

## N

## neighbor

**Description:** For OSPF, configures OSPF neighbors on an NBMA network. For RIP, specifies a RIP neighbor to which the router sends unicast messages. The **no** version removes the neighbor (OSPF and RIP) or restores the default values (OSPF).

**Syntax:** For OSPF:

```
neighbor ipAddress [ pollinterval seconds | priority number ]
```

```
no neighbor ipAddress [ pollinterval | priority ]
```

- *ipAddress* – IP address of the neighbor's interface; this interface must itself be configured for the NBMA network type
- *number* – router priority value of the neighbor in the range 1–4294967295; the default is 0
- *seconds* – interval in seconds at which the neighbor is polled; should be much larger than the hello interval (per RFC 1247); ranges from 0–255, the default is 120 seconds

For RIP:

```
[ no ] neighbor ipAddress
```

- *ipAddress* – IP address of the neighbor's interface; this interface must be defined as a passive interface with the **passive-interface** command

**Mode(s):** Router Configuration

## neighbor activate

**Description:** Specifies a peer or peer group with which routes of the current address family are exchanged. A peer or peer group can be activated in more than one address family. By default, a peer or peer group is activated only for the IPv4 unicast address family. The address families that are actively exchanged over a BGP session are negotiated when the session is established. The **no** version indicates that routes of the current address family should not be exchanged with the peer or peer group. The **default** version removes the explicit configuration from the peer or peer group and reestablishes inheritance of the feature configuration.

**Syntax:** [ no | default ] neighbor { *ipAddress* | *peerGroupName* } activate

- *ipAddress* – neighbor's IP address
- *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.

**Mode(s):** Address Family Configuration, Router Configuration

## neighbor advertisement-interval

---

- Description:** Sets the minimum interval between the sending of BGP updates for a given prefix. The **no** version restores the default interval.
- Syntax:** neighbor { *ipAddress* | *peerGroupName* } advertisement-interval *seconds*  
no neighbor { *ipAddress* | *peerGroupName* } advertisement-interval [ *seconds* ]
- *ipAddress* – neighbor's IP address
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - *seconds* – interval in seconds between update messages; range is from 0–600; default is 30 seconds for external peers and 5 seconds for internal peers
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor allowas-in

---

- Description:** Specifies the number of times that the AS path of a received route may contain the recipient BGP speaker's AS number and still be accepted. The **no** version restores the default state, which is to reject as a loop any route whose path contains the speaker's AS number.
- Syntax:** [ no ] neighbor { *ipAddress* | *peerGroupName* } allowas-in *number*
- *ipAddress* – neighbor's IP address
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - *number* – number in the range 1–10
- Mode(s):** Router Configuration

## neighbor as-override

---

- Description:** Prevents routing loops between routers within a VPN by substituting the current router's AS number in routing tables for that of the neighboring router. The **no** version halts this substitution. The **default** version removes the explicit configuration from the peer or peer group and reestablishes inheritance of the feature configuration.
- Syntax:** [ no | default ] neighbor { *ipAddress* | *peerGroupName* } as-override
- *ipAddress* – neighbor's IP address
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
- Mode(s):** Router Configuration

## neighbor default-originate

---

- Description:** Allows a BGP speaker (the local router) to send the default route 0.0.0.0/0 to a neighbor for use as a default route. The **no** version halts sending a default route to the neighbor. The **default** version removes the explicit configuration from the peer or peer group and reestablishes inheritance of the feature configuration.
- Syntax:** [ no | default ] neighbor { *ipAddress* | *peerGroupName* } default-originate
- *ipAddress* – neighbor's IP address
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor description

---

- Description:** Associates a textual description with a BGP neighbor. The **no** version removes the description.
- Syntax:** neighbor { *ipAddress* | *peerGroupName* } description *text*  
no neighbor { *ipAddress* | *peerGroupName* } description
- *ipAddress* – neighbor's IP address
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - *text* – up to 80 characters of text that describes the neighbor
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor distribute-list

