ZebOS-XP®
Network Platform

Version 1.4
Extended Performance

Border Gateway Protocol
Command Reference

December 2015

IP Infusion Inc.
3965 Freedom Circle, Suite 200
Santa Clara, CA 95054
+1 408-400-1900
http://www.ipinfusion.com/

For support, questions, or comments via E-mail, contact:
support@ipinfusion.com

Contents

Preface

This document describes the ZebOS-XP commands for Border Gateway Protocol (BGP).

Audience

This document is intended for network administrators and other engineering professionals who configure and manage BGP.

Conventions

Table P-1 shows the conventions used in this guide.

Table P-1: Conventions

Convention	**Description**
Italics	Emphasized terms; titles of books
Note:	Special instructions, suggestions, or warnings
`monospaced type`	Code elements such as commands, functions, parameters, files, and directories

Contents

This document contains these chapters and appendices:

- Chapter 1, *Command Line Interface*
- Chapter 2, *BGP Commands*
- Chapter 3, *BGP4+ Commands*
- Chapter 4, *BGP Virtual Private Network Commands*
- Chapter 5, *BGP Show Commands*
- Appendix A, *Regular Expressions*

Related Documents

The following guides are related to this document:

- *Border Gateway Protocol Developer Guide*
- *Unicast Configuration Guide*
- *Installation Guide*

Note: All ZebOS-XP technical manuals are available to licensed customers at http://www.ipinfusion.com/support/document_list.

Support

For support-related questions, contact support@ipinfusion.com.

Comments

If you have comments, or need to report a problem with the content, contact techpubs@ipinfusion.com.

CHAPTER 1 Command Line Interface

This chapter introduces the ZebOS-XP Command Line Interface (CLI) and how to use its features.

Overview

You use the CLI to configure, monitor, and maintain ZebOS-XP devices. The CLI is text-based and each command is usually associated with a specific task.

You can give the commands described in this manual locally from the console of a device running ZebOS-XP or remotely from a terminal emulator such as `putty` or `xterm`. You can also use the commands in scripts to automate configuration tasks.

Starting the Command Line Interface

You must start daemons as described in this section before you can use the CLI. The general steps are listed below. For details about the ZebOS-XP daemons, see the *Installation Guide.*

1. Start your terminal emulator and connect to the device or go to the console of the device running ZebOS-XP.
2. Connect to the directory where you installed the ZebOS-XP executables.
3. Start the Network Services Module (NSM).

```
# ./nsm -d
```

4. Start the protocol module daemons that your organization uses, such as `mstpd`, `ospf6d`, or `ripd`.

```
# ./mstpd –d
```

5. Start the Integrated Management Interface (IMI) daemon.

```
# ./imi -d
```

6. Start the IMI shell.

```
# ./imish
```

Note: Your organization may use a ZebOS-XP build that does not include `imish`. If that is the case, you must connect to a port on which a protocol daemon is listening. For details, see the *Installation Guide.*

You can now begin using the CLI.

Command Line Interface Help

You access the CLI help by entering a full or partial command string and a question mark “?”. The CLI displays the command keywords or parameters along with a short description. For example, at the CLI command prompt, type:

```
> show ?
```

The CLI displays this keyword list with short descriptions for each keyword:

```
show ?
  application-priority            Application Priority
```

```
      arp                             Internet Protocol (IP)
      bfd                             Bidirectional Forwarding Detection (BFD)
      bgp                             Border Gateway Protocol (BGP)
      bi-lsp                          Bi-directional lsp status and configuration
      bridge                          Bridge group commands
      ce-vlan                         COS Preservation for Customer Edge VLAN
      class-map                       Class map entry
      cli                             Show CLI tree of current mode
      clns                            Connectionless-Mode Network Service (CLNS)
      control-adjacency               Control Adjacency status and configuration
      control-channel                 Control Channel status and configuration
      cspf                            CSPF Information
      customer                        Display Customer spanning-tree
      cvlan                           Display CVLAN information
      debugging                       Debugging functions (see also 'undebug')
      dot1x                           IEEE 802.1X Port-Based Access Control
      etherchannel                    LACP etherchannel
      ethernet                        Layer-2
    ...
```

If you type the ? in the middle of a keyword, the CLI displays help for that keyword only.

```
    > show de?
    debugging  Debugging functions (see also 'undebug')
```

If you type the ? in the middle of a keyword, but the incomplete keyword matches several other keywords, ZebOS-XP displays help for all matching keywords.

```
    > show i? (CLI does not display the question mark).
    interface  Interface status and configuration
    ip         IP information
    isis       ISIS information
```

Command Completion

The CLI can complete the spelling of a command or a parameter. Begin typing the command or parameter and then press the tab key. For example, at the CLI command prompt type `sh`:

```
    > sh
```

Press the tab key. The CLI displays:

```
    > show
```

If the spelling of a command or parameter is ambiguous, the CLI displays the choices that match the abbreviation. Type `show i` and press the tab key. The CLI displays:

```
    > show i
      interface  ip         ipv6       isis
    > show i
```

The CLI displays the `interface` and `ip` keywords. Type `n` to select `interface` and press the tab key. The CLI displays:

```
    > show in
    > show interface
```

Type ? and the CLI displays the list of parameters for the `show interface` command.

```
    > show interface
      IFNAME  Interface name
      |       Output modifiers
```

```
>       Output redirection
<cr>
```

The CLI displays the only parameter associated with this command, the `IFNAME` parameter.

Command Abbreviations

The CLI accepts abbreviations that uniquely identify a keyword in commands. For example:

```
> sh in eth0
```

is an abbreviation for:

```
> show interface eth0
```

Command Line Errors

Any unknown spelling causes the CLI to display the error `Unrecognized command` in response to the ?. The CLI displays the command again as last entered.

```
> show dd?
% Unrecognized command
> show dd
```

When you press the Enter key after typing an invalid command, the CLI displays:

```
(config)#router ospf here
                          ^
% Invalid input detected at '^' marker.
```

where the ^ points to the first character in error in the command.

If a command is incomplete, the CLI displays the following message:

```
> show
% Incomplete command.
```

Some commands are too long for the display line and can wrap mid-parameter or mid-keyword, as shown below. This does *not* cause an error and the command performs as expected:

```
area 10.10.0.18 virtual-link 10.10.0.19 authent
ication-key 57393
```

Command Negation

Many commands have a `no` form that resets a feature to its default value or disables the feature. For example:

- The `ip address` command assigns an IPv4 address to an interface
- The `no ip address` command removes an IPv4 address from an interface

Syntax Conventions

Table 1-1 describes the conventions used to represent command syntax in this reference.

Table 1-1: Syntax conventions

Convention	Description	Example
monospaced font	Command strings entered on a command line	show bgp
lowercase	Keywords that you enter exactly as shown in the command syntax.	show bgp
UPPERCASE	See Variable Placeholders	IFNAME
()	Optional parameters, from which you must select one. Vertical bars delimit the selections. Do not enter the parentheses or vertical bars as part of the command.	(A.B.C.D|<0-4294967295>)
()	Optional parameters, from which you select one or none. Vertical bars delimit the selections. Do not enter the parentheses or vertical bars as part of the command.	(A.B.C.D|<0-4294967295>|)
()	Optional parameter which you can specify or omit. Do not enter the parentheses or vertical bar as part of the command.	(IFNAME|)
{}	Optional parameters, from which you must select one or more. Vertical bars delimit the selections. Do not enter the braces or vertical bars as part of the command.	{intra-area <1-255>|inter-area <1-255>|external <1-255>}
[]	Optional parameters, from which you select zero or more. Vertical bars delimit the selections. Do not enter the brackets or vertical bars as part of the command. A '?' before a parameter in square brackets limits that parameter to one occurrence in a command string.	[<1-65535>|AA:NN|internet|local-AS|no-advertise|no-export]
.	Repeatable parameter. The parameter that follows a period can be repeated more than once. Do not enter the period as part of the command.	set as-path prepend .<1-65535>

Variable Placeholders

Table 1-2 shows the tokens used in command syntax use to represent variables for which you supply a value.

Table 1-2: Variable placeholders

Token	Description
WORD	A contiguous text string (excluding spaces)
LINE	A text string, including spaces; no other parameters can follow this parameter
IFNAME	Interface name whose format varies depending on the platform; examples are: eth0, Ethernet0, ethernet0, xe0
A.B.C.D	IPv4 address
A.B.C.D/M	IPv4 address and mask/prefix
X:X::X:X	IPv6 address
X:X::X:X/M	IPv6 address and mask/prefix
HH:MM:SS	Time format
AA:NN	BGP community value
XX:XX:XX:XX:XX:XX	MAC address
<1-5> <1-65535> <0-2147483647> <0-4294967295>	Numeric range

Command Description Format

Table 1-3 explains the sections used to describe each command in this reference.

Table 1-3: Command descriptions

Section	Description
Command Name	The name of the command, followed by what the command does and when should it be used
Command Syntax	The syntax of the command
Parameters	Parameters and options for the command
Default	The state before the command is executed
Command Mode	The mode in which the command runs; see Command Modes
Example	An example of the command being executed

Keyboard Operations

Table 1-4 lists the operations you can perform from the keyboard.

Table 1-4: Keyboard operations

Key combination	Operation
Left arrow or Ctrl+b	Moves one character to the left. When a command extends beyond a single line, you can press left arrow or Ctrl+b repeatedly to scroll toward the beginning of the line, or you can press Ctrl+a to go directly to the beginning of the line.
Right arrow or Ctrl-f	Moves one character to the right. When a command extends beyond a single line, you can press right arrow or Ctrl+f repeatedly to scroll toward the end of the line, or you can press Ctrl+e to go directly to the end of the line.
Esc, b	Moves back one word
Esc, f	Moves forward one word
Ctrl+e	Moves to end of the line
Ctrl+a	Moves to the beginning of the line
Ctrl+u	Deletes the line
Ctrl+w	Deletes from the cursor to the previous whitespace
Alt+d	Deletes the current word
Ctrl+k	Deletes from the cursor to the end of line
Ctrl+y	Pastes text previously deleted with Ctrl+k, Alt+d, Ctrl+w, or Ctrl+u at the cursor

Table 1-4: Keyboard operations (Continued)

Key combination	Operation
Ctrl+t	Transposes the current character with the previous character
Ctrl+c	Ignores the current line and redisplays the command prompt
Ctrl+z	Ends configuration mode and returns to exec mode
Ctrl+l	Clears the screen
Up Arrow or Ctrl+p	Scroll backward through command history
Down Arrow or Ctrl+n	Scroll forward through command history

Show Command Modifiers

You can use two tokens to modify the output of a `show` command. Enter a question mark to display these tokens:

```
# show users ?
  | Output modifiers
  > Output redirection
```

You can type the | (vertical bar character) to use output modifiers. For example:

```
> show rsvp | ?
  begin     Begin with the line that matches
  exclude   Exclude lines that match
  include   Include lines that match
  redirect  Redirect output
```

Begin Modifier

The `begin` modifier displays the output beginning with the first line that contains the input string (everything typed after the `begin` keyword). For example:

```
# show run | begin eth1
...skipping
interface eth1
 ipv6 address fe80::204:75ff:fee6:5393/64
!
interface eth2
 ipv6 address fe80::20d:56ff:fe96:725a/64
!
line con 0
 login
!
end
```

You can specify a regular expression after the `begin` keyword, This example begins the output at a line with either “eth3” or “eth4”:

```
# show run | begin eth[3-4]

...skipping
interface eth3
```

```
 shutdown
!
interface eth4
 shutdown
!
interface svlan0.1
 no shutdown
!
route-map myroute permit 3
!
route-map mymap1 permit 10
!
route-map rmap1 permit 3
!
line con 0
 login
line vty 0 4
 login
!
end
```

Include Modifier

The `include` modifier includes only those lines of output that contain the input string. In the output below, all lines containing the word “input” are included:

```
# show interface eth1 | include input
    input packets 80434552, bytes 2147483647, dropped 0, multicast packets 0
    input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 1, missed 0
```

You can specify a regular expression after the `include` keyword. This examples includes all lines with “input” or “output”:

```
#show int eth0 | include (in|out)put
    input packets 597058, bytes 338081476, dropped 0, multicast packets 0
    input errors 0, length 0, overrun 0, CRC 0, frame 0, fifo 0, missed 0
    output packets 613147, bytes 126055987, dropped 0
    output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
```

Exclude Modifier

The `exclude` modifier excludes all lines of output that contain the input string. In the following output example, all lines containing the word “input” are excluded:

```
# show interface eth1 | exclude input
Interface eth1
  Scope: both
  Hardware is Ethernet, address is 0004.75e6.5393
  index 3 metric 1 mtu 1500 <UP,BROADCAST,RUNNING,MULTICAST>
  VRF Binding: Not bound
  Administrative Group(s): None
  DSTE Bandwidth Constraint Mode is MAM
  inet6 fe80::204:75ff:fee6:5393/64
    output packets 4438, bytes 394940, dropped 0
    output errors 0, aborted 0, carrier 0, fifo 0, heartbeat 0, window 0
    collisions 0
```

You can specify a regular expression after the `exclude` keyword. This example excludes lines with “output” or “input”:

```
# show interface eth0 | exclude (in|out)put
Interface eth0
  Scope: both
  Hardware is Ethernet  Current HW addr: 001b.2139.6c4a
  Physical:001b.2139.6c4a  Logical:(not set)
  index 2 metric 1 mtu 1500 duplex-full arp ageing timeout 3000
  <UP,BROADCAST,RUNNING,MULTICAST>
  VRF Binding: Not bound
  Bandwidth 100m
  DHCP client is disabled.
  inet 10.1.2.173/24 broadcast 10.1.2.255
  VRRP Master of :  VRRP is not configured on this interface.
  inet6 fe80::21b:21ff:fe39:6c4a/64
    collisions 0
```

Redirect Modifier

The `redirect` modifier writes the output into a file. The output is not displayed.

```
# show history | redirect /var/frame.txt
```

The output redirection token (>) does the same thing:

```
# show history >/var/frame.txt
```

Command Modes

Commands are grouped into modes arranged in a hierarchy. Each mode has its own set of commands. Table 1-5 lists the command modes common to all protocols.

