0/2
```

```
IPv6 Address                               Age Link-layer Addr State Interface
2000:0:0:4::2                             0 0003.a0d6.141e REACH tenGigE
FE80::203:A0FF:FED6:141E                  0 0003.a0d6.141e REACH tenGigE
3001:1::45a                               - 0002.7d1a.9472 REACH tenGigE
```



This is the sample output of the **show ipv6 neighbors** command:

```
RP/0/RP0/CPU0:router# show ipv6 neighbors
```

IPv6 Address	Age	Link-layer Addr	State	Interface
Location				
[Mcast adjacency]	-	0000.0000.0000	DELETE	Hu0/2/0/25
0/2/CPU0				
[Mcast adjacency]	-	0000.0000.0000	DELETE	Hu0/2/0/4
0/2/CPU0				
[Mcast adjacency]	-	0000.0000.0000	DELETE	Te0/2/0/30/3
0/2/CPU0				
[Mcast adjacency]	-	0000.0000.0000	REACH	Te0/2/0/30/2
0/2/CPU0				
[Mcast adjacency]	-	0000.0000.0000	REACH	Te0/2/0/30/1
0/2/CPU0				
fe80::d66d:50ff:fe38:9544	97	d46d.5038.9544	REACH	Te0/2/0/30/0
0/2/CPU0				
[Mcast adjacency]	-	0000.0000.0000	REACH	Te0/2/0/30/0
0/2/CPU0				
10:0:8::2	89	10f3.114c.719c	REACH	Te0/2/0/24/0
0/2/CPU0				
fe80::12f3:11ff:fe4c:719c	135	10f3.114c.719c	REACH	Te0/2/0/24/0
0/2/CPU0				
[Mcast adjacency]	-	0000.0000.0000	REACH	Te0/2/0/24/0
0/2/CPU0				
10:0:9::2	150	e607.2b8d.3484	REACH	BE54
0/2/CPU0				
fe80::e407:2bff:fe8d:3484	149	e607.2b8d.3484	REACH	BE54
0/2/CPU0				
[Mcast adjacency]	-	0000.0000.0000	REACH	BE54
0/2/CPU0				
[Mcast adjacency]	-	0000.0000.0000	DELETE	BE1900
0/2/CPU0				
[Mcast adjacency]	-	0000.0000.0000	DELETE	BE1901
0/2/CPU0				
[Mcast adjacency]	-	0000.0000.0000	DELETE	BE1903
0/2/CPU0				
[Mcast adjacency]	-	0000.0000.0000	DELETE	BE1904
0/2/CPU0				
1000::2	50	0010.9400.000d	REACH	Hu0/4/0/0
0/4/CPU0				
fe80::1	153	0010.9400.000d	REACH	Hu0/4/0/0
0/4/CPU0				
[Mcast adjacency]	-	0000.0000.0000	REACH	Hu0/4/0/0
0/4/CPU0				
[Mcast adjacency]	-	0000.0000.0000	DELETE	Hu0/4/0/6
0/4/CPU0				
[Mcast adjacency]	-	0000.0000.0000	DELETE	Hu0/4/0/18
0/4/CPU0				
[Mcast adjacency]	-	0000.0000.0000	DELETE	Hu0/4/0/25
0/4/CPU0				
[Mcast adjacency]	-	0000.0000.0000	REACH	Te0/4/0/30/0
0/4/CPU0				
[Mcast adjacency]	-	0000.0000.0000	REACH	Te0/4/0/30/1
0/4/CPU0				
[Mcast adjacency]	-	0000.0000.0000	DELETE	BE1901
0/4/CPU0				
[Mcast adjacency]	-	0000.0000.0000	DELETE	BE1902
0/4/CPU0				
[Mcast adjacency]	-	0000.0000.0000	DELETE	BE1903
0/4/CPU0				

## show ipv6 neighbors