---

- Description:** Distributes BGP neighbor information as specified in an access list. The **no** version removes an entry.
- Using distribute lists is one of several ways to filter BGP advertisements. You can also use route maps or use AS-path filters, as with the **ip as-path access-list** Global Configuration command and the **neighbor filter-list** command.
- Syntax:** neighbor { *ipAddress* | *peerGroupName* } distribute-list *accessListName* { in | out }  
no neighbor { *ipAddress* | *peerGroupName* } distribute-list [ *accessListName* ] { in | out }
- *ipAddress* – neighbor's IP address
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - *accessListName* – string of up to 32 alphanumeric characters identifying an access list
  - in – applies list to incoming routes (inbound policy)
  - out – applies list to outgoing routes (outbound policy); you cannot configure a member of a peer group to override the inherited peer group characteristic for outbound policy
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor ebgp-multihop

---

**Description:** Allows BGP to accept route updates from external peers residing on networks that are not directly connected. The **no** version halts accepting such routers. The **default** version removes the explicit configuration from the peer or peer group and reestablishes inheritance of the feature configuration.

**Syntax:** [ no | default ] neighbor { *ipAddress* | *peerGroupName* } ebgp-multihop [ *tth* ]

- *ipAddress* – neighbor's IP address
- *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
- *tth* – maximum number of hops to the peer; the value ranges from 1 to 255, with a default of 255

**Mode(s):** Address Family Configuration, Router Configuration

## neighbor filter-list

---

**Description:** Applies an AS path access list to advertisements inbound from or outbound to the specified neighbor, or assigns a weight to incoming routes that match the AS path access list. The **no** version stops the application of the list or assignment of the weight.

**Syntax:** To apply an access list:

```
neighbor { ipAddress | peerGroupName } filter-list accessListName { in | out }  
no neighbor { ipAddress | peerGroupName } filter-list [ accessListName ]  
{ in | out }
```

To assign a weight:

```
neighbor { ipAddress | peerGroupName } filter-list accessListName weight  
value
```

```
no neighbor { ipAddress | peerGroupName } filter-list [ accessListName ]  
weight [ value ]
```

- *ipAddress* – neighbor's IP address
- *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
- *accessListName* – name of a single AS path access list; string of up to 32 characters
- in – applies access list to incoming routes (inbound policy)
- out – applies access list to outgoing routes (outbound policy); you cannot configure a member of a peer group to override the inherited peer group characteristic for outbound policy
- *value* – number in the range 0–65535; assigns relative importance to incoming routes matching AS paths

**Mode(s):** Address Family Configuration, Router Configuration

## neighbor local-as

- Description:** Assigns a local AS number for the specified BGP peer or peer group. The **no** version restores the default value set globally for the BGP instance with the **router bgp** command.
- Syntax:**
- ```
neighbor { ipAddress | peerGroupName } local-as number
no neighbor { ipAddress | peerGroupName } local-as [ number ]
```
- *ipAddress* – neighbor IP address
  - *peerGroupName* – name of the BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - *number* – number in the range 1–65535; the local AS to assign to the peer
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor maximum-prefix

- Description:** Sets the number of prefixes that can be received from a neighbor. The **no** version removes the maximum prefix limitation.
- Syntax:**
- ```
neighbor { ipAddress | peerGroupName } maximum-prefix maximum
[ threshold ] [ strict ] [ warning-only ]
no neighbor { ipAddress | peerGroupName } maximum-prefix [ maximum ]
[ threshold ] [ strict ] [ warning-only ]
```
- *ipAddress* – neighbor IP address
  - *peerGroupName* – name of the BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - *maximum* – maximum number of prefixes; the default is no limit
  - *threshold* – percent of maximum at which to log a warning in the range 0–100; the default is 75
  - *strict* – checks the maximum prefix limit against all received routes rather than the default behavior of checking it only against accepted routes
  - *warning-only* – causes BGP software to log a warning, rather than reset the connection if the *maximum* or *threshold* value is exceeded
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor maximum-update-size