Table 1-5: Common command modes

Name	Description
Executive mode	Also called *view* mode, this is the first mode to appear after you start the CLI. It is a base mode from where you can perform basic commands such as `show`, `exit`, `quit`, `help`, `list`, and `enable`.
Privileged executive mode	Also called *enable* mode, in this mode you can run additional basic commands such as `debug`, `write`, and `show`.
Configure mode	Also called *configure terminal* mode, in this mode you can run configuration commands and go into other modes such as interface, router, route map, key chain, and address family.
Interface mode	In this mode you can configure protocol-specific settings for a particular interface. Any setting you configure in this mode overrides a setting configured in router mode.
Router mode	This mode is used to configure router-specific settings for a protocol such as RIP or OSPF.

Command Mode Tree

The diagram below shows the common command mode hierarchy.

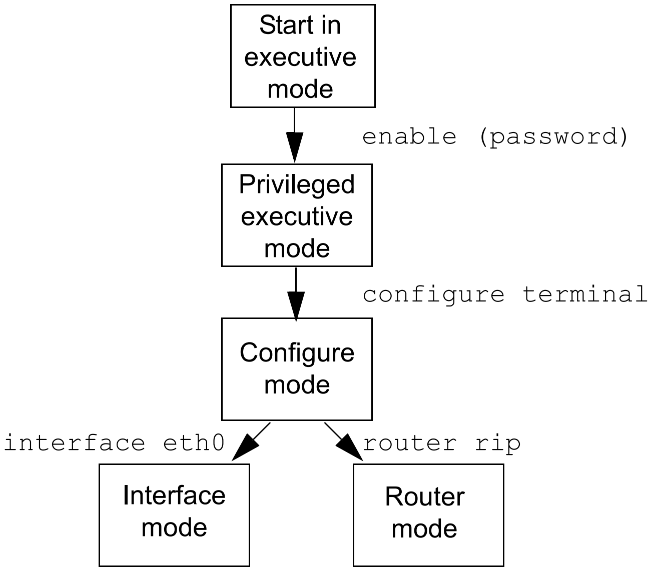

Figure 1-1: Common command modes

To change modes:

1. Enter privileged executive mode by entering `enable` in Executive mode.
2. Enter configure mode by entering `configure terminal` in Privileged Executive mode.

The example below shows starting `imish` and then moving from executive mode to privileged executive mode to configure mode and finally to router mode:

```
# ./imish
> enable mypassword
# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
(config)# router rip
(config-router)#
```

Note: Each protocol can have modes in addition to the common command modes. See the command reference for the respective protocol for details.

Debug Command

Whether the settings you make for a `debug` command persist between sessions depends on the mode where you make the settings:

- When you make settings for a `debug` command in executive mode, the configuration is valid for the current session only and is not saved in the `ZebOS.conf` file.
- When you make settings for a `debug` command in configuration mode, the configuration is retained and saved in `ZebOS.conf` and used even after the session restarts.

CHAPTER 2 BGP Commands

This chapter describes the BGP configuration commands.

- address-family
- aggregate-address
- auto-summary
- bgp aggregate-nexthop-check
- bgp always-compare-med
- bgp as-local-count
- bgp bestpath as-path ignore
- bgp bestpath compare-confed-aspath
- bgp bestpath compare-routerid
- bgp bestpath dont-compare-originator-id
- bgp bestpath med
- bgp bestpath tie-break-on-age
- bgp client-to-client reflection
- bgp cluster-id
- bgp confederation identifier
- bgp confederation peers
- bgp config-type
- bgp dampening
- bgp default ipv4-unicast
- bgp default local-preference
- bgp deterministic-med
- bgp enforce-first-as
- bgp extended-asn-cap
- bgp fast-external-failover
- bgp graceful-restart
- bgp g-shut
- bgp g-shut-capable
- bgp g-shut-local-preference
- bgp log-neighbor-changes
- bgp multiple-instance
- bgp nexthop-trigger delay
- bgp nexthop-trigger enable
- bgp rfc1771-path-select

- bgp rfc1771-strict
- bgp router-id
- bgp scan-time
- bgp update-delay
- clear bgp (A.B.C.D|X:X::X:X)
- clear bgp *
- clear bgp <1-4294967295>
- clear bgp <1-65535>
- clear bgp dampening
- clear bgp external
- clear bgp flap-statistics
- clear bgp peer-group
- clear bgp view
- clear ip bgp A.B.C.D
- clear ip bgp A.B.C.D vrf
- clear ip bgp table-map
- debug bgp
- distance bgp
- dump bgp all
- dump bgp routes-mrt
- dump bgp updates
- exit-address-family
- ip as-path access-list
- ip community-list <1-99>
- ip community-list <100-199>
- ip community-list expanded
- ip community-list standard
- ip community-list WORD
- ip extcommunity-list <1-99>
- ip extcommunity-list <100-199>
- ip extcommunity-list expanded
- ip extcommunity-list standard
- match ip peer
- max-paths
- mpls-resolution
- neighbor activate
- neighbor advertisement-interval
- neighbor allowas-in
- neighbor as-origination-interval

- neighbor attribute-unchanged
- neighbor capability dynamic
- neighbor capability graceful-restart
- neighbor capability orf prefix-list
- neighbor capability route-refresh
- neighbor collide-established
- neighbor connection-retry-time
- neighbor default-originate
- neighbor description
- neighbor disallow-infinite-holdtime
- neighbor distribute-list
- neighbor dont-capability-negotiate
- neighbor ebgp-multihop
- neighbor enforce-multihop
- neighbor fall-over bfd
- neighbor filter-list
- neighbor g-shut
- neighbor g-shut-timer
- neighbor local-as
- neighbor maximum-prefix
- neighbor next-hop-self
- neighbor override-capability
- neighbor passive
- neighbor password
- neighbor peer-group
- neighbor port
- neighbor prefix-list
- neighbor remote-as
- neighbor remove-private-AS
- neighbor restart-time
- neighbor route-map
- neighbor route-reflector-client
- neighbor route-server-client
- neighbor send-community
- neighbor send-label explicit-null
- neighbor shutdown
- neighbor soft-reconfiguration inbound
- neighbor strict-capability-match
- neighbor timers

- neighbor transparent-as
- neighbor transparent-nexthop
- neighbor unsuppress-map
- neighbor update-source
- neighbor version
- neighbor weight
- neighbor WORD peer-group
- network
- network synchronization
- redistribute
- restart bgp gracefull
- router bgp
- router bgp view
- snmp restart bgp
- synchronization
- table-map
- timers bgp
- undebug bgp

address-family

Use the address family command to enter the IPv4, IPv6 or VPNv4/v6 address family mode allowing configuration of address-family specific parameters. To leave the address family mode and return to the Configure mode use the `exit-address-family` command.

This command is used to configure routing exchange between Provider Edge (PE) and Customer Edge (CE) devices. The BGP sessions between PE routers can carry different types of routes (VPN-IPv4, IPv4, VPN-IPv6 and IPv6 routes). Address families are used to control the type of BGP session. Configure a BGP address family for each VRF configured on the PE router and a separate address family to carry VPN-IPv4 routes between PE routers. All non VPN BGP neighbors are defined using the `Router` mode. All VPN BGP neighbors are defined under its associated `Address Family` mode. The BGP process with no address-family specified, is the default address-family where any sessions are configured that either are not associated with a VRF or are used to carry IPv4 or IPv6 routes.

Use the `no` parameter with this command to disable the address-family configurations.

Command Syntax

```
address-family ipv4

address-family ipv4 (unicast|multicast)

address-family ipv4 vrf NAME

address-family ipv6 labeled-unicast

address-family ipv6 (unicast|)

address-family ipv6 vrf NAME

address-family vpnv4

address-family vpn4 unicast

address-family vpnv6

address-family vpnv6 unicast

no address-family ipv4 vrf NAME

no address-family ipv6 vrf NAME
```

Parameters

`ipv4`	IPv4 address family
`unicast`	Unicast address prefixes
`multicast`	Multicast address prefixes
`vrf`	VPN routing/forwarding instance
`NAME`	VPN routing/forwarding instance name
`ipv6`	IPv6 address family
`labeled-unicast`	Enter IPv6 Provider Edge (6PE) Address Family mode to exchange labeled routes data among ISP PE-devices in Address-Family IPv6 Labeled-Unicast mode. When a `neighbor activate` command is given in this mode, the device becomes 6PE capable.
`unicast`	Unicast address prefixes
`vrf`	VPN routing/forwarding instance
`NAME`	VPN routing/forwarding instance name

vpnv4	Virtual Private Network (VPN) version 4 address family
unicast	Unicast address prefixes
vpnv6	VPN version 6 address family
unicast	Unicast address prefixes

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 7657
(config-router)#neighbor 3ffe:506::1 remote-as 7657
(config-router)#neighbor 3ffe:506::1 interface eth1

#configure terminal
(config)#router bgp 7657
(config-router)#address-family ipv6
(config-router-af)#neighbor 3ffe:506::1 activate
(config-router-af)#exit-address-family
```

aggregate-address

Use this command to configure BGP aggregate entries.

Aggregates are used to minimize the size of routing tables. Aggregation combines the characteristics of several different routes and advertises a single route.The `aggregate-address` command creates an aggregate entry in the BGP routing table if any more-specific BGP routes are available in the specified range. Using the `summary-only` parameter advertises the prefix only, suppressing the more-specific routes to all neighbors. In the following example Router1 will propagate network 172.0.0.0 and suppresses the more specific route 172.10.0.0.

The `as-set` parameter creates an aggregate entry advertising the path for this route, consisting of all elements contained in all paths being summarized. Use this parameter to reduce the size of path information by listing the AS number only once, even if it was included in multiple paths that were aggregated. The `as-set` parameter is useful when aggregation of information results in an incomplete path information.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
aggregate-address A.B.C.D/M
aggregate-address A.B.C.D/M as-set
aggregate-address A.B.C.D/M as-set summary-only
aggregate-address A.B.C.D/M summary only
aggregate-address A.B.C.D/M summary-only as-set
no aggregate-address A.B.C.D/M
no aggregate-address A.B.C.D/M as-set
no aggregate-address A.B.C.D/M as-set summary-only
no aggregate-address A.B.C.D/M summary only
no aggregate-address A.B.C.D/M summary-only as-set
```

Parameters

`A.B.C.D/M`	Aggregate prefix
`as-set`	Generate AS set path information
`summary-only`	Filter more specific routes from updates

Default

Disabled

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#aggregate-address 10.0.0.0/8 as-set summary-only

(config)#router bgp 100
(config-router)#no aggregate-address 10.0.0.0/8 as-set summary-only
```

auto-summary

Use this command to enable sending summarized routes by a BGP speaker to its peers in the router configuration mode or in the address-family configuration mode. Auto-summary is used by a BGP router to advertise summarized routes to its peers. Auto-summary can be enabled if certain routes have already been advertised: in this case, configuring auto-summary advertises the summarized routes first, then corresponding non-summarized routes are withdrawn. If certain routes have already been advertised, and auto-summary is disabled, non-summarized routes are first advertised, then the corresponding summarized routes are withdrawn from all the connected peers.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
    auto-summary

    no auto-summary
```

Parameters

None

Default

Disabled

Command Mode

Router mode and Address Family mode

Examples

The following example enables auto-summary in Router mode.

```
    #configure terminal
    (config)#router bgp 11
    (config-router)#auto-summary
```

The following example enables auto-summary using the IPv4 address family.

```
    #configure terminal
    (config)#router bgp 1
    (config)#address-family ipv4
    (config-af)#auto-summary
```

bgp aggregate-nexthop-check

Use this command to set the BGP option to perform aggregation only when next-hop matches the specified IP address.