```

[Mcast adjacency]          - 0000.0000.0000 DELETE BE1906
0/4/CPU0
[Mcast adjacency]          - 0000.0000.0000 DELETE Hu0/6/0/35
0/6/CPU0
200:1::2                   157 0010.9400.0013 REACH Hu0/6/0/34
0/6/CPU0
fe80::1                   130 0010.9400.0013 REACH Hu0/6/0/34
0/6/CPU0
[Mcast adjacency]          - 0000.0000.0000 REACH Hu0/6/0/34
0/6/CPU0
[Mcast adjacency]          - 0000.0000.0000 DELETE Hu0/6/0/16
0/6/CPU0
[Mcast adjacency]          - 0000.0000.0000 DELETE Hu0/6/0/18
0/6/CPU0
[Mcast adjacency]          - 0000.0000.0000 DELETE Hu0/6/0/19
0/6/CPU0
[Mcast adjacency]          - 0000.0000.0000 DELETE Hu0/6/0/20
0/6/CPU0
[Mcast adjacency]          - 0000.0000.0000 DELETE Hu0/6/0/21
0/6/CPU0
[Mcast adjacency]          - 0000.0000.0000 DELETE Te0/6/0/2/2
0/6/CPU0
[Mcast adjacency]          - 0000.0000.0000 DELETE Te0/6/0/2/1
0/6/CPU0
[Mcast adjacency]          - 0000.0000.0000 DELETE BE2
0/6/CPU0
[Mcast adjacency]          - 0000.0000.0000 DELETE BE1900
0/6/CPU0
[Mcast adjacency]          - 0000.0000.0000 DELETE BE1902
0/6/CPU0
[Mcast adjacency]          - 0000.0000.0000 DELETE BE1904
0/6/CPU0
[Mcast adjacency]          - 0000.0000.0000 DELETE BE1906
0/6/CPU0

```

This is the sample output of the **show ipv6 neighbors** command when entered with a location:

```
RP/0/RP0/CPU0:router# show ipv6 neighbors location 0/2/CPU0
```

IPv6 Address	Age	Link-layer Addr	State	Interface	Location
2001:3::2	119	0013.9400.0002	REACH	BE3	0/2/CPU0
2001:3::3	179	0013.9400.0003	DELAY	BE3	0/2/CPU0
2001:3::4	166	0013.9400.0004	REACH	BE3	0/2/CPU0
2001:3::5	78	0013.9400.0005	REACH	BE3	0/2/CPU0
2001:3::6	19	0013.9400.0006	REACH	BE3	0/2/CPU0
2001:3::7	173	0013.9400.0007	REACH	BE3	0/2/CPU0
2001:3::8	140	0013.9400.0008	REACH	BE3	0/2/CPU0
2001:3::9	163	0013.9400.0009	REACH	BE3	0/2/CPU0
2001:3::a	40	0013.9400.000a	REACH	BE3	0/2/CPU0
2001:3::b	90	0013.9400.000b	REACH	BE3	0/2/CPU0
2001:3::c	35	0013.9400.000c	REACH	BE3	0/2/CPU0
2001:3::d	114	0013.9400.000d	REACH	BE3	0/2/CPU0
2001:3::e	117	0013.9400.000e	REACH	BE3	0/2/CPU0
2001:3::f	157	0013.9400.000f	REACH	BE3	0/2/CPU0
2001:3::10	9	0013.9400.0010	REACH	BE3	0/2/CPU0
2001:3::11	120	0013.9400.0011	REACH	BE3	0/2/CPU0
2001:3::12	87	0013.9400.0012	REACH	BE3	0/2/CPU0
2001:3::13	180	0013.9400.0013	DELAY	BE3	0/2/CPU0
2001:3::14	103	0013.9400.0014	REACH	BE3	0/2/CPU0
2001:3::15	132	0013.9400.0015	REACH	BE3	0/2/CPU0
2001:3::16	33	0013.9400.0016	REACH	BE3	0/2/CPU0
2001:3::17	150	0013.9400.0017	REACH	BE3	0/2/CPU0
2001:3::18	117	0013.9400.0018	REACH	BE3	0/2/CPU0