---

- Description:** Sets the maximum size of update messages transmitted to a BGP peer. The **no** version removes the maximum update size limitation.
- Syntax:** neighbor { *ipAddress* | *peerGroupName* } maximum-update-size *value*  
no neighbor { *ipAddress* | *peerGroupName* } maximum-update-size [ *value* ]
- *ipAddress* – neighbor IP address
  - *peerGroupName* – name of the BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - *value* – maximum update size in octets in the range 256–4096; the default is 1024
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor next-hop-self

---

- Description:** Forces the BGP speaker to report itself as the next hop for an advertised route it advertised to a neighbor. Typically you use this command to prevent third-party next hops from being used on NBMA media such as Frame Relay. The **no** version disables the feature. The **default** version removes the explicit configuration from the peer or peer group and reestablishes inheritance of the feature configuration.
- Syntax:** [ no | default ] neighbor { *ipAddress* | *peerGroupName* } next-hop-self
- *ipAddress* – IP address of BGP-speaking neighbor
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor password

- Description:** Enables MD5 authentication on a TCP connection between two BGP peers. The **no** version disables MD5 authentication.
- Syntax:** neighbor { *ipAddress* | *peerGroupName* } password [ 0 | 8 ] *string*  
no neighbor { *ipAddress* | *peerGroupName* } password
- *ipAddress* – IP address of the BGP neighbor that belongs to the peer group specified by the name
  - *peerGroupName* – name of the BGP peer group to which this neighbor belongs. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - 0 – indicates the MD5 password is entered in unencrypted form (plaintext)
  - 8 – indicates the MD5 password is entered in encrypted form (ciphertext)
  - *string* – MD5 password, an alphanumeric text string of up to 80 characters
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor peer-group

- Description:** When used from Router Configuration mode without specifying an IP address, creates a BGP peer group. The **no** version removes a peer group.
- When used from Router Configuration mode with an IP address or from Address Family Configuration mode, configures a BGP neighbor to be a member of a peer group. The **no** version removes a neighbor from a peer group.
- Syntax:** Creating a peer group:  
neighbor *peerGroupName* peer-group  
no neighbor *peerGroupName* [ peer-group ]
- *peerGroupName* – name of BGP peer group
- Assigning members to a peer group:  
neighbor *ipAddress* peer-group *peerGroupName*  
no neighbor *ipAddress* peer-group [ *peerGroupName* ]
- *ipAddress* – IP address of the BGP neighbor that belongs to the peer group specified by the name
  - *peerGroupName* – name of the BGP peer group to which this neighbor belongs
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor prefix-list

---

- Description:** Assigns an inbound or outbound prefix list. The **no** version removes the prefix list.
- Syntax:** neighbor { *ipAddress* | *peerGroupName* } prefix-list *prefix-list-name* { in | out }  
no neighbor { *ipAddress* | *peerGroupName* } prefix-list [ *prefixListName* ] { in | out }
- *ipAddress* – IP address of BGP-speaking neighbor
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - *prefixListName* – name of a BGP prefix list
  - in – assigns prefix list to incoming routes (inbound policy)
  - out – assigns prefix list to outgoing routes (outbound policy); you cannot configure a member of a peer group to override the inherited peer group characteristic for outbound policy
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor prefix-tree

---

- Description:** Assigns an inbound or outbound prefix tree. The **no** version removes the prefix tree.
- Syntax:** neighbor { *ipAddress* | *peerGroupName* } prefix-tree *prefixTreeName* { in | out }  
no neighbor { *ipAddress* | *peerGroupName* } prefix-tree [ *prefixTreeName* ] { in | out }
- *ipAddress* – IP address of BGP-speaking neighbor
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - *prefixTreeName* – name of a BGP prefix tree
  - in – assigns prefix tree to incoming routes (inbound policy)
  - out – assigns prefix tree to outgoing routes (outbound policy); you cannot configure a member of a peer group to override the inherited peer group characteristic for outbound policy
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor remote-as

---

- Description:** Adds an entry to the BGP neighbor table. The **no** version removes an entry from the table. Specifying a neighbor with an AS number that matches the AS number specified in the **router bgp** Global Configuration command identifies the neighbor as internal to the local AS. Otherwise, the neighbor is considered external.
- Syntax:** neighbor { *ipAddress* | *peerGroupName* } remote-as *number*  
no neighbor { *ipAddress* | *peerGroupName* } [ remote-as [ *number* ] ]
- *ipAddress* – IP address
  - *peerGroupName* – the name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - *number* – number in the range 1–65535; the AS to which the neighbor belongs
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor remove-private-as