Use the `no` parameter with this command to disable this functionality.

Command Syntax

```
bgp aggregate-nexthop-check
no bgp aggregate-nexthop-check
```

Parameters

None

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#bgp aggregate-nexthop-check
```

bgp always-compare-med

Use this command to compare the Multi Exit Discriminator (MED) for paths from neighbors in different autonomous systems. Multi Exit Discriminator (MED) is used in best path selection by BGP. MED is compared after BGP attributes weight, local preference, AS-path and origin have been compared and are equal. MED comparison is done only among paths from the same autonomous system (AS). Use `bgp always-compare-med` command to allow comparison of MEDs from different ASs. The MED parameter is used to select the best path. A path with lower MED is preferred. If the bgp table shows the following and the always-compare-med is enabled:

```
Route1: as-path 400, med 300
Route2: as-path 200, med 200
Route3: as-path 400, med 250
```

Route1 is compared to Route2. Route2 is best of the two (lower MED). Next, Route2 is compared to Route3 and Route2 is chosen best path again (lower MED). If always-compare-med was disabled, MED is not taken into account when Route1 and Route2 are compared, because of different ASs and MED is compared for only Route1 and Route3. In this case, Route3 would be the best path. The selected route is also affected by the `bgp deterministic-med` command. Please see `bgp deterministic-med` command for details. If this command is used to compare MEDs for all paths, it should be configured on every BGP router in the AS.

Use the `no` parameter with this command to disallow the comparison.

Command Syntax

```
    bgp always-compare-med

    no bgp always-compare-med
```

Parameters

None

Default

Disabled

Command Mode

Router mode

Examples

```
    #configure terminal
    (config)#router bgp 100
    (config-router)#bgp always-compare-med
```

bgp as-local-count

Use this command to set the number of times the local-AS (Autonomous System) is to be prepended.

Use the no parameter with this command to stop prepending the local AS count.

Command Syntax

```
bgp as-local-count <2-64>

no bgp as-local-count <2-64>
```

Parameter

<2-64>	The number of times the local-AS is to be prepended

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp as-local-count 55

(config)#router bgp 100
(config-router)#no bgp as-local-count 55
```

bgp bestpath as-path ignore

Use this command to prevent the router from considering the autonomous system (AS) path length as a factor in the algorithm for choosing a best path route.

Use the `no` parameter with this command to allow the router to consider the AS path length in choosing a best path route.

Command Syntax

```
bgp bestpath as-path ignore
no bgp bestpath as-path ignore
```

Parameters

None

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp bestpath as-path ignore

(config)#router bgp 100
(config-router)#no bgp bestpath as-path ignore
```

bgp bestpath compare-confed-aspath

Use this command to allow comparing of the confederation AS path length. This command specifies that the AS confederation path length must be used when available in the BGP best path decision process. It is effective only when bgp bestpath as-path ignore command has not been used.

Use the `no` parameter with this command to ignore consideration of AS confederation path length in BGP best path selection.

Command Syntax

```
bgp bestpath compare-confed-aspath

no bgp bestpath compare-confed-aspath
```

Parameters

None

Default

BGP receives routes with identical eBGP paths from eBGP peers and selects the first route received as the best path.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp bestpath compare-confed-aspath

(config)#router bgp 100
(config-router)#no bgp bestpath compare-confed-aspath
```

bgp bestpath compare-routerid

Use this command to compare router IDs for identical eBGP paths. When comparing similar routes from peers, the BGP router does not consider the router ID of the routes. By default, it selects the first received route. Use this command to include router ID in the selection process; similar routes are compared and the route with the lowest router ID is selected. The router IS is the highest IP address on the router, with preference given to loopback addresses. Router ID can be manually set by using the bgp router-id command.

Use the `no` parameter with this command to disable this functionality.

Command Syntax

```
bgp bestpath compare-routerid
```

```
no bgp bestpath compare-routerid
```

Parameters

None

Default

BGP receives routes with identical eBGP paths from eBGP peers and selects the first route received as the best path.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp bestpath compare-routerid

(config)#router bgp 100
(config-router)#no bgp bestpath compare-routerid
```

bgp bestpath dont-compare-originator-id

Use this command to change the default bestpath selection by not comparing an originator-ID for an identical EBGP path.

Use the `no` parameter with this command to disable this functionality.

Command Syntax

```
bgp bestpath dont-compare-originator-id
no bgp bestpath dont-compare-originator-id
```

Parameters

None

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp bestpath dont-compare-originator-id

(config)#router bgp 100
(config-router)#no bgp bestpath dont-compare-originator-id
```

bgp bestpath med

Use this command to specify two MED (Multi Exit Discriminator) attributes, `confed` and `missing-as-worst`.

The `confed` attribute enables MED comparison along paths learned from confederation peers. The MEDs are compared only if there is no external Autonomous System (an AS not within the confederation) in the path. If there is an external autonomous system in the path, the MED comparison is not made. For example in the following paths, the MED is not compared with Route3 as it is not in the confederation. MED is compared for Route1 and Route2 only.

```
Path1 = 32000 32004, med=4
Path2 = 32001 32004, med=2
Path3 = 32003 1, med=1
```

The `missing-as-worst` attribute to consider a missing MED attribute in a path as having a value of infinity, making the path without a MED value the least desirable path. If `missing-as-worst` is disabled, the missing MED is assigned the value of 0, making the path with the missing MED attribute the best path.

Use the `no` parameter with this command to prevent BGP from considering the MED attribute in comparing paths.

Command Syntax

```
bgp bestpath med confed missing-as-worst
bgp bestpath med (confed|missing-as-worst|remove-recv-med|remove-send-med)
bgp bestpath med missing-as-worst confed
no bgp bestpath med confed missing-as-worst
no bgp bestpath med (confed|missing-as-worst|remove-recv-med|remove-send-med)
no bgp bestpath med missing-as-worst confed
```

Parameters

`confed`	Compare MED along confederation paths
`missing-as-worst`	Treat missing MED as the least preferred one
`remove-recv-med`	Remove received MED attribute
`remove-send-med`	Remove sent MED attribute

Command Mode

Router mode

Default

MED value is zero.

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp bestpath med missing-as-worst

(config)#router bgp 100
```

```
(config-router)#bgp bestpath med remove-recv-med
(config-router)#no bgp bestpath med remove-recv-med

(config)#router bgp 100
(config-router)#bgp bestpath med remove-send-med
(config-router)#no bgp bestpath med remove-send-med
```

bgp bestpath tie-break-on-age

Use this command to always select a preferred older route even when the `bgp bestpath compare-routerid` command is configured.

Use the `no` parameter with this command to disable this functionality.

Command Syntax

```
bgp bestpath tie-break-on-age

no bgp bestpath tie-break-on-age
```

Parameters

None

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp bestpath tie-break-on-age

(config)#router bgp 100
(config-router)#no bgp bestpath tie-break-on-age
```

bgp client-to-client reflection

Use this command to configure routers as route reflectors. Route reflectors are used when all Interior Border Gateway Protocol (iBGP) speakers are not fully meshed. If the clients are fully meshed the route reflector is not required, use `no bgp client-to-client reflection` command to disable the client-to-client route reflection.

Use the `no` parameter with this command to turn off client-to-client reflection.

Command Syntax

```
bgp client-to-client reflection

no bgp client-to-client reflection
```

Parameters

None

Default

When a router is configured as a route reflector, client-to-client reflection is enabled by default.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp client-to-client reflection

(config)#router bgp 100
(config-router)#no bgp client-to-client reflection
```

bgp cluster-id

Use this command to configure the cluster ID if the BGP cluster has more than one route reflector. A cluster includes route reflectors and its clients. Usually, each cluster is identified by the router ID of its single route reflector but to increase redundancy sometimes a cluster may have more than one route reflector. All router reflectors in such a cluster are then identified by a cluster ID. The `bgp cluster-id` command is used to configure the 4 byte cluster ID for clusters with more than one route reflectors.

Use the `no` parameter with this command (without any arguments) to remove a previously configured route reflector cluster ID.

Command Syntax

```
bgp cluster-id <1-4294967295>
bgp cluster-id A.B.C.D
no bgp cluster-id
```

Parameters

`<1-4294967295>`	Route reflector ID as a 32-bit quantity
`A.B.C.D`	Route reflector ID in an IPv4 address format

Command Mode

Router mode

Examples

The following configuration creates a cluster-id 5 including two route-reflector-clients.

```
#configure terminal
(config)#router bgp 100
(config-router)#neighbor 2.2.2.2 remote-as 200
(config-router)#neighbor 3.3.3.3 remote-as 200
(config-router)#neighbor 3.3.3.3 route-reflector-client
(config-router)#neighbor 5.5.5.5 remote-as 200
(config-router)#neighbor 5.5.5.5 route-reflector-client
(config-router)#neighbor 6.6.6.6 remote-as 200
(config-router)#bgp cluster-id 5
```

bgp confederation identifier

Use this command to specify a BGP confederation identifier.

Use the `no` parameter with this command to remove a BGP confederation identifier.

Command Syntax

```
bgp confederation identifier <1-65535>
```

```
no bgp confederation identifier
```

Parameter

`<1-65535>`	Routing domain confederation AS number

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp confederation identifier 1
```

bgp confederation peers

Use this command to configure the Autonomous Systems (AS) that belong to a confederation. A confederation allows an AS to be divided into several ASs. The AS is given a confederation identifier. External BGP (eBGP) routers view only the whole confederation as one AS. Each AS is fully meshed within itself and is visible internally to the confederation.

Use the `no` parameter with this command to remove an autonomous system from the confederation.

Command Syntax

```
bgp confederation peers <1-65535>
no bgp confederation peers <1-65535>
```

Parameter

`<1-65535>`	AS numbers of eBGP peers that are in the same confederation

Command Mode

Router mode

Examples

In the following configuration example, the neighbor 172.210.30.2 and 172.210.20.1 have iBGP connection within AS 100, neighbor 173.213.30.1 is a BGP connection with a confederation peer 200 and neighbor 6.6.6.6 has an eBGP connection to external AS 300.

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp confederation identifier 300
(config-router)#bgp confederation peers 200
(config-router)#neighbor 172.210.30.2 remote-as 100
(config-router)#neighbor 172.210.20.1 remote-as 100
(config-router)#neighbor 173.213.30.1 remote-as 200
(config-router)#neighbor 6.6.6.6 remote-as 300
```

In this configuration, the neighbor 5.5.5.4 has an eBGP connection to confederation 300.

```
#configure terminal
(config)#router bgp 500
(config-router)#neighbor 5.5.5.4 remote-as 300
```

bgp config-type

Use this command to set the BGP configuration to the `standard` type. After setting the configuration to the `standard` type, use the neighbor send-community command to send out BGP community attributes. The `zebos` configuration type is the default and requires no specific configuration for sending out BGP standard community and extended community attributes.

For the `standard` type, the `no synchronization` command is always shown in the configuration, whereas for the `zebos` type, this command is the default.

Use the `no` parameter with this command to remove this configuration.

Command Syntax

```
bgp config-type (standard|zebos)
```

```
no bgp config-type
```

Parameters

`standard`	Standard style configuration
`zebos`	ZebOS-XP style configuration

Default

The default configuration type is: `bgp config-type zebos`

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#bgp config-type standard
```

bgp dampening

Use this command to enable BGP route dampening and set various parameters. Route dampening minimizes the instability caused by route flapping. A penalty is added for every flap in a flapping route. As soon as the total penalty reaches the `suppress` limit the advertisement of the route is suppressed. This penalty is decayed according to the configured `half time` value. Once the penalty is lower than the `reuse` limit, the route advertisement is un-suppressed. The dampening information is purged from the router once the penalty becomes less than half of the `reuse` limit.