```

2001:3::19      48    0013.9400.0019 REACH BE3      0/2/CPU0
2001:3::1a      67    0013.9400.001a REACH BE3      0/2/CPU0
2001:3::1b      91    0013.9400.001b REACH BE3      0/2/CPU0
2001:3::1c      33    0013.9400.001c REACH BE3      0/2/CPU0
2001:3::1d     174    0013.9400.001d DELAY BE3      0/2/CPU0
2001:3::1e     144    0013.9400.001e REACH BE3      0/2/CPU0
2001:3::1f     121    0013.9400.001f REACH BE3      0/2/CPU0
2001:3::20      53    0013.9400.0020 REACH BE3      0/2/CPU0

```

This table describes significant fields shown in the display.

**Table 11: show ipv6 neighbors Command Field Descriptions**

Field	Description
IPv6 Address	IPv6 address of neighbor or interface.
Age	Time (in minutes) since the address was confirmed to be reachable. A hyphen (-) indicates a static entry.
Link-layer Addr	MAC address. If the address is unknown, a hyphen (-) is displayed.
State	<p>The state of the neighbor cache entry. These are the states for dynamic entries in the IPv6 neighbor discovery cache:</p> <ul style="list-style-type: none"> <li>• INCMP (incomplete)—Address resolution is being performed on the entry. A neighbor solicitation message has been sent to the solicited-node multicast address of the target, but the corresponding neighbor advertisement message has not yet been received.</li> <li>• reach (reachable)—Positive confirmation was received within the last ReachableTime milliseconds that the forward path to the neighbor was functioning properly. While in reach state, the device takes no special action as packets are sent.</li> <li>• stale—More than ReachableTime milliseconds have elapsed since the last positive confirmation was received that the forward path was functioning properly. While in stale state, the device takes no action until a packet is sent.</li> <li>• delay—More than ReachableTime milliseconds have elapsed since the last positive confirmation was received that the forward path was functioning properly. A packet was sent within the last DELAY_FIRST_PROBE_TIME seconds. If no reachability confirmation is received within DELAY_FIRST_PROBE_TIME seconds of entering the delay state, send a neighbor solicitation message and change the state to probe.</li> <li>• probe—A reachability confirmation is actively sought by resending neighbor solicitation messages every RetransTimer milliseconds until a reachability confirmation is received.</li> </ul> <p>These are the possible states for static entries in the IPv6 neighbor discovery cache:</p> <ul style="list-style-type: none"> <li>• reach (reachable)—The interface for this entry is up.</li> <li>• INCMP (incomplete)—The interface for this entry is down.</li> </ul> <p><b>Note</b> Reachability detection is not applied to static entries in the IPv6 neighbor discovery cache; therefore, the descriptions for the INCMP (incomplete) and reach (reachable) states are different for dynamic and static cache entries.</p>
Interface	Interface from which the address is reachable.

# show ipv6 neighbors summary

To display summary information for the neighbor entries, use the **show ipv6 neighbors summary** command in the XR EXEC mode.

**show ipv6 neighbors summary**

<b>Syntax Description</b>	This command has no keywords or arguments.
---------------------------	--

<b>Command Default</b>	The default value is disabled.
------------------------	--------------------------------

<b>Command Modes</b>	XR EXEC mode
----------------------	--------------

Command History	Release	Modification
	Release 7.0.12	This command was introduced.

Task ID	Task ID	Operations
	ipv6	read

**Examples**

This is the sample output of the **show ipv6 neighbors summary** command that shows the summary information for the neighbor entries:

```
XR EXEC mode# show ipv6 neighbors summary
```

```
Mcast nbr entries:
  Subtotal: 0
Static nbr entries:
  Subtotal: 0
Dynamic nbr entries:
  Subtotal: 0
```

```
Total nbr entries: 0
```

# show ipv6 path-mtu

To display path maximum transmission unit (MTU) details of IPv6 packets, use the **show ipv6 path-mtu** command in the XR Config mode.

**show ipv6 path-mtu** [ **vrf** { *vrf-name* | **all** } ] [ **location** *node-id* ] [ **location** *node-id* ]

<b>Syntax Description</b>	<b>location</b> <i>node-id</i> (Optional) The designated node. The node-id argument is entered in the <i>rack/slot/module</i> notation.								
<b>Command Default</b>	None.								
<b>Command Modes</b>	XR Config mode								
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.				
Release	Modification								
Release 7.0.12	This command was introduced.								
<b>Usage Guidelines</b>	If the location option is specified, only the details of the node specified in the <b>location</b> <i>node-id</i> keyword and argument are displayed. Path MTU discovery for IPv6 packets is supported only for applications using TCP and Ping protocol.								
<b>Task ID</b>	<table> <tr> <th>Task ID</th><th>Operations</th></tr> <tr> <td>ipv6</td><td>read, write</td></tr> <tr> <td>network</td><td>read, write</td></tr> <tr> <td>config-services</td><td>read, write</td></tr> </table>	Task ID	Operations	ipv6	read, write	network	read, write	config-services	read, write
Task ID	Operations								
ipv6	read, write								
network	read, write								
config-services	read, write								

## Examples

This example shows how to display path MTU details of IPv6 packets:

```
RP/0/RP0/CPU0:router(config)# show ipv6 pmtu
```

Destination	Ifhandle	Vrfid	Path Mtu	Time Left
bb::1	0x300	0x60000000	1300	00:01:27
cd::1	0x300	0x60000000	1300	00:01:42

# show ipv6 traffic

To display the IPv6 traffic statistics, use the **show traffic** command in the XR EXEC mode.

**show ipv6 traffic [brief]**

<b>Syntax Description</b>	<b>brief</b> (Optional) Displays only IPv6 and Internet Control Message Protocol version 6 (ICMPv6) traffic statistics.						
<b>Command Default</b>	None						
<b>Command Modes</b>	XR EXEC mode						
<b>Command History</b>	<table> <tr> <th>Release</th><th>Modification</th></tr> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </table>	Release	Modification	Release 7.0.12	This command was introduced.		
Release	Modification						
Release 7.0.12	This command was introduced.						
<b>Usage Guidelines</b>	The <b>show ipv6 traffic</b> command provides output similar to the <b>show ipv4 traffic</b> command, except that it is IPv6-specific.						
<b>Task ID</b>	<table> <tr> <th>Task ID</th><th>Operations</th></tr> <tr> <td>ipv6</td><td>read</td></tr> <tr> <td>network</td><td>read</td></tr> </table>	Task ID	Operations	ipv6	read	network	read
Task ID	Operations						
ipv6	read						
network	read						

## Examples

This is the sample output of the **show ipv6 traffic** command:

```
RP/0/RP0/CPU0:router# show ipv6 traffic

IPv6 statistics:
  Rcvd:  0 total, 0 local destination
         0 source-routed, 0 truncated
         0 format errors, 0 hop count exceeded
         0 bad header, 0 unknown option, 0 bad source
         0 unknown protocol
         0 fragments, 0 total reassembled
         0 reassembly timeouts, 0 reassembly failures
         0 reassembly max drop
         0 sanity address check drops
  Sent:  0 generated, 0 forwarded
         0 fragmented into 0 fragments, 0 failed
         0 no route, 0 too big
  Mcast: 0 received, 0 sent

ICMP statistics:
  Rcvd: 0 input, 0 checksum errors, 0 too short
         0 unknown error type
  unreach: 0 routing, 0 admin, 0 neighbor,
           0 address, 0 port, 0 unknown
```

```

parameter: 0 error, 0 header, 0 option,
           0 unknown
0 hopcount expired, 0 reassembly timeout,
0 unknown timeout, 0 too big,
0 echo request, 0 echo reply
Sent: 0 output, 0 rate-limited
unreach: 0 routing, 0 admin, 0 neighbor,
         0 address, 0 port, 0 unknown
parameter: 0 error, 0 header, 0 option
         0 unknown
0 hopcount expired, 0 reassembly timeout,
0 unknown timeout, 0 too big,
0 echo request, 0 echo reply

Neighbor Discovery ICMP statistics:
Rcvd: 0 router solicit, 0 router advert, 0 redirect
      0 neighbor solicit, 0 neighbor advert
Sent: 0 router solicit, 0 router advert, 0 redirect
      0 neighbor solicit, 0 neighbor advert

UDP statistics:
0 packets input, 0 checksum errors
0 length errors, 0 no port, 0 dropped
0 packets output