---

- Description:** Removes private AS numbers in updates sent to external peers. The **no** version halts removing private AS numbers. The **default** version removes the explicit configuration from the peer or peer group and reestablishes inheritance of the feature configuration.
- Syntax:** [ no | default ] neighbor { *ipAddress* | *peerGroupName* } remove-private-as
- *ipAddress* – neighbor's IP address
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor rib-out disable

---

**Description:** Disables storage of routes to the Adj-RIBs-Out table for the neighbor or peer group. Storage is disabled by default. The **no** version enables the route storage. The **default** version removes the explicit configuration from the peer or peer group and reestablishes inheritance of the feature configuration.

**Syntax:** [ no | default ] neighbor { *ipAddress* | *peerGroupName* } rib-out disable

- *ipAddress* – neighbor's IP address
- *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.



**Note:** If you issue the **no neighbor rib-out disable** command to reen able storage of routes in the Adj-RIBs-Out table for a peer, the routes are not stored and you cannot display them with the **show ip bgp neighbor advertised-routes** command until you perform a hard clear of the peer with the **clear ip bgp** command.

**Mode(s):** Address Family Configuration, Router Configuration

## neighbor route-map

---

**Description:** Applies a route map to incoming or outgoing routes. If an outbound route map is specified, BGP advertises only routes that match at least one section of the route map. The **no** version removes a route map.

**Syntax:** neighbor { *ipAddress* | *peerGroupName* } route-map *mapTag* { in | out }  
no neighbor { *ipAddress* | *peerGroupName* } route-map [ *mapTag* ] { in | out }

- *ipAddress* – neighbor's IP address
- *peerGroupName* – name of BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
- *mapTag* – name of the route map; a string of up to 32 alphanumeric characters
- in – apply to incoming routes
- out – apply to outgoing routes; you cannot configure a member of a peer group to override the inherited peer group characteristic for outbound policy

**Mode(s):** Address Family Configuration, Router Configuration

## neighbor route-reflector-client

---

- Description:** Configures a system as a BGP route reflector and configures the specified neighbor as its client. The **no** version indicates that the neighbor is not a client. The **default** version removes the explicit configuration from the peer or peer group and reestablishes inheritance of the feature configuration.
- Syntax:** [ no | default ] neighbor { *ipAddress* | *peerGroupName* } route-reflector-client
- *ipAddress* – IP address of BGP neighbor being identified as a client
  - *peerGroupName* – name of BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor send-community

---

- Description:** Enables a BGP speaker to send a community attribute to the peer. The **no** version causes the speaker to send only standard communities to the peer. The **default** version removes the explicit configuration from the peer or peer group and reestablishes inheritance of the feature configuration.
- Syntax:** [ no | default ] neighbor { *ipAddress* | *peerGroupName* } send-community [ standard | extended | both ]
- *ipAddress* – neighbor's IP address
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
  - standard – sends only standard communities
  - extended – sends only extended communities
  - both – sends both standard and extended communities
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor shutdown

---

- Description:** Shuts down the specified neighbor or peer group without removing the neighbor or peer group configuration. The **no** version reenables a neighbor or peer group that was previously shut down. The **default** version removes the explicit configuration from the peer or peer group and reestablishes inheritance of the feature configuration.
- Syntax:** [ no | default ] neighbor { *ipAddress* | *peerGroupName* } shutdown
- *ipAddress* – neighbor's IP address
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor soft-reconfiguration inbound

---

- Description:** Initiates storage of unmodified copies of routes from the specified neighbor or all members of the specified peer group. The **no** version halts this storage. The **default** version removes the explicit configuration from the peer or peer group and reestablishes inheritance of the feature configuration.
- Syntax:** [ no | default ] neighbor { *ipAddress* | *peerGroupName* } soft-reconfiguration inbound
- *ipAddress* – neighbor's IP address
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor timers