Use the `no` parameter with this command to unset BGP dampening parameters.

Command Syntax

```
bgp dampening
bgp dampening <1-45>
bgp dampening <1-45> <1-20000> <1-20000> <1-255>
bgp dampening <1-45> <1-20000> <1-20000> <1-255> <1-45>
bgp dampening route-map WORD
no bgp dampening
no bgp dampening <1-45>
no bgp dampening <1-45> <1-20000> <1-20000> <1-255>
no bgp dampening <1-45> <1-20000> <1-20000> <1-255> <1-45>
no bgp dampening route-map
no bgp dampening route-map WORD
```

Parameters

`<1-45>` Reachability half-life time for the penalty in minutes. The time for the penalty to decrease to one-half of its current value.

`<1-20000>` Value to start reusing a route. When the penalty for a suppressed route decays below the reuse value, the routes become unsuppressed.

`<1-20000>` Value to start suppressing a route. When the penalty for a route exceeds the suppress value, the route is suppressed

`<1-255>` Maximum duration to suppress a stable route in minutes.

`<1-45>` Un-reachability half-life time for the penalty in minutes.

`route-map` Route map to specify criteria for dampening.

  `WORD` Route-map name.

Defaults

The default reachability half-life is 15 minutes.

The default reuse limit is 750.

The default suppress limit is 2000.

The default max-suppress value is 4 times the half-life time, or 60 minutes.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 11
(config-router)#bgp dampening 20 800 2500 80 25
```

bgp default ipv4-unicast

Use this command to configure BGP defaults and activate IPv4-unicast for a peer by default. This affects the BGP global configuration.

Use the no parameter with this command to disable the default behavior of the BGP routing process of exchanging IPv4 addressing information with BGP neighbor routers.

Command Syntax

```
bgp default ipv4-unicast

no bgp default ipv4-unicast
```

Parameters

None

Default

IPv4 unicast is the default BGP behavior.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp default ipv4-unicast
```

bgp default local-preference

Use this command to change the default local preference value.Local preference indicates the preferred path when there are multiple paths to the same destination. The path having a higher preference is preferred. The preference is sent to all routers and access servers in the local autonomous system.

Use the `no` parameter with this command to revert to the default value for local preference.

Command Syntax

```
bgp default local-preference <0-4294967295>
no bgp default local-preference
no bgp default local-preference <0-4294967295>
```

Parameter

`<0-4294967295>`	Local preference value

Default

The default local preference value is 100.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp default local-preference 2345555
```

bgp deterministic-med

Use this command to compare the Multi Exit Discriminator (MED) variable when choosing among routes advertised by different peers in the same autonomous system. MED is compared after BGP attributes weight, local preference, AS-path and origin have been compared and are equal.

For a correct comparison result, enable this command on all routers in a local AS. After enabling this command, all paths for the same prefix are grouped together and arranged according to their MED value. Based on this comparison, the best path is then chosen. This command compares MED variable when choosing routes advertised by different peers in the same AS, to compare MED, when choosing routes from neighbors in different ASs use the bgp always-compare-med command.

When the bgp deterministic-med command is enabled, routes from the same AS are grouped together, and the best routes of each group are compared. If the BGP table showed:

```
Route1: as-path 200, med 300, internal
Route2: as-path 400, med 200, internal
Route3: as-path 400, med 250, external
```

BGP would have a group of Route1 and a second group of Route2 and Route3 (the same ASs). The best of each group is compared. Route1 is the best of its group because it is the only route from AS 200. Route1 is compared to the Route2, the best of group AS 400 (the lower MED). Since the two routes are not from the same AS, the MED is not considered in the comparison. The external BGP route is preferred over the internal BGP route, making Route3 the best route; the preferred route would be different if `always-compare-med` command is enabled (See `always-compare-med` command).

Use the `no` parameter with this command to disallow this setting.

Command Syntax

```
    bgp deterministic-med
    no bgp deterministic-med
```

Parameters

None

Default

Disabled

Command Mode

Router mode

Examples

```
    #configure terminal
    (config)#router bgp 100
    (config-router)#bgp deterministic-med

    (config)#router bgp 100
    (config-router)#no bgp deterministic-med
```

bgp enforce-first-as

Use this command to enforce the first AS for eBGP routes. This command specifies that any updates received from an external neighbor that do not have the neighbor's configured Autonomous System (AS) at the beginning of the AS_PATH in the received update must be denied. Enabling this feature adds to the security of the BGP network by not allowing traffic from unauthorized systems.

Using the `no` parameter with this command to disable this feature.

Command Syntax

```
bgp enforce-first-as

no bgp enforce-first-as
```

Parameters

None

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp enforce-first-as

(config)#router bgp 100
(config-router)#no bgp enforce-first-as
```

bgp extended-asn-cap

Use this command to configure a BGP router to send 4-octet ASN capabilities. If attempting to change the AS capability from 2 to 4 or 4 to 2, a prompt occurs to remove the VRF configuration (if it exists), and reconfiguration is required, because the route distinguisher (RD) configuration would have been created with the current (2 octet or 4 octet) capability, and must be reconfigured before attempting to change the capability.

While loading from a saved configuration with AS4 capability and BGP VRF configuration, the capability will not be changed because of the above described reason.

Use the `no` parameter with this command to prevent a BGP router from sending 4-octet ASN capabilities.

Command Syntax

```
bgp extended-asn-cap

no bgp extended-asn-cap
```

Parameters

None

Default

Disabled

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#bgp extended-asn-cap
```

bgp fast-external-failover

Use this command to reset a BGP session immediately, if the interface used for BGP connection goes down.

Use the `no` parameter with this command to disable this feature.

Command Syntax

```
bgp fast-external-failover

no bgp fast-external-failover
```

Parameters

None

Default

Fast-external failover is enabled

Command Mode

Router mode

Example

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp fast-external-failover
```

bgp graceful-restart

Use this command to enable BGP graceful-restart capabilities. The restart-time parameter is used for setting the maximum time that a graceful-restart neighbor waits to come back up after a restart. This value is applied to all neighbors unless you explicitly override it by configuring the corresponding value on the neighbor. The stalepath-time parameter is used to set the maximum time to preserve stale paths from a gracefully restarted neighbor. All stalepaths, unless reinstated by the neighbor after a re-establishment, will be deleted at the expiration of this timer.

Use the `no` parameter with this command to restore the router to its default state.

Command Syntax

```
bgp graceful-restart
bgp graceful-restart graceful-reset
bgp graceful-restart restart-time <1-3600>
bgp graceful-restart stalepath-time <1-3600>
no bgp graceful-restart
no bgp graceful-restart graceful-reset
no bgp graceful-restart restart-time
no bgp graceful-restart stalepath-time
```

Parameters

`graceful-reset`	The BGP daemon is not restarted, so that any changes in network configurations that cause BGP reset do not affect packet forwarding.
`restart-time`	Maximum time needed for neighbors to restart. Default is 120 seconds.
`<1-3600>`	Delay value in seconds.
`stalepath-time`	Maximum time to retain stale paths from restarting neighbors. Default is 360 seconds.
`<1-3600>`	Delay value in seconds.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#bgp graceful-restart

#configure terminal
(config)#router bgp 10
(config-router)#no bgp graceful-restart
```

bgp g-shut

Use this command to gracefully shut down all BGP IPv4 and IPv6 sessions under this router. The BGP graceful shutdown feature reduces packet loss during maintenance activity.

Use the `no` parameter with this command to bring up all the sessions under this router after completion of maintenance activity using the `bgp g-shut` command.

For details about the graceful shutdown feature, see the *Border Gateway Protocol Developer Guide*.

Command Syntax

```
bgp g-shut

no bgp g-shut
```

Parameters

None

Default

Disabled

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp g-shut

#configure terminal
(config)#router bgp 100
(config-router)#no bgp g-shut
```

bgp g-shut-capable

Use this command to enable the graceful shutdown capability at the router level and make available the graceful-shutdown related commands at the router and BGP neighbor levels.

Use the `no` parameter with this command to disable the graceful shutdown capability on a router.

For details about the graceful shutdown capability, see the *Border Gateway Protocol Developer Guide*.

Note: The graceful shutdown capability cannot be disabled on a router that is in a graceful shutdown state until it comes out this state--after the graceful shutdown has been initiated and the impacted BGP sessions are up again.

Command Syntax

```
bgp g-shut-capable
no bgp g-shut-capable
```

Parameters

None

Default

By default, the graceful shutdown capability is disabled at the router level.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp g-shut-capable

#configure terminal
(config)#router bgp 100
(config-router)#no bgp g-shut-capable
```

bgp g-shut-local-preference

Use this command to configure the local preference value of the router to be used during graceful shutdown. The local preference value is used to indicate the preferred path when there are multiple paths to the same destination in a single routing database. The path having a higher preference value is the preferred one. The preferred path is sent to all routers and access servers in the local autonomous system.

Use the `no` parameter with this command to revert to the default setting.

Command Syntax

```
bgp g-shut-local-preference <0-4294967295>
no bgp g-shut-local-preference
```

Parameters

`<0-4294967295>`	Local preference value

Default

By default, the local preference value is set to 0.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp g-shut-local-preference 22
```

bgp log-neighbor-changes

Use this command to enable logging of status change messages without turning on debug bgp commands. ZebOS-XP has many logging services for neighbor status, including `debug bgp fsm` and `debug bgp events`. However, these commands cause system performance degradation. If you need to log neighbor status changes only, IP Infusion Inc. recommends turning off all debug commands and using the `bgp log-neighbor-changes` command instead. A sample output of the log is:

```
%Protocol-Severity-Events: Message-text
```

A sample output of the log for an `interface down` event is:

```
%BGP-5-ADJCHANGE: neighbor 10.10.0.24 Down Interface flap
```

This command logs the following events:

- BGP Notification Received
- Erroneous BGP Update Received
- User reset request
- Peer time-out
- Peer Closing down the session
- Interface flap
- Router ID changed
- Neighbor deleted
- Member added to peer group
- Administrative shutdown
- Remote AS changed
- RR client configuration modification
- Soft reconfiguration modification

Use the `no` parameter with this command to disable this feature.

Command Syntax

```
bgp log-neighbor-changes

no bgp log-neighbor-changes
```

Parameters

None

Default

Disabled

Command Mode

Router mode

Example

```
(config)#router bgp 100
(config-router)#bgp log-neighbor-changes
```

bgp multiple-instance

Use this command to enable BGP multiple instance support.

Use the `no` parameter with this command to disable this function.

Note: The `no bgp multiple-instance` command is not valid when any BGP instances are present.

Command Syntax

```
bgp multiple-instance (allow-same-peer|)
no bgp multiple-instance (allow-same-peer|)
```

Parameters

`allow-same-peer`

Allow the same peer in multiple instances

Default

By default, there is no multiple-instance support in BGP.

Command Mode

Configure mode

Examples

The following example shows the use of the `bgp multiple-instance` command allowing the configuration of two instances.

```
(config)#bgp multiple-instance

(config)#quit
#show run

Current configuration:
hostname ZebOS-XP
password zebra
log stdout
!
debug bgp
debug bgp events
debug bgp updates
debug bgp fsm
!
bgp multiple-instance
!
router bgp 11
 bgp router-id 10.10.10.50
 neighbor 10.10.10.51 remote-as 11
!
```

bgp nexthop-trigger delay

Use this command to set the delay time for nexthop address tracking. This command configures the delay interval between routing table walks for nexthop delay tracking, after which BGP does a routing table scan on receiving a nexthop change trigger from NSM. The time period determines how long BGP waits before it walks the full BGP table to determine which prefixes are affected by the nexthop changes, after it receives the trigger from NSM about one or more nexthop changes.

Use the `no` parameter with this command to reset the timer value to the default value.

Command Syntax

```
bgp nexthop-trigger delay <1-100>
no bgp nexthop-trigger delay
```

Parameter

`<1-100>`	Nexthop trigger delay interval in seconds

Default

The default nexthop-trigger delay time is 5 seconds.