TCP statistics:s
0 packets input, 0 checksum errors, 0 dropped
0 packets output, 0 retransmitted

```

This table describes the significant fields shown in the display.

**Table 12: show ipv6 traffic Command Field Descriptions**

Field	Description
Rcvd:	Statistics in this section refer to packets received by the router.
total	Total number of packets received by the software.
local destination	Locally destined packets received by the software.
source-routed	Packets seen by the software with RH.
truncated	Truncated packets seen by the software.
bad header	An error was found in generic HBH, RH, DH, or HA. Software only.
unknown option	Unknown option type in IPv6 header.
unknown protocol	Protocol specified in the IP header of the received packet is unreachable.
Sent:	Statistics in this section refer to packets sent by the router.
forwarded	Packets forwarded by the software. If the packet cannot be forwarded in the first lookup (for example, the packet needs option processing), then the packet is not included in this count, even if it ends up being forwarded by the software.
Mcast:	Multicast packets.

 show ipv6 traffic

Field	Description
ICMP statistics:	Internet Control Message Protocol statistics.



# show linux networking interfaces address-only

To display virtual IP addresses and IP addresses for address-only interfaces, use the **show linux networking interfaces address-only** command in the XR EXEC mode. Address-only interfaces are those interfaces whose addresses are copied to the Linux loopback device by XLNC (XR Linux networking coordinator).

## show linux networking interfaces address-only

<b>Syntax Description</b>	This command has no keywords or arguments.	
<b>Command Default</b>	None	
<b>Command Modes</b>	XR EXEC mode	
<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.5.2	The virtual IP addresses are displayed in the output of the command.
	Release 7.3.2	This command was introduced.
<b>Usage Guidelines</b>	None	
<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	system	read

## Example

This is the sample output of the **show linux networking interfaces address-only** command:

```
Router# show linux networking interfaces address-only
The following interface addresses have been added to the Linux loopback device for L3
reachability.

VRF default
-----
MgmtEth0/RP0/CPU0/0
  IPv4: 10.0.0.3 (virtual address)
  IPv6: 10::3 (virtual address)
```

# show local pool

To display IPv4 local pool details, use the **show local pool** command in XR EXEC mode.

**show** *{local|other\_pool\_types}* **pool** [**vrf** *vrf\_name*] *{ipv4 | ipv6}* *{default|poolname}*

<b>Syntax Description</b>	<b>local</b>	Specifies that the address pool is local.
	<b>vrf</b>	Specifies that a VRF name will be given. If is parameter is missing, the default VRF is assumed.
	<i>vrf_name</i>	Specifies the name of the VRF to which the addresses of the pool belongs. If no name is given, the default VRF is assumed.
	<b>default</b>	Creates a default local IPv4 address pool that is used if no other pool is named.
	<i>poolname</i>	Specifies the name of the local IPv4 address pool.

**Command Default** None

**Command Modes** XR EXEC mode

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

**Syntax Description** This command has no keywords or arguments.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv4	read
	network	read

## Examples

The following is sample output from the **show ipv4 local pool** with a poolname of P1:

```
RP/0/RP0/CPU0:router# show ipv4 local pool P1
```

```
Pool Begin End FreeInUse
P1 172.30.228.11172.30.228.1660
Available addresses:
172.30.228.11
172.30.228.12
172.30.228.13
172.30.228.14
172.30.228.15
172.30.228.16
Inuse addresses:
None
```

This table describes the significant fields shown in the display.

**Table 13: show ipv4 local pool Command Descriptions**

Field	Description
Pool	Name of the pool.
Begin	First IP address in the defined range of addresses in this pool.
End	Last IP address in the defined range of addresses in this pool.
Free	Number of addresses available.
InUse	Number of addresses in use.

# show mpa client

To display information about the Multicast Port Arbitrator (MPA) clients, use the **show mpa client** command in XR EXEC mode.

**show mpa client** {consumers | producers}

<b>Syntax Description</b>	<b>consumers</b>	Displays the clients for the consumers.
	<b>producers</b>	Displays the clients for the producers.

**Command Default** None

**Command Modes** XR EXEC mode

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.0.12	This command was introduced.

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	network	read

## Examples

The following sample output is from the **show mpa client** command:

```
RP/0/RP0/CPU0:router# show mpa client consumers
```

List of producer clients for ipv4 MPA

Location	Protocol	Process
0/1/CPU0	255	raw
0/1/CPU0	17	udp
0/4/CPU0	17	udp
0/4/CPU0	255	raw
0/4/CPU1	17	udp
0/4/CPU1	255	raw
0/6/CPU0	17	udp
0/6/CPU0	255	raw
0/RP1/CPU0	17	udp
0/RP1/CPU0	255	raw

# show mpa groups

To display Multicast Port Arbitrator (MPA) multicast group information, use the **show mpa groups** command in XR EXEC mode .

**show mpa groups** *type interface-path-id*

Syntax Description	<p><i>type</i> Interface type. For more information, use the question mark (?) online help function.</p> <p><i>interface-path-id</i> Either a physical interface instance or a virtual interface instance as follows:</p> <ul style="list-style-type: none"> <li>Physical interface instance. Naming notation is <i>rack/slot/module/port</i> and a slash between values is required as part of the notation. <ul style="list-style-type: none"> <li><i>rack</i>: Chassis number of the rack.</li> <li><i>slot</i>: Physical slot number of the modular services card or line card.</li> <li><i>module</i>: Module number. A physical layer interface module (PLIM) is always 0.</li> <li><i>port</i>: Physical port number of the interface.</li> </ul> </li> </ul> <p><b>Note</b> In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RSP0) and the module is CPU0. Example: interface MgmtEth0/RSP0 /CPU0/0.</p> <ul style="list-style-type: none"> <li>Virtual interface instance. Number range varies depending on interface type.</li> </ul> <p>For more information about the syntax for the router, use the question mark (?) online help function.</p>				
Command Default	None				
Command Modes	XR EXEC mode				
Command History	<table> <thead> <tr> <th>Release</th><th>Modification</th></tr> </thead> <tbody> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </tbody> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				
Usage Guidelines	No specific guidelines impact the use of this command.				
Task ID	<table> <thead> <tr> <th>Task ID</th><th>Operations</th></tr> </thead> <tbody> <tr> <td>network</td><td>read</td></tr> </tbody> </table>	Task ID	Operations	network	read
Task ID	Operations				
network	read				
Examples	<p>The following sample output is from the <b>show mpa groups</b> command:</p> <pre>RP/0/RP0/CPU0:router# show mpa groupsHundredGigE0/0/0/2 Mon Jul 27 04:07:19.802 DST HundredGigE0/0/0/2 :-</pre>				

```
224.0.0.1 : includes 0, excludes 1, mode EXCLUDE
<no source filter>
224.0.0.2 : includes 0, excludes 1, mode EXCLUDE
<no source filter>
224.0.0.5 : includes 0, excludes 1, mode EXCLUDE
<no source filter>
224.0.0.6 : includes 0, excludes 1, mode EXCLUDE
<no source filter>
224.0.0.13 : includes 0, excludes 1, mode EXCLUDE
<no source filter>
224.0.0.22 : includes 0, excludes 1, mode EXCLUDE
<no source filter>
```

# show mpa ipv4

To display information for Multicast Port Arbitrator (MPA) for IPv4, use the **show mpa ipv4** command in XR EXEC mode.

**show mpa ipv4** {**client** {**consumers** | **producers**} | **groups** *type interface-path-id* | **trace**}

Syntax Description	<b>client</b> Displays information about the MPA clients.				
	<b>consumers</b> Displays the clients for the consumers.				
	<b>producers</b> Displays the clients for the producers.				
	<b>groups</b> Displays information about the MPA multicast group.				
	<i>type</i> Interface type. For more information, use the question mark (?) online help function.				
	<i>interface-path-id</i> Either a physical interface instance or a virtual interface instance as follows: <ul style="list-style-type: none"> <li>Physical interface instance. Naming notation is <i>rack/slot/module/port</i> and a slash between values is required as part of the notation.               <ul style="list-style-type: none"> <li><i>rack</i>: Chassis number of the rack.</li> <li><i>slot</i>: Physical slot number of the modular services card or line card.</li> <li><i>module</i>: Module number. A physical layer interface module (PLIM) is always 0.</li> <li><i>port</i>: Physical port number of the interface.</li> </ul> </li> </ul> <p><b>Note</b> In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RSP0) and the module is CPU0. Example: interface MgmtEth0/RSP0/CPU0/0.</p> <ul style="list-style-type: none"> <li>Virtual interface instance. Number range varies depending on interface type.</li> </ul> <p>For more information about the syntax for the router, use the question mark (?) online help function.</p>				
	<b>trace</b> Displays MPA trace information				
Command Default	None				
Command Modes	XR EXEC mode				
Command History	<table> <thead> <tr> <th>Release</th><th>Modification</th></tr> </thead> <tbody> <tr> <td>Release 7.0.12</td><td>This command was introduced.</td></tr> </tbody> </table>	Release	Modification	Release 7.0.12	This command was introduced.
Release	Modification				
Release 7.0.12	This command was introduced.				
Usage Guidelines	No specific guidelines impact the use of this command.				

---

**Task ID**

---

**Task ID Operations**

---

---

network read

---

---

**Examples**

The following sample output is from the **show mpa ipv4** command:

```
RP/0/RP0/CPU0:router# show mpa ipv4 client producers
```

List of producer clients for ipv4 MPA

Location	Protocol	Process
0/1/CPU0	17	udp
0/1/CPU0	255	raw
0/4/CPU0	17	udp
0/4/CPU0	255	raw
0/4/CPU1	17	udp
0/4/CPU1	255	raw
0/6/CPU0	17	udp
0/6/CPU0	255	raw
0/RP0/CPU0	17	udp
0/RP0/CPU0	255	raw
0/RP1/CPU0	255	raw
0/RP1/CPU0	17	udp



# show mpa ipv6

To display information for Multicast Port Arbitrator (MPA) for IPv6, use the **show mpa ipv6** command in XR EXEC mode.

```
show mpa ipv6 {client {consumers | producers} | groups type interface-path-id}
```

Syntax Description	client	Displays information about the MPA clients.
	consumers	Displays the clients for the consumers.
	producers	Displays the clients for the producers.
	groups	Displays information about the MPA multicast group.
	type	Interface type. For more information, use the question mark (?) online help function.
	interface-path-id	<p>Either a physical interface instance or a virtual interface instance as follows:</p> <ul style="list-style-type: none"><li>Physical interface instance. Naming notation is <i>rack/slot/module/port</i> and a slash between values is required as part of the notation.<ul style="list-style-type: none"><li><i>rack</i>: Chassis number of the rack.</li><li><i>slot</i>: Physical slot number of the modular services card or line card.</li><li><i>module</i>: Module number. A physical layer interface module (PLIM) is always 0.</li><li><i>port</i>: Physical port number of the interface.</li></ul></li></ul> <p><b>Note</b> In references to a Management Ethernet interface located on a route processor card, the physical slot number is alphanumeric (RP0 or RP1) and the module is CPU0. Example: interface MgmtEth0/RP1/CPU0/0.</p> <ul style="list-style-type: none"><li>Virtual interface instance. Number range varies depending on interface type.</li></ul> <p>For more information about the syntax for the router, use the question mark (?) online help function.</p>
Command Default	None	
Command History	Release	Modification
	Release 7.0.12	This command was introduced.
Usage Guidelines	No specific guidelines impact the use of this command.	
Task ID	Task ID	Operations
	network	read

---

## Examples

The following sample output is from the **show mpa ipv6** command:

```
RP/0/RP0/CPU0:router# show mpa ipv6 client producers
```

List of producer clients for ipv6 MPA

Location	Protocol	Process
0/RP1/CPU0	17	udp
0/RP1/CPU0	255	raw

# show hw-module profile route-scale

To display the status of the configured IPv6 prefix scale expansion feature, run the **show hw-module profile route-scale** command in XR EXEC mode.

**show hw-module profile route-scale**

<b>Syntax Description</b>	This command has no keywords or arguments.
---------------------------	--

<b>Command Default</b>	None
------------------------	------

<b>Command Modes</b>	XR EXEC mode
----------------------	--------------

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	Release 7.5.3	This command was introduced.

<b>Usage Guidelines</b>	The chassis must be reloaded for the <b>hw-module</b> command to be functional.
-------------------------	---

<b>Task ID</b>	<b>Task ID</b>	<b>Operations</b>
	ipv6	read, write

After the configuration of the [hw-module profile route scale ipv6-unicast connected-prefix high](#) command is complete, reload the router for the feature to take effect. The **Applied** column in the **show hw-module profile route-scale** command displays *No* if the line card is not reloaded.

```
Router# show hw-module profile route-scale
Tue Aug 23 18:27:03.551 UTC
-----
Knob                               Status      Applied    Action
-----
Route-Scale                       Configured  No         Reload
```

After you reload the router for the feature to take effect, the **Applied** column displays *Yes*.

```
Router# reload location all
Tue Aug 23 18:27:56.482 UTC
Proceed with reload? [confirm] y

Router# show hw-module profile route-scale
Tue Aug 23 18:33:47.768 UTC
-----
Knob                               Status      Applied    Action
-----
Route-Scale                       Configured  Yes         None
```

```
show hw-module profile route-scale
```