- Description:** Sets keepalive and hold-time timers for the specified neighbor or peer group. Overrides values set for the router via the **timers bgp** command. The **no** version restores the default values.
- Syntax:**
- ```
neighbor { ipAddress | peerGroupName } timers keepaliveTime holdTime
no neighbor { ipAddress | peerGroupName } timers [ keepaliveTime ]
[ holdTime ]
```
- *ipAddress* – neighbor's IP address
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - *keepaliveTime* – interval in seconds between keepalive messages; range is from 0–65535 seconds, default is 30 seconds; a value of zero prevents BGP from sending keepalive messages
  - *holdTime* – period in seconds that BGP waits for keepalive messages before declaring the neighbor to be unavailable; range is from 0–65535 seconds; default is 90 seconds; a value of zero informs BGP not to expect any keepalive messages
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor unsuppress-map

- Description:** Restores the advertisement of routes suppressed by policy-based route flap dampening. The **no** version restores the default values.
- Syntax:**
- ```
neighbor { ipAddress | peerGroupName } unsuppress-map mapTag
no neighbor { ipAddress | peerGroupName } unsuppress-map [ mapTag ]
```
- *ipAddress* – neighbor's IP address
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command. You cannot override the characteristic for a specific member of the peer group.
  - *mapTag* – name of the route map; a string of up to 32 alphanumeric characters
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor update-source

---

- Description:** Allows a BGP session to use the IP address of any operational interface as the source address of TCP connections used by BGP. The **no** version restores the interface assignment to the closest interface.
- Syntax:**
- ```
neighbor { ipAddress | peerGroupName } update-source  
{ interfaceType interfaceSpecifier | updateSourceAddress }
```
- ```
no neighbor { ipAddress | peerGroupName } update-source  
[ interfaceType interfaceSpecifier | updateSourceAddress ]
```
- *ipAddress* – IP address of BGP-speaking neighbor
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - *interfaceType* – interface type; see *Interface Types and Specifiers* in *About This Guide*
  - *interfaceSpecifier* – particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *About This Guide*
  - *updateSourceAddress* – the source IP address
- Mode(s):** Address Family Configuration, Router Configuration

## neighbor weight

---

- Description:** Assigns a weight to a neighbor connection. The **no** version removes a weight assignment. All routes learned from this neighbor will have the assigned weight initially. The route with the highest weight will be chosen as the preferred route when multiple routes are available to a particular network.
- The weights assigned with the **match as-path** and **set weight route-map** commands override the weights assigned using the **neighbor weight** and **neighbor filter-list** commands.
- Syntax:**
- ```
neighbor { ipAddress | peerGroupName } weight value
```
- ```
no neighbor { ipAddress | peerGroupName } weight [ value ]
```
- *ipAddress* – neighbor's IP address
  - *peerGroupName* – name of a BGP peer group. If you specify a BGP peer group by using the *peerGroupName* argument, all the members of the peer group inherit the characteristic configured with this command, unless it is overridden for a specific peer.
  - *value* – number in the range 0–65535; the weight to assign
- Mode(s):** Address Family Configuration, Router Configuration

## net

**Description:** Configures an IS-IS network entity title for the specified routing process. The **no** version removes a specific NET. You must specify a NET. You can add multiple manual area IDs by adding multiple NETs with the same system ID. The last NET cannot be removed.

**Syntax:** [ no ] net *networkEntityTitle*

- *networkEntityTitle* – NET that specifies the area ID and the system ID for an IS-IS routing process; can be either an address or a name; in the form of: *areaID.systemID.nSelector*

For example:

```

47.0010.0000.0000.0001.0001,1111.1111.1111,00
      area ID          system ID  N selector
  
```

- › *areaID* – all bytes in front of the system ID; the number of bytes can vary from 1–13 bytes
- › *systemID* – always 6 bytes and cannot vary
- › *nSelector* – the last byte; always 0

**Mode(s):** Router Configuration

## netbios-name-server

**Description:** Assigns a Net-Bios server to subscribers of an address pool. The **no** version removes the association between the address pool and the Net-Bios server.