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#bgp nexthop-trigger delay 6

#configure terminal
(config)#no bgp nexthop-trigger delay
```

bgp nexthop-trigger enable

Use this command to enable nexthop address tracking. Nexthop address tracking is an event-driven notification system that monitors the status of routes installed in the Routing Information Base (RIB) and reports nexthop changes that affect internal BGP (iBGP) or external BGP (eBGP) prefixes directly to the BGP process. This improves the overall BGP convergence time, by allowing BGP to respond rapidly to nexthop changes for routes installed in the RIB.

If nexthop tracking is enabled after certain routes are learned, the registration of all nexthops for selected BGP routes is done after the nexthop tracking feature is enabled. If nexthop tracking is disabled, and if there are still some selected BGP routes, BGP de-registers the nexthops of all selected BGP routes from NSM.

Use the `no` parameter with this command to disable this feature. If the `no` command is given when nexthop tracking is in the process of execution, an error appears and nexthop tracking is not disabled. However, if the nexthop tracking timer is running at the time of negation, the nexthop tracking timer is stopped, and nexthop tracking is disabled.

Command Syntax

```
bgp nexthop-trigger enable
```

```
no bgp nexthop-trigger enable
```

Parameters

None

Default

Nexthop address tracking is disabled by default.

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#bgp nexthop-trigger enable
```

bgp rfc1771-path-select

Use this command to set RFC 1771 compatible path selection.

Use the `no` parameter with this command to revert this setting.

Command Syntax

```
bgp rfc1771-path-select
no bgp rfc1771-path-select
```

Parameters

None

Default

Standard compatible path selection mechanism.

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#bgp rfc1771-path-select
```

bgp rfc1771-strict

Use this command to set the origin path attribute to “IGP” when the origin is a protocol such as RIP, OSPF, or ISIS as specified in RFC 1771. Otherwise, the origin is always set to “incomplete” which is the industry standard.

Use the `no` parameter with this command to revert this setting.

Command Syntax

```
bgp rfc1771-strict
```

```
no bgp rfc1771-strict
```

Parameters

None

Default

Disabled

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#bgp rfc1771-strict
```

bgp router-id

Use this command to manually configure a fixed router ID as a BGP router identifier. When this command is used to configure a fixed router ID, the current router identifier is overridden and the peers are reset.

Use the `no` parameter with this command to remove a manually configured fixed router ID.

Command Syntax

```
bgp router-id A.B.C.D
no bgp router-id
no bgp router-id A.B.C.D
```

Parameter

`A.B.C.D`	Router ID in an IPv4 address format

Default

When a loopback interface is configured, the router ID is set to the IP address of the loopback interface. If no loopback interface is configured, the highest IP address is the router-id.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp router-id 10.1.2.7

(config)#router bgp 100
(config-router)#no bgp router-id 10.1.2.7
```

bgp scan-time

Use this command to configure scanning intervals of BGP routers. This interval is the period after which router checks the validity of the routes in its database. To disable BGP scanning, set the scan-time interval to 0 seconds.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
bgp scan-time <0-60>

no bgp scan-time

no bgp scan-time <0-60>
```

Parameter

`<0-60>`	Scanning interval in seconds

Default

The default scan-time interval is 60 seconds.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp scan-time 10
```

bgp update-delay

Use this command to specify the update-delay value for a graceful-restart capable router. The update-delay value is the maximum time a graceful-restart capable router, which is restarting, will defer route-selection and advertisements to all its graceful-restart capable neighbors. This maximum time starts from the instance the first neighbor attains established state after restart. The restarting router prematurely terminates this timer when end-of-rib markers are received from all its graceful-restart capable neighbors.

Use the no parameter with this command to revert to the default update-delay value.

Command Syntax

```
bgp update-delay <1-3600>
no bgp update-delay
no bgp update-delay <1-3600>
```

Parameters

<1-3600>	Delay interval in seconds

Default

The default update-delay value is 120 seconds.

Command Mode

Router mode

Example

```
#configure terminal
(config)#router bgp 10
(config-router)#bgp update-delay 345
```

clear bgp (A.B.C.D|X:X::X:X)

Use this command to reset a BGP neighbor address.

Command Syntax

```
clear bgp (A.B.C.D|X:X::X:X)
clear bgp (A.B.C.D|X:X::X:X) in
clear bgp (A.B.C.D|X:X::X:X) in prefix-filter
clear bgp (A.B.C.D|X:X::X:X) out
clear bgp (A.B.C.D|X:X::X:X) soft
clear bgp (A.B.C.D|X:X::X:X) soft in
clear ip bgp (A.B.C.D|X:X::X:X)
```

Parameters

`A.B.C.D`	IPv4 neighbor address.
`X:X::X:X`	IPv6 neighbor address.
`in`	Clear incoming advertised routes.
`prefix-filter`	Push out prefix-list ORF and do inbound soft reconfig.
`out`	Clear outgoing advertised routes.
`soft`	Clear both incoming and outgoing routes.
`in`	Soft reconfig inbound update.

Command Mode

Privileged Exec mode

Example

```
#clear bgp 3.3.3.3
```

clear bgp *

Use this command to reset the BGP connection for all peers.

Command Syntax

```
clear bgp *
clear bgp * in
clear bgp * in prefix-filter
clear bgp * out
clear bgp * soft
clear bgp * soft in
clear bgp * soft out
clear ip bgp *
clear ip bgp * in
clear ip bgp * in prefix-filter
clear ip bgp * out
clear ip bgp * soft
clear ip bgp * soft in
clear ip bgp * soft out
clear ip bgp * ipv4 (unicast|multicast) in
clear ip bgp * ipv4 (unicast|multicast) in prefix-filter
clear ip bgp * ipv4 (unicast|multicast) out
clear ip bgp * ipv4 (unicast|multicast) soft
clear ip bgp * ipv4 (unicast|multicast) soft in
clear ip bgp * ipv4 (unicast|multicast) soft out
```

Parameters

`in`	Incoming advertised routes should be cleared.
`prefix-filter`	Push out prefix-list ORF and do inbound soft reconfig.
`out`	Clear outgoing advertised routes.
`soft`	Clear both incoming and outgoing routes.
`in`	Soft reconfig inbound update.
`out`	Soft reconfig outbound update.
`ipv4`	Clear incoming advertised routes.
`multicast`	Multicast prefixes.
`unicast`	Unicast prefixes.
`in`	Clear incoming advertised routes.
`prefix-filter`	

	Push out prefix-list ORF and do inbound soft reconfig.
out	Clear outgoing advertised routes.
soft	Clear both incoming and outgoing routes.
in	Soft reconfig inbound update.
out	Soft reconfig outbound update.

Command Mode

Privileged Exec mode

Examples

```
#clear bgp *
#clear ip bgp * ipv4 unicast in prefix-filter
```

clear bgp <1-4294967295>

Use this command to reset a BGP connection for all peers in a specified Autonomous System.

Command Syntax

```
clear bgp <1-4294967295>
clear bgp <1-4294967295> in
clear bgp <1-4294967295> in prefix-filter
clear bgp <1-4294967295> out
clear bgp <1-4294967295> soft
clear bgp <1-4294967295> soft in
clear bgp <1-4294967295> soft out
clear ip bgp <1-4294967295>
clear ip bgp <1-4294967295> in
clear ip bgp <1-4294967295> in prefix-filter
clear ip bgp <1-4294967295> out
clear ip bgp <1-4294967295> soft
clear ip bgp <1-4294967295> soft in
clear ip bgp <1-4294967295> soft out
clear ip bgp <1-4294967295> ipv4 (unicast|multicast) in
clear ip bgp <1-4294967295> ipv4 (unicast|multicast) in prefix-filter
clear ip bgp <1-4294967295> ipv4 (unicast|multicast) out
clear ip bgp <1-4294967295> ipv4 (unicast|multicast) soft
clear ip bgp <1-4294967295> ipv4 (unicast|multicast) soft in
clear ip bgp <1-4294967295> ipv4 (unicast|multicast) soft out
```

Parameters

`in`	Clear incoming advertised routes.
`prefix-filter`	
	Push out prefix-list ORF and do inbound soft reconfig.
`out`	Clear outgoing advertised routes.
`soft`	Clear both incoming and outgoing routes.
`in`	Soft reconfig inbound update.
`out`	Soft reconfig outbound update.
`ipv4`	Clear incoming advertised routes.
`multicast`	Multicast prefixes.
`unicast`	Unicast prefixes.
	Clear incoming advertised routes.
`in`	Clear incoming advertised routes.

`prefix-filter`

Push out prefix-list ORF and do inbound soft reconfig.

`out` Clear outgoing advertised routes.

`soft` Clear both incoming and outgoing routes.

`in` Soft reconfig inbound update.

`out` Soft reconfig outbound update.

Command Mode

Privileged Exec mode

Examples

```
#clear bgp 4294967277
#clear ip bgp 4294967277
```

clear bgp <1-65535>

Use this command to reset a BGP connection for all peers in a specified Autonomous System.

Command Syntax

```
clear bgp <1-65535>
clear bgp <1-65535> in
clear bgp <1-65535> in prefix-filter
clear bgp <1-65535> out
clear bgp <1-65535> soft
clear bgp <1-65535> soft in
clear bgp <1-65535> soft out
clear ip bgp <1-65535>
clear ip bgp <1-65535> in
clear ip bgp <1-65535> in prefix-filter
clear ip bgp <1-65535> out
clear ip bgp <1-65535> soft
clear ip bgp <1-65535> soft in
clear ip bgp <1-65535> soft out
clear ip bgp <1-65535> in prefix-filter
clear ip bgp <1-65535> ipv4 (unicast|multicast) in
clear ip bgp <1-65535> ipv4 (unicast|multicast) in prefix-filter
clear ip bgp <1-65535> ipv4 (unicast|multicast) out
clear ip bgp <1-65535> ipv4 (unicast|multicast) soft
clear ip bgp <1-65535> ipv4 (unicast|multicast) soft in
clear ip bgp <1-65535> ipv4 (unicast|multicast) soft out
```

Parameters

`in` Clear incoming advertised routes.

  `prefix-filter`

  Push out prefix-list ORF and do inbound soft reconfig.

`out` Clear outgoing advertised routes.

`soft` Clear both incoming and outgoing routes.

  `in` Soft reconfig inbound update.

  `out` Soft reconfig outbound update.

`ipv4` Clear incoming advertised routes.

  `multicast` Multicast prefixes.

  `unicast` Unicast prefixes.

Clear incoming advertised routes.

`in` Clear incoming advertised routes.

`prefix-filter`

Push out prefix-list ORF and do inbound soft reconfig.

`out` Clear outgoing advertised routes.

`soft` Clear both incoming and outgoing routes.

`in` Soft reconfig inbound update.

`out` Soft reconfig outbound update.

Command Mode

Privileged Exec mode

Examples

```
#clear bgp 100
#clear ip bgp 200
```

clear bgp dampening

Use this command to reset BGP route flap dampening information.

Command Syntax

```
clear bgp ipv4 (unicast|multicast) dampening
clear bgp ipv4 (unicast|multicast) dampening A.B.C.D
clear bgp ipv4 (unicast|multicast) dampening A.B.C.D/M
clear ip bgp dampening
clear ip bgp dampening A.B.C.D
clear ip bgp dampening A.B.C.D/M
clear ip bgp ipv4 (unicast|multicast) dampening
clear ip bgp ipv4 (unicast|multicast) dampening A.B.C.D
clear ip bgp ipv4 (unicast|multicast) dampening A.B.C.D/M
```

Parameters

`ipv4`	IPv4 address family.
`multicast`	Multicast prefixes
`unicast`	Unicast prefixes
`A.B.C.D`	IP prefix (network), for example, 35.0.0.0
`A.B.C.D/M`	IP prefix (network/length), for example, 35.0.0.0/8

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp dampening 10.10.0.121
#clear ip bgp ipv4 unicast dampening
```

clear bgp external

Use this command to reset the BGP connection for all external peers.

Command Syntax

```
clear bgp external
clear bgp external in
clear bgp external in prefix-filter
clear bgp external out
clear bgp external soft
clear bgp external soft in
clear bgp external soft out
clear ip bgp external
clear ip bgp external in
clear ip bgp external in prefix-filter
clear ip bgp external out
clear ip bgp external soft
clear ip bgp external soft in
clear ip bgp external soft out
clear ip bgp external ipv4 (unicast|multicast) in
clear ip bgp external ipv4 (unicast|multicast) in prefix-filter
clear ip bgp external ipv4 (unicast|multicast) out
clear ip bgp external ipv4 (unicast|multicast) soft
clear ip bgp external ipv4 (unicast|multicast) soft in
clear ip bgp external ipv4 (unicast|multicast) soft out
```

Parameters

`in` Clear incoming advertised routes.

  `prefix-filter`

  Push out prefix-list ORF and do inbound soft reconfig.

`out` Clear outgoing advertised routes.

`soft` Clear both incoming and outgoing routes.

  `in` Soft reconfig inbound update.

  `out` Soft reconfig outbound update.

`ipv4` Clear incoming advertised routes.

  `multicast` Multicast prefixes.

  `unicast` Unicast prefixes.

  `in` Clear incoming advertised routes.

    `prefix-filter`

Push out prefix-list ORF and do inbound soft reconfig.

`out` Clear outgoing advertised routes.

`soft` Clear both incoming and outgoing routes.

- `in` Soft reconfig inbound update.
- `out` Soft reconfig outbound update.

Command Mode

Privileged Exec mode

Example

```
#clear ip bgp external
```

clear bgp flap-statistics

Use this command to reset BGP flap statistics.

Command Syntax

```
clear bgp ipv4 (unicast|multicast) flap-statistics
clear bgp ipv4 (unicast|multicast) flap-statistics A.B.C.D
clear bgp ipv4 (unicast|multicast) flap-statistics A.B.C.D/M
clear ip bgp flap-statistics
clear ip bgp flap statistics A.B.C.D
clear ip bgp flap-statistics A.B.C.D/M
clear ip bgp ipv4 (unicast|multicast) flap-statistics
clear ip bgp ipv4 (unicast|multicast) flap-statistics A.B.C.D
clear ip bgp ipv4 (unicast|multicast) flap-statistics A.B.C.D/M
```

Parameters

`ipv4`	IPv4 address family.
`multicast`	Multicast prefixes.
`unicast`	Unicast prefixes.
`A.B.C.D`	IP prefix (network) for example, 35.0.0.0
`A.B.C.D/M`	IP prefix (network/length), for example, 35.0.0.0/8

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp flap-statistics 10.10.0.121
#clear ip bgp ipv4 unicast flap-statistics
```

clear bgp peer-group

Use this command to reset the BGP connection for all members of a peer group.

Command Syntax

```
clear bgp peer-group WORD
clear bgp peer-group WORD in
clear bgp peer-group WORD in prefix-filter
clear bgp peer-group WORD out
clear bgp peer-group WORD soft
clear bgp peer-group WORD soft in
clear bgp peer-group WORD soft out
clear ip bgp peer-group WORD
clear ip bgp peer-group WORD in
clear ip bgp peer-group WORD in prefix-filter
clear ip bgp peer-group WORD out
clear ip bgp peer-group WORD soft
clear ip bgp peer-group WORD soft in
clear ip bgp peer-group WORD soft out
clear ip bgp peer-group WORD ipv4 (unicast|multicast) in
clear ip bgp peer-group WORD ipv4 (unicast|multicast) in prefix-filter
clear ip bgp peer-group WORD ipv4 (unicast|multicast) out
clear ip bgp peer-group WORD ipv4 (unicast|multicast) soft
clear ip bgp peer-group WORD ipv4 (unicast|multicast) soft in
clear ip bgp peer-group WORD ipv4 (unicast|multicast) soft out
```

Parameters

`WORD`	BGP peer-group name.
`in`	Clear incoming advertised routes.
`prefix-filter`	Push out prefix-list ORF and do inbound soft reconfig.
`out`	Clear outgoing advertised routes.
`soft`	Clear both incoming and outgoing routes.
`in`	Soft reconfig inbound update.
`out`	Soft reconfig outbound update.
`ipv4`	Clear incoming advertised routes.
`multicast`	Multicast prefixes.
`unicast`	Unicast prefixes.
	Clear incoming advertised routes.

`in` Clear incoming advertised routes.

`prefix-filter`

Push out prefix-list ORF and do inbound soft reconfig.

`out` Clear outgoing advertised routes.

`soft` Clear both incoming and outgoing routes.

`in` Soft reconfig inbound update.

`out` Soft reconfig outbound update.

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp peer-group P1
```

clear bgp view

Use this command to reset all peers in a BGP view.

Command Syntax

```
clear bgp view WORD *
clear bgp view WORD * soft
clear bgp view WORD * soft in
clear bgp view WORD * soft out
clear ip bgp view WORD *
clear ip bgp view WORD * in prefix-filter
clear ip bgp view WORD * soft
clear ip bgp view WORD * soft in
clear ip bgp view WORD * soft out
clear ip bgp view WORD * ipv4 (unicast|multicast) in prefix-filter
clear ip bgp view WORD * ipv4 (unicast|multicast) soft
clear ip bgp view WORD * ipv4 (unicast|multicast) soft in
clear ip bgp view WORD * ipv4 (unicast|multicast) soft out
```

Parameters

`WORD`	BGP peer group name.
`in`	Clear incoming advertised routes.
`prefix-filter`	Push out prefix-list ORF and do inbound soft reconfig.
`soft`	Clear both incoming and outgoing routes.
`in`	Soft reconfig inbound update.
`out`	Soft reconfig outbound update.
`ipv4`	IPv4 address family.
`multicast`	Multicast prefixes.
`unicast`	Unicast prefixes.
`in`	Clear incoming advertised routes.
`prefix-filter`	Push out prefix-list ORF and do inbound soft reconfig.
`soft`	Clear both incoming and outgoing routes.
`in`	Soft reconfig inbound update.
`out`	Soft reconfig outbound update.

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp view myview *
```

clear ip bgp A.B.C.D

Use this command to reset an IPv4 BGP neighbor address.

Command Syntax

```
clear ip bgp A.B.C.D in
clear ip bgp A.B.C.D in prefix-filter
clear ip bgp A.B.C.D out
clear ip bgp A.B.C.D soft
clear ip bgp A.B.C.D soft in
clear ip bgp A.B.C.D soft out
clear ip bgp A.B.C.D ipv4 (unicast|multicast) in
clear ip bgp A.B.C.D ipv4 (unicast|multicast) in prefix-filter
clear ip bgp A.B.C.D ipv4 (unicast|multicast) out
clear ip bgp A.B.C.D ipv4 (unicast|multicast) soft
clear ip bgp A.B.C.D ipv4 (unicast|multicast) soft in
clear ip bgp A.B.C.D ipv4 (unicast|multicast) soft out
```

Parameters

`in`	Clear incoming advertised routes.
`prefix-filter`	Push out prefix-list ORF and do inbound soft reconfig.
`out`	Clear outgoing advertised routes.
`soft`	Clear both incoming and outgoing routes.
`in`	Soft reconfig inbound update.
`out`	Soft reconfig outbound update.
`ipv4`	Clear incoming advertised routes.
`multicast`	Multicast prefixes.
`unicast`	Unicast prefixes.
`in`	Clear incoming advertised routes.
`prefix-filter`	Push out prefix-list ORF and do inbound soft reconfig.
`out`	Clear outgoing advertised routes.
`soft`	Clear both incoming and outgoing routes.
`in`	Soft reconfig inbound update.
`out`	Soft reconfig outbound update.

Command Mode

Privileged Exec mode

Examples

```
    #clear ip bgp 35.0.0.1 in
```

clear ip bgp A.B.C.D vrf

Use this command to reset the VPN Routing/Forwarding (VRF) instance for a peer address.

Command Syntax

```
clear ip bgp A.B.C.D vrf WORD
clear ip bgp A.B.C.D vrf WORD in
clear ip bgp A.B.C.D vrf WORD out
clear ip bgp A.B.C.D vrf WORD soft
clear ip bgp A.B.C.D vrf WORD soft in
clear ip bgp A.B.C.D vrf WORD soft out
```

Parameters

`A.B.C.D`	IPv4 address
`WORD`	VPN routing/forwarding instance name
`in`	Clear incoming advertised routes
`out`	Clear outgoing advertised routes
`soft`	Clear both incoming and outgoing routes
`in`	Soft reconfig inbound update
`out`	Soft reconfig outbound update

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp 35.0.0.1 vrf
```

clear ip bgp table-map

Use this command to apply the modified table map or route map rules to the BGP routes in the existing IP routing table.

Command Syntax

```
clear ip bgp table-map (vrf (VRFNAME|all|default))
clear ip bgp ipv4 (unicast | multicast) table-map(vrf (VRFNAME|all|default))
```

Parameters

`vrf`	Select a VPN Routing/Forwarding Instance.
`VRFNAME`	Specify a VPN Routing/Forwarding instance name.
`all`	Select all VRFs.
`default`	Select default VRFs.
`unicast`	Unicast prefixes.
`multicast`	Multicast prefixes.

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp table-map vrf all
```

debug bgp

Use this command to enable all BGP troubleshooting functions. Use this command without any parameters to turn on normal bgp debug information.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
    debug bgp (all|)
    debug bgp bfd
    debug bgp dampening
    debug bgp events
    debug bgp filters
    debug bgp fsm
    debug bgp keepalives
    debug bgp mpls
    debug bgp nht
    debug bgp nsm
    debug bgp updates
    debug bgp updates (in|out)
    debug bgp vpls
    no debug bgp (all|)
    no debug bgp bfd
    no debug bgp dampening
    no debug bgp events
    no debug bgp filters
    no debug bgp fsm
    no debug bgp keepalives
    no debug bgp mpls
    no debug bgp nht
    no debug bgp nsm
    no debug bgp updates
    no debug bgp vpls
    undebug bgp (all|)
    undebug bgp bfd
    undebug bgp dampening
    undebug bgp events
    undebug bgp filters
    undebug bgp fsm
    undebug bgp keepalives
```

```
undebug bgp mpls
undebug bgp nht
undebug bgp nsm
undebug bgp updates
undebug bgp vpls
```

Parameters

`all`	Used only with the `no` form; turns off all debugging for BGP
`bfd`	Enable debugging for BGP Bidirectional Forwarding Detection
`dampening`	Enable debugging for BGP dampening
`events`	Enable debugging for BGP events
`filters`	Enable debugging for BGP filters
`fsm`	Enable debugging for BGP Finite State Machine (FSM)
`keepalives`	Enable debugging for BGP keepalives
`mpls`	Enable debugging for BGP Multiprotocol Label Switching (MPLS)
`nht`	Enable debugging for BGP NHT
`nsm`	Enable debugging for NSM messages
`updates`	Enable debugging for BGP updates
`in`	Debug inbound updates
`out`	Debug outbound updates
`vpls`	Enable debugging for BGP Virtual Private LAN Service (VPLS)

Command Mode

Privileged Exec mode and Configure Mode

Examples

```
#debug bgp
#debug bgp events
```

distance bgp

Use this command to define an administrative distance. A distance is a rating of trustworthiness of a router. The higher the distance the lower the trust rating. Administrative distances can be set for external, internal and local routes. External paths are routes learned from a neighbor outside of the AS. Internal routes are routes learned from another router within the same AS. Local routes are for a router that is redistributed from another process.

If the administrative distance is changed, it could create inconsistency in the routing table and obstruct routing. Use this command in Router mode to set the administrative distance for all address families. Use this command in Address Family mode to set the administrative distance per an IPv4 or IPv6 family.

Use the `no` parameter with this command to remove an administrative distance.

Command Syntax

```
distance bgp <1-255> <1-255> <1-255>
```

```
no distance bgp
```

```
no distance bgp <1-255> <1-255> <1-255>
```

Parameters

`<1-255>`	Distance for BGP external routes
`<1-255>`	Distance for BGP internal routes
`<1-255>`	Distance for BGP local routes

Command Mode

Router mode, Address Family IPv4 mode, and Address Family IPv6 mode

Defaults

Default distance for external routes is 20.

Default distance for internal routes is 200.

Default distance for local routes is 200.

Examples

The following example shows how to set the administrative distance for BGP for all address families.

```
#configure terminal
(config)#router bgp 100
(config-router)#distance bgp 34 23 15
```

The following example shows how to set the administrative distance for BGP for an IPv6 address family.

```
(config)#router bgp 100
(config-router)#address family ipv6
(config-router-af)#distance bgp 34 23 14
```

dump bgp all

Use this command to dump all BGP packets.

Use the `no` option with this command to disable this function.

Command Syntax

```
dump bgp all PATH
dump bgp all PATH INTERVAL
no dump bgp all PATH INTERVAL
```

Parameters

`PATH`	Output filename
`INTERVAL`	Interval for output of BGP packets

Command Mode

Configure mode

Example

```
#configure terminal
(config)#dump bgp all pathfilename
```

dump bgp routes-mrt

Use this command to dump the entire BGP routing table.

Use the no option with this command to disable this feature.