**Syntax:** netbios-name-server *ipAddressPrimary* [ *ipAddressSecondary* ]  
no netbios-name-server

- *ipAddressPrimary* – IP address of preferred Netbios server
- *ipAddressSecondary* – IP address of secondary DNS server

**Mode(s):** Pool Configuration

## netbios-node-type

---

**Description:** Specifies a Net-Bios node type. The **no** version restores the default situation, in which the node type is unspecified.

**Syntax:** netbios-node-type *nodeType*

no netbios-node-type

- *nodeType* – one of the following types of Net-Bios servers:
  - › b-node – Net-Bios Broadcast node
  - › p-node – Net-Bios Peer-to-Peer node
  - › m-node – Net-Bios mixed node
  - › h-node – Net-Bios hybrid node

**Mode(s):** Pool Configuration

## network

**Description:** For BGP, configures a BGP speaker with a prefix originating within its AS that it advertises to its peers if a non-BGP route to the prefix exists in the IP forwarding table. The **no** version removes the prefix.

For RIP, enables RIP on a specific network (not on a range of networks). If you do not associate a network with RIP, the system cannot advertise the network in any RIP update. The **no** version disables RIP on a specific network. If you do not specify a network mask, the system applies the natural mask. Use the **ip rip** commands to configure RIP attributes on the network.

For DHCP local server, specifies the IP addresses that the DHCP local server can provide from an address pool. The **no** version removes the network address and mask.

**Syntax:** For BGP:

```
[ no ] network networkNumber [ mask networkMask ] [ route-map mapTag ]
[ weight weight ] [ backdoor ]
```

- *networkNumber* – prefix that BGP will advertise
- *networkMask* – subnet mask
- *mapTag* – name of the route map; a string of up to 32 alphanumeric characters
- *weight* – number in the range 0–65535; default is 32768; assigns an absolute weight to the network route that overrides a weight assigned by the **redistribute** command
- *backdoor* – lowers the preference of an EBGp route to the specified prefix by setting the administrative distance to the value of an internal BGP route. Use this option to favor an IGP backdoor route over an EBGp route to a specific network. BGP does not advertise the prefix specified with this option.

For RIP:

```
[ no ] network networkAddress [ networkMask ]
```

- *networkAddress* – IP address of the network
- *networkMask* – subnet mask; the command accepts either the standard mask (network 10.2.1.0 255.255.255.0) or the inverse mask (network 10.2.1.0 0.0.0.255)

For DHCP local server:

```
network networkAddress { networkMask | prefix }
```

```
no network [ force ]
```

- *ipAddress* – IP address of the network

- *networkMask* – subnet mask for the network
- *prefix* – network prefix
- *force* – deletes address pool even if the pool is in use

**Mode(s):** Address Family Configuration or Router Configuration for BGP; Router Configuration for RIP; Pool Configuration for DHCP local server

## network area

---

**Description:** Defines the interfaces on which OSPF runs and the area ID for those interfaces. The **no** version deletes OSPF interfaces, ranges, and areas.

**Syntax:** [ no ] network *ipNet maskWildcard* area { *areald* | *arealdInt* }

- *ipNet* – network number
- *maskWildcard* – wild-card mask for the network number
- *areald* – OSPF area ID in IP address format
- *arealdInt* – OSPF area ID as a decimal value, in the range 0–4294967295



**Note:** Before you issue this command, you must first configure one or more interface with an IP address that is within the range specified by *ipNet*.

**Mode(s):** Router Configuration

## next-address

---

**Description:** Configures an IPv4 hop at the end of the MPLS explicit path. There is no **no** version.

**Syntax:** next-address *ipAddress* [ mask *ipMask* ] [ loose ]

- *ipAddress* – address of the node
- *ipMask* – [not currently used] mask for the next adjacent address
- *loose* – indicates the node is not necessarily directly connected (adjacent) to the previous node in the path. If *loose* is not configured, the configuration defaults to *strict*. *Strict* indicates that the node is directly connected to the previous node.

**Mode(s):** Explicit Path Configuration

## next-hop

---

**Description:** Defines the IP address of the next hop for a policy list. The **classifier-group** keyword and *clacIName* argument specify the classifier control list. If you do not specify a classifier group, the system selects all packets from the interface associated with this policy list for this rule. Use the **suspend** keyword to suspend a filter rule temporarily. The **no** version removes a next hop rule from a policy list.



**Note:** The SRP module does not support the **next-hop** command.

**Syntax:** [ no ] [ suspend ] next-hop *address* [ classifier-group *clacIName* ]  
[ precedence *precValue* ]

- **suspend** – suspend a policy rule
- *address* – IP address for the next hop
- *clacIName* – classifier control list used to classify packets for this next-hop policy
- *precValue* – precedence of this rule in relation to other rules within this set

**Mode(s):** Policy Configuration

## next-interface

---

**Description:** Defines an output interface for a policy list. Use the **classifier-group** option to specify the classifier control list. If you do not specify a classifier control list, the system selects all packets from the interface associated with this policy list for this rule. Specify a **next-hop** command when the *interfaceType* is a broadcast medium. The **no** version removes a next-interface rule from a policy list. The **suspend** version suspends a filter rule temporarily.



**Note:** The SRP module does not support the **next-interface** command.

**Syntax:** [ no ] [ suspend ] next-interface *interfaceType interfaceNumber*  
[ next-hop *nextHop* ] [ classifier-group *clacIName* ] [ precedence *precValue* ]

- **suspend** – suspend a policy rule
- *interfaceType* – type of interface to route packets to
- *interfaceNumber* – number of the interface to route packets to
- *nextHop* – next-hop IP address
- *clacIName* – classifier control list used to classify packets for this next-interface policy
- *precValue* – precedence of this rule in relation to other rules within this set

**Mode(s):** Policy Configuration

---

**no area**

---

- Description:** Removes the specified OSPF area if there are no OSPF interfaces configured in the area.
- Syntax:** no area { *areald* | *arealdInt* }
- *areald* – OSPF area ID in IP address format
  - *arealdInt* – OSPF area ID as a decimal value 0–4294967295
- Mode(s):** Router Configuration

---

**no ip interface**

---

- Description:** Removes the IP configuration from the interface or subinterface and disables IP processing on the interface.
- Syntax:** no ip interface
- Mode(s):** Interface Configuration, Subinterface Configuration

---

**no log filters**

---

- Description:** Turns off all log filters. To turn off a specific filter, use the **no** version of the **log severity** command that you used to add the filter.
- Syntax:** no log filters
- Mode(s):** Global Configuration

---

**no radius client**

---

- Description:** Use to unconfigure all RADIUS servers for the virtual router context and to delete the RADIUS client for the virtual router context.
- Syntax:** no radius client
- Mode(s):** Global Configuration

## node

---

<b>Description:</b>	Specifies that a scheduler node be configured for each interface of the given interface type. The <b>no</b> version removes this rule from the QoS profile.
<b>Syntax:</b>	[ no ] <i>interfaceType</i> node [ scheduler-profile <i>schedulerProfileName</i> ] [ group <i>groupName</i> ] <ul style="list-style-type: none"> <li>• <i>interfaceType</i> – one of the following interface types for which scheduler nodes should be configured: <b>atm-vc, cbf, fr-vc, ip, ip-tunnel, l2tp-tunnel, vlan</b></li> <li>• <i>schedulerProfileName</i> – name of the scheduler profile</li> <li>• <i>groupName</i> – name of the traffic class group</li> </ul>
<b>Mode(s):</b>	QoS Profile Configuration

## nrzi-encoding

---

<b>Description:</b>	Use to enable NRZI encoding.
<b>Syntax:</b>	[ no ] nrzi-encoding
<b>Mode(s):</b>	Interface Configuration

## ntp access-group

---

<b>Description:</b>	Specifies the type of broadcasts that the system will accept and respond to, and specifies the servers from which the system will accept broadcasts. The <b>no</b> version enables the system to receive all NTP broadcasts on interfaces configured to receive broadcasts.
<b>Syntax:</b>	ntp access-group { peer   serve-only   serve   query-only } <i>accessListName</i> no ntp access-group { peer   serve-only   serve   query-only }