Command Syntax

```
dump bgp routes-mrt PATH

dump bgp routes-mrt PATH INTERVAL

no dump bgp routes-mrt PATH INTERVAL
```

Parameters

PATH	Output filename
INTERVAL	Interval for dumping BGP packets

Command Mode

Configure mode

Example

```
#configure terminal
(config)#dump bgp route-mrt pathfilename
```

dump bgp updates

Use this command to dump BGP updates only.

Use the `no` option with this command to disable this function.

Command Syntax

```
dump bgp updates PATH
dump bgp updates PATH INTERVAL
no dump bgp updates PATH INTERVAL
```

Parameters

`PATH`	Output filename
`INTERVAL`	Interval for dumping BGP packets

Command Mode

Configure mode

Example

```
#configure terminal
(config)#dump bgp updates pathfilename
```

exit-address-family

Use this command to exit Address-Family-vrf, Address-Family-vpnv4, or Address-Family-vpnv6 mode.

For information on how to enter the address family mode (IPv4, IPv6, VPNv4, or VPNv6), see address-family.

Command Syntax

```
exit-address-family
```

Parameters

None

Command Mode

Address Family-vrf, Address Family-vpnv4 and Address Family-vpnv6 mode.

Examples

The following examples shows the change in the prompt after using this command.

```
#configure terminal
(config)#router bgp 100
(config-router)#address-family ipv4 multicast
(config-router-af)#exit-address-family
(config-router)#

(config)#router bgp 100
(config-router)#address-family vpnv6 unicast
(config-router-af)#exit-address-family
```

ip as-path access-list

Use this command to define a BGP Autonomous System (AS) path access list. A named community list is a filter based on regular expressions. If the regular expression matches the specified string representing the AS path of the route, then the permit or deny condition applies. Use this command to define the BGP access list globally; use the neighbor router configuration command to apply a specific access list.

Use the no parameter with this command to disable use of the access list.

Command Syntax

```
ip as-path access-list WORD (deny|permit) LINE
no ip as-path access-list WORD
no ip as-path access-list WORD (deny|permit) LINE
```

Parameters

`WORD`	Access list name
`deny`	Reject packets
`permit`	Forward packets
`LINE`	An ordered list as a regular expression

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#ip as-path access-list mylist deny ^65535$
```

ip community-list <1-99>

Use this command to specify a standard community list (1 to 99) that specifies BGP community attributes.

Use the `no` parameter with this command to delete the community list entry.

Command Syntax

```
ip community-list <1-99> (deny|permit)
ip community-list <1-99> (deny|permit) [AA:NN|local-AS|no-advertise|no-export]
ip community-list <1-99> (deny|permit) LINE
no ip community-list <1-99> (deny|permit)
no ip community-list <1-99> (deny|permit) [AA:NN|local-AS|no-advertise|no-export]
no ip community-list <1-99> (deny|permit) LINE
```

Parameters

`deny`	Reject the community
`permit`	Accept the community
`AA:NN`	Community number
`local-AS`	Do not advertise routes to external BGP peers
`no-advertise`	Do not advertise routes to other BGP peers
`no-export`	Do not advertise routes outside of Autonomous System boundary
`LINE`	An ordered list as a regular expression

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#ip community-list 55 permit 7675:80 7675:90

(config)#no ip community-list 55 permit 7675:80 7675:90
```

ip community-list <100-199>

Use this command to specify an expanded community list (100 to 199) that specifies BGP community attributes.

Use the `no` parameter with this command to delete the community list entry.

Command Syntax

```
ip community-list <100-199> (deny|permit)
ip community-list <100-199> (deny|permit) LINE
no ip community-list <100-199>
no ip community-list <100-199> (deny|permit) LINE
```

Parameters

deny	Reject community
permit	Accept community
LINE	An ordered list as a regular expression

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#ip community-list 125 permit 6789906
(config)#ip community-list expanded CLIST permit .*
```

ip community-list expanded

Use the community-lists to specify BGP community attributes. The community attribute is used for implementing policy routing. It is an optional, transitive attribute and facilitates transfer of local policies through different autonomous systems. It includes community values that are 32 bits long.

There are two kinds of community-lists: expanded and standard. The standard community-list defines the community attributes in a specified format and not with regular expressions. The expanded community-list defines the community attributes with regular expressions.Use the `no` parameter with this command to delete the community list entry.

Command Syntax

```
ip community-list expanded WORD (deny|permit) LINE
no ip community-list expanded WORD
no ip community-list expanded WORD (deny|permit) LINE
```

Parameters

`WORD`	Community list name
`deny`	Reject community
`permit`	Accept community
`LINE`	An ordered list as a regular expression

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#ip community-list 125 permit 6789906
(config)#ip community-list expanded CLIST permit .*
```

ip community-list standard

Use the community-lists to specify BGP community attributes. The community attribute is used for implementing policy routing. It is an optional, transitive attribute and facilitates transfer of local policies through different autonomous systems. It includes community values that are 32 bits long. There are two kinds of community-lists: expanded and standard. The standard community-list defines the community attributes in a specified format without regular expressions. The expanded community-list defines the community attributes with regular expressions.

Use this command to add a standard community-list entry. The standard community-list is compiled into binary format and is directly compared with the BGP communities attribute in the BGP updates. The comparison is faster than the expanded community-list. Any community value that does not match the standard community value is automatically treated as expanded.

Use the `no` parameter with this command to delete the standard community-list entry.

Command Syntax

```
ip community-list standard WORD (deny|permit)
```

```
ip community-list standard WORD (deny|permit) [AA:NN|local-AS|no-advertise|no-
  export]
```

```
no ip community-list standard WORD (deny|permit) [AA:NN|local-AS|no-advertise|no-
  export]
```

Parameters

`WORD`	Community list name
`deny`	Reject the community
`permit`	Accept the community
`AA:NN`	Community number
`local-AS`	Do not advertise routes to external BGP peers
`no-advertise`	Do not advertise routes to other BGP peers
`no-export`	Do not advertise routes outside of Autonomous System boundary

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#ip community-list standard CLIST permit 7675:80 7675:90 no-export
(config)#ip community-list 34 permit 5675:50 no-advertise
```

ip community-list WORD

Use the community-list commands to specify BGP community attributes. The community attribute is used for implementing policy routing. It is an optional, transitive attribute and facilitates transfer of local policies through different autonomous systems. There are two kinds of community-lists: the expanded and standard. The `standard community-list` defines the community attributes in a specified format and not with regular expressions. The `expanded community-list` defines the community attributes with regular expressions.

Use the `no` parameter with this command to delete the community list entry.

Command Syntax

```
ip community-list WORD (deny|permit)
ip community-list WORD (deny|permit) [AA:NN|local-AS|no-advertise|no-export]
no ip community-list WORD
no ip community-list WORD (deny|permit) [AA:NN|local-AS|no-advertise|no-export]
```

Parameters

`WORD`	Community list name
`deny`	Reject the community
`permit`	Accept the community
`AA:NN`	Community number
`local-AS`	Do not advertise routes to external BGP peers
`no-advertise`	Do not advertise routes to other BGP peers
`no-export`	Do not advertise routes outside of Autonomous System boundary

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#ip community-list mylist permit 7675:80 7675:90

(config)#no ip community-list mylist permit 7675:80 7675:90
```

ip extcommunity-list <1-99>

Use this command to create an entry for a standard extended community list.

Use the `no` parameter with this command to delete the community-list entry.

Command Syntax

```
ip extcommunity-list <1-99> (deny|permit) LINE
no ip extcommunity-list <1-99> (deny|permit) LINE
```

Parameters

`deny`	Reject community
`permit`	Accept community
`LINE`	An ordered list as a regular expression

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#ip extcommunity-list 3 permit 4567335
```

ip extcommunity-list <100-199>

Use this command to create an expanded extended community list.

Use the `no` parameter with this command to delete the community-list entry.

Command Syntax

```
    ip extcommunity-list <100-199> (deny|permit) LINE

    no ip extcommunity-list <100-199> (deny|permit) LINE
```

Parameters

`<100-199>`	Extended community list number (expanded)
`deny`	Reject the community
`permit`	Accept the community
`LINE`	An ordered list as a regular expression

Command Mode

Configure mode

Examples

```
    #configure terminal
    (config)#ip extcommunity-list 125 permit 4567335
```

ip extcommunity-list expanded

Use this command to create an expanded extended community list.

Use the `no` parameter with this command to delete the extended community-list entry.

Command Syntax

```
ip extcommunity-list expanded WORD
ip extcommunity-list expanded WORD (deny|permit) LINE
no ip extcommunity-list expanded WORD
no ip extcommunity-list expanded WORD (deny|permit) LINE
```

Parameters

`WORD`	Extended community list name
`deny`	Reject the community
`permit`	Accept the community
`LINE`	An ordered list as a regular-expression

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#ip extcommunity-list 125 permit 4567335
(config)#ip extcommunity-list expanded CLIST permit .*
```

ip extcommunity-list standard

Use this command to create and delete a standard extended-community list. The extended community attribute is 8 bytes in 2 formats. The sub-type can be route target (`rt`) or site of origin (`soo`). Thus, the sub-type of each community must be specified when creating the extended community list. Regarding the formats, an extended community is based on the 6 byte value; these 6 bytes are represented in 4btyes:2bytes format:

- Format 1, aa.nn: The 16 bit value of the AS number is represented in higher-order 4 bytes. If the extended ASN capability is enabled, the AS number is represented using higher-order 4 bytes. The NN assigned value is represented in low-order 2 bytes in both cases.
- Format 2, IPaddr:nn: In this format, the higher-order 4 bytes are used to represent the IP address, and the low-order 2 bytes are used to represent the assigned value.

Use the `no` parameter with this command to delete the extended-community-list entry.

Command Syntax

```
ip extcommunity-list standard WORD (deny|permit) LINE
ip extcommunity-list standard WORD (deny|permit) LINE [rt|soo]
no ip extcommunity-list standard WORD
no ip extcommunity-list standard WORD (deny|permit) LINE
no ip extcommunity-list standard WORD (deny|permit) LINE [rt|soo]
```

Parameters

`WORD`	Extended community list name
`deny`	Reject the community
`permit`	Accept the community
`LINE`	An ordered list as a regular-expression.
`rt`	Route target extended community in aa:nn or IPaddr:nn format
`soo`	Site-of-origin extended community in aa:nn or IPaddr:nn format

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#ip extcommunity-list 36 permit rt 5675:50
(config)#ip extcommunity-list standard CLIST permit soo 7645:70
```

match ip peer

Use this command to apply policies based on the route source of which the BGP TCP/IP session formed using an IPv4 address in the update message.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
match ip peer (<1-199>|<1300-2699>|WORD)
```

```
no match ip peer (<1-199>|<1300-2699>|WORD)
```

Parameters

`<1-199>`	IP access-list number
`<1300-2699>`	IP access-list number (expanded range)
`WORD`	Access-list name

Command Mode

Route-map mode

Examples

```
#configure terminal
(config)#route-map in-A permit 10
(route-map)#match ip peer 1
```

max-paths

Use this command to set the number of equal-cost multi-path (ECMP) routes for eBGP or iBGP. You can install multiple BGP paths to the same destination to balance the load on the forwarding path.

Use the `no` parameter with this command to disable this feature.

For more information about ECMP for BGP, see the *Border Gateway Protocol Developer Guide*.

Command Syntax

```
max-paths (ebgp|ibgp|) <2-64>

no max-paths ebgp (<2-64>|)

no max-paths ibgp (<2-64>|)
```

Parameters

`ebgp`	eBGP ECMP session
`ibgp`	iBGP ECMP session
`<2-64>`	Number of routes

Default

Available for the default BGP instance and for IPv4 and IPv6 unicast addresses

Command Mode

Router mode and Address Family mode

Examples

The following example configures 7 routes for ECMP for iBGP.

```
#configure terminal
(config)#router bgp 11
(config-router)#max-paths ibgp 7
```

mpls-resolution

Use this command to allow BGP to query the NSM for preexisting LSPs (from RSVP or LDP), enabling BGP to map prefixes to these LSPs. For example, BGP peer (5.5.5.5/32) advertises a prefix 10.10.10.10/32 to the current BGP daemon. If `mpls-resolution` is enabled, BGP queries NSM to confirm if there is an LSP from current router to 5.5.5.5/32. If yes, BGP installs this IP route into the NSM, and also creates an FTN entry in the NSM. NSM subsequently installs this in the MPLS Forwarder and uses the `outgoing label`, `ifindex` and `nexthop` data of the LSP. This allows IP packets destined for 10.10.10.10/32 to be `mapped` onto a preexisting MPLS LSP.