**Note:** The system can accept, but does not use, NTP control queries.

- peer – enables the system to receive time requests, receive NTP control queries, and synchronize itself to the servers specified on the access-list
- serve-only – enables the system to receive time requests and NTP control queries from servers specified on the access-list, but not to synchronize itself to the specified servers
- serve – enables the system only to receive time requests from the servers specified on the access-list
- query-only – enables the system only to receive NTP control queries from the servers specified on the access-list
- *accessListName* – name of the access list

<b>Mode(s):</b>	Global Configuration
-----------------	----------------------

## ntp broadcast-client

---

- Description:** Enables an interface to receive NTP broadcasts. The **no** version prevents an interface from receiving NTP broadcasts.
- Syntax:** [ no ] ntp broadcast-client
- Mode(s):** Interface Configuration

## ntp broadcast-delay

---

- Description:** Sets the estimated round-trip delay between the broadcast NTP server and the system. The **no** version restores the round-trip delay to the default value, 3000 microseconds.
- Syntax:** ntp broadcast-delay *delayTime*  
no ntp broadcast-delay
- *delayTime* – a value in the range 0–999999 microseconds
- Mode(s):** Global Configuration

## ntp disable

---

- Description:** Disables NTP on an interface. The **no** version enables NTP on an interface. The default setting is enable.
- Syntax:** [ no ] ntp disable
- Mode(s):** Interface Configuration

## ntp enable

---

- Description:** Enables NTP services on the system and attaches the NTP client to the current virtual router. The **no** version disables reception of NTP packets on the system and removes the association between NTP and the virtual router. The default setting is disable.
- Syntax:** [ no ] ntp enable
- Mode(s):** Global Configuration

ntp master

---

**Description:** Specifies the stratum number of a virtual router you configured as an NTP server. By default, if the system is configured as an NTP server, the stratum number is set to the stratum number of the master plus one. The **no** version restores the default stratum number.

**Syntax:** [ no ] ntp master [ *stratumNumber* ]



**Note:** Although you can specify a stratum number of 1, the system does not support stratum 1 service. The system can synchronize only with an NTP server, and not directly with an atomic clock or radio clock.

- *stratumNumber* – number in the range 1–15 that indicates how many hops the NTP server is from an accurate time source, such as a radio clock or atomic clock. Stratum *n* servers are *n* hops from an accurate time source. The default is 8.

**Mode(s):** Global Configuration

ntp server

---

**Description:** Specifies an NTP server for time synchronization. The **source** option for this command overrides the **ntp source** command. The **no** version terminates NTP communications between this server and the interface.

**Syntax:** ntp server *ipAddress* [ version *number* ] [ prefer ] [ source *interfaceType interfaceSpecifier* ]

no ntp server *ipAddress*

- *ipAddress* – IP address of the NTP server
- *number* – value from 1 to 4; indicates the version of the NTP software on the server
- prefer – indicates that this server is the first choice for time synchronization
- source – directs responses from the NTP server to a specific interface on the system; overrides the **ntp source** command
- *interfaceType* – interface type; see *Interface Types and Specifiers* in *About This Guide*
- *interfaceSpecifier* – particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *About This Guide*

**Mode(s):** Global Configuration

## ntp server enable

---

- Description:** Enables a virtual router to act as an NTP server. The **no** version prevents a virtual router from acting as an NTP server.
- Syntax:** [ no ] ntp server enable
- Mode(s):** Global Configuration

## ntp source

---

- Description:** Directs responses from all NTP servers to a specific interface. Using the **source** option with the **ntp server** command overrides the **ntp source** command. The **no** version restores the default situation in which servers reply to the interface from which the NTP request was sent.
- Syntax:** ntp source *interfaceType interfaceSpecifier*  
no ntp source
- *interfaceType* – interface type; see *Interface Types and Specifiers* in *About This Guide*
  - *interfaceSpecifier* – particular interface; format varies according to interface type; see *Interface Types and Specifiers* in *About This Guide*
- Mode(s):** Global Configuration