If the LSP is already up, the mapping is done while BGP is installing IP routes to the NSM. If the LSP is removed after BGP has already mapped a prefix to an LSP, the mapping is withdrawn from the NSM.

The `no mpls-resolution` resets all peer information in BGP, and BGP needs to set up connections with its peers again. Since the `mpls-resolution` flag is not set, no mapping takes place.

Command Syntax

```
mpls-resolution
no mpls-resolution
```

Parameters

None

Command Mode

Router mode

Example

```
#configure terminal
(config)#router bgp 100
(config-router)#mpls-resolution
```

neighbor activate

Use this command to enable the exchange of specific AF routes with a neighboring router. After the TCP connection is opened with the neighbor, use this command to enable or disable the exchange of AF information with a neighboring router. To enable the exchange of multicast and VPNv4 address prefix types, neighbors are activated using the `neighbor activate` command in address family mode.

Use the `no` parameter with this command to disable exchange of information with a neighbor.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) activate
no neighbor (A.B.C.D|X:X::X:X|WORD) activate
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Default

A neighbor under address-family IPv4/IPv6 is activated by default. For all other address-families, use this command to enable a neighbor to exchange routing information of a specific address-family with a neighbor.

Command Mode

Address Family mode and Router mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 1.2.3.4 activate

(config)#router bgp 100
(config-router)#neighbor 10.10.20.1 remote-as 100
(config-router)#address-family vpnv4 unicast
(config-router-af)#neighbor 10.10.20.1 activate

(config)#router bgp 100
(config)#bgp router-id 13.13.13.13
(config-router)#neighbor 3ffe:15:15:15:15::0 remote-as 100
(config-router)#address-family vpnv6 unicast
(config-router-af)#neighbor 3ffe:15:15:15:15::0 activate
```

neighbor advertisement-interval

Use this command to set a minimum interval between the sending of BGP routing updates. To reduce the flapping of routes to internet, a minimum advertisement interval is set, so that the BGP routing updates are sent only per interval seconds. BGP dampening can also be used to control the effects of flapping routes.

Use the `no` parameter with this command to set the interval time to default.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) advertisement-interval <0-65535>

no neighbor (A.B.C.D|X:X::X:X|WORD) advertisement-interval

no neighbor (A.B.C.D|X:X::X:X|WORD) advertisement-interval <0-65535>
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`<0-65535>`	Advertisement-interval value in seconds

Command Mode

Router mode

Example

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.3 advertisement-interval 45
```

neighbor allowas-in

Use this command to advertise prefixes (routes) even when the source of the prefixes is from the same Autonomous System (AS) number.

Use this command in a scenario where two routers at different locations use the same Autonomous System number and are connected via an ISP. Once prefixes arrive from one branch at the ISP, they are tagged with the customer's AS number. By default, when the ISP passes the prefixes to the other router, the prefixes are dropped if the other router uses the same AS number. Use this command to advertise the prefixes at the other side. Control the number of times an AS number is advertised by specifying a number.

In a hub and spoke configuration in a VPN, a PE (Provider Edge) router advertises all prefixes containing duplicate AS numbers. Use this command to configure two VRFs on each PE router to receive and advertise prefixes. One of the VRFs receives prefixes with AS numbers from all PE routers and then advertises them to neighboring PE routers. The other VRF receives prefixes with AS numbers from the CE (Customer Edge) router and advertises them to all PE routers in the hub and spoke configuration.

Use the `no` parameter with this command to reset to default.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) allowas-in
neighbor (A.B.C.D|X:X::X:X|WORD) allowas-in <1-10>
no neighbor (A.B.C.D|X:X::X:X|WORD) allowas-in
```

Parameters

`A.B.C.D`	IPv4 neighbor address.
`X:X::X:X`	IPv6 neighbor address.
`WORD`	Name of peer group.

Note: For information on how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies on all peers in the specified group.

`<1-10>`	Number of times to allow the advertisement of an AS number

Command Mode

Router mode and Address Family mode

Example

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.3 allowas-in 4

#configure terminal
(config)#router bgp 7657
(config-router)#address-family ipv4 vrf VRF_A
(config-router-af)#neighbor 10.10.0.1 allowas-in 3

#configure terminal
(config)#router bgp 7657
(config-router)#address-family ipv6 vrf VRF_A
(config-router-af)#neighbor 3ffe:15:15:15:15::0 allowas-in 3
```

neighbor as-origination-interval

Use this command to adjust the interval of sending AS origination routing updates. This command is used to change the minimum interval between the sending of AS-origination routing updates.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) as-origination-interval <1-65535>
no neighbor (A.B.C.D|X:X::X:X|WORD) as-origination-interval
no neighbor (A.B.C.D|X:X::X:X|WORD) as-origination-interval <1-65535>
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`<1-65535>`	AS origination interval in seconds

Command Mode

Router mode

Example

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.75 as-origination-interval 555
```

neighbor attribute-unchanged

Use this command to advertise unchanged BGP attributes to the specified neighbor.

Use the no parameter with this command to disable this function.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) attribute-unchanged ({ as-path|next-hop|med }|)

no neighbor (A.B.C.D|X:X::X:X|WORD) attribute-unchanged (({ as-path|next-hop|
  med }|)
```

Parameters

A.B.C.D	Address of the BGP neighbor in an IPv4 format
X:X::X:X	Address of the BGP neighbor in an IPv6 format
WORD	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

as-path	AS path attribute
next-hop	Nexthop attribute
med	Multi-exit discriminator attribute

Command Mode

Router mode and Address Family mode

Example

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.75 attribute-unchanged as-path med
```

neighbor capability dynamic

Use this command to enable the dynamic capability for a specific peer. This command allows a BGP speaker to advertise or withdraw an address family capability to a peer in a non-disruptive manner.

Use the `no` parameter with this command to disable the dynamic capability.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) capability dynamic
no neighbor (A.B.C.D|X:X::X:X|WORD) capability dynamic
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Default

Disabled

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.10.1 capability dynamic
```

neighbor capability graceful-restart

Use this command to configure the router to advertise the Graceful Restart Capability to the neighbors. This configuration indicates that the BGP speaker has the ability to preserve its forwarding state for the address family when BGP restarts. Use this command to advertise to the neighbor routers the capability of graceful restart. However, users must first specify a neighbor's `remote-as` identification number assigned by the neighbor router.

Note: The graceful restart capability is advertised only when the graceful restart capability has been enabled using the `bgp graceful-restart` command.

Use the `no` parameter with this command to configure router so it does not advertise the Graceful Restart Capability to its neighbor.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) capability graceful-restart
no neighbor (A.B.C.D|X:X::X:X|WORD) capability graceful-restart
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode and Address Family mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.10.50 capability graceful-restart
```

neighbor capability orf prefix-list

Use this command to enable Outbound Router Filtering (ORF), and advertise the ORF capability to its neighbors. The ORFs send and receive capabilities to lessen the number of updates exchanged between neighbors. By filtering updates, this option minimizes generating and processing of updates. The local router advertises the ORF capability in `send` mode, and the remote router receives the ORF capability in `receive` mode applying the filter as outbound policy. The two routers exchange updates to maintain the ORF for each router. Only an individual router or a peer group can be configured to be in `receive` or `send mode`. A peer-group member cannot be configured to be in `receive` or `send` mode.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) capability orf prefix-list (both|receive|send)
no neighbor (A.B.C.D|X:X::X:X|WORD) capability orf prefix-list (both|receive|send)
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`both`	The local router can send ORF entries to its peer, as well as receive ORF entries from its peer.
`receive`	The local router is willing to receive ORF entries from its peer
`send`	The local router is willing to send ORF entries to its peer

Command Mode

Router mode and Address Family (IPv4 unicast, IPv4 multicast, IPv6) mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.5 capability orf prefix-list both
(config-router)#neighbor effe:2897::0003:3ed5 capability orf prefix-list
receive
```

neighbor capability route-refresh

Use this command to advertise to peer about route refresh capability support. If route refresh capability is supported, then router can dynamically request that the peer re-advertises its Adj-RIB-Out.

Use the `no` parameter with this command to disable this function

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) capability route-refresh
```

```
no neighbor (A.B.C.D|X:X::X:X|WORD) capability route-refresh
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.10.1 capability route-refresh
```

neighbor collide-established

Use this command to include a neighbor already in an established state for conflict resolution when a TCP connection collision is detected. This command is not required for most network deployments, so users should only use this command when required.

Note: The associated functionality of including an 'established' neighbor into TCP connection collision conflict resolution is automatically enabled when a neighbor is configured for BGP graceful-restart.

Use the `no` option with this command to turn this feature off.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) collide-established

no neighbor (A.B.C.D|X:X::X:X|WORD) collide-established
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode

Example

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 3.3.3.3 collide-established
```

neighbor connection-retry-time

Use this command to set the connection retry time for a specific BGP neighbor.

Use the `no` parameter with this command to clear the connection retry time for a specific BGP neighbor.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) connection-retry-time <1-65535>
no neighbor (A.B.C.D|X:X::X:X|WORD) connection-retry-time
no neighbor (A.B.C.D|X:X::X:X|WORD) connection-retry-time <1-65535>
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`<1-65535>`	Connection retry time in seconds

Default

The default connection retry time is 120 seconds.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 12
(config-router)#neighbor 10.10.10.10 connection-retry-time 125
```

neighbor default-originate

Use this command to allow a BGP local router to send the default route 0.0.0.0 to a neighbor for use as a default route. This command can be used with standard or extended access lists.

Use the `no` parameter with this command to send no route as a default.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) default-originate

neighbor (A.B.C.D|X:X::X:X|WORD) default-originate route-map WORD

no neighbor (A.B.C.D|X:X::X:X|WORD) default-originate

no neighbor (A.B.C.D|X:X::X:X|WORD) default-originate route-map WORD
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`WORD`	Route map name

Command Mode

Router mode and Address Family

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.10.1 default-originate route-map myroute
```

neighbor description

Use this command to associate a description with a neighbor. This command helps in identifying a neighbor quickly. It is useful for an ISP that has multiple neighbor relationships.

Use the `no` parameter with this command to remove the description.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) description LINE
no neighbor (A.B.C.D|X:X::X:X|WORD) description
no neighbor (A.B.C.D|X:X::X:X|WORD) description LINE
```

Parameters

`A.B.C.D` Address of the BGP neighbor in an IPv4 format

`X:X::X:X` Address of the BGP neighbor in an IPv6 format

`WORD` Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`LINE` Neighbor description (up to 80 characters)

Command Mode

Router mode and Address Family

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 1.2.3.4 description Backup router for sales

(config)#router bgp 100
(config-router)#address-family ipv4 vrf VRF_A
(config-router-af)#neighbor 10.10.0.1 description Bank of America

(config)#router bgp 100
(config-router)#address-family ipv6 vrf VRF_A
(config-router-af)#neighbor 3ffe:15:15:15:15::0 description Bank of America
```

neighbor disallow-infinite-holdtime

Use this command to disallow configuration of infinite hold-time. This command enables the local BGP speaker to reject a hold-time of “0” seconds from the peer (during exchange of open messages) or the user (during configuration).

The `no` form of this command allows the BGP speaker to accept a hold-time of “0” from the peer or during configuration.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) disallow-infinite-holdtime
no neighbor (A.B.C.D|X:X::X:X|WORD) disallow-infinite-holdtime
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode

Examples

```
(config-router)#neighbor 10.11.4.26 disallow-infinite-holdtime
(config-router)#neighbor 3ffe::45 disallow-infinite-holdtime
```

neighbor distribute-list

Use this command to filter route updates from a particular BGP neighbor. Use only one distribute-list per BGP neighbor.

Use the `no` parameter with this command to remove an entry.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) distribute-list (<1-199>|<1300-2699>|WORD)
  (in|out)
```

```
no neighbor (A.B.C.D|X:X::X:X|WORD) distribute-list (<1-199>|<1300-2699>|WORD)
  (in|out)
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`<1-199>`	IP access-list number
`<1300-2699>`	IP access-list number (expanded-range)
`WORD`	Access-list name
`in`	Filter incoming advertised routes
`out`	Filter outgoing advertised routes

Command Mode

Router mode and Address Family mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 1.2.3.4 distribute-list mylist out
```

neighbor dont-capability-negotiate

Use this command to disable capability negotiation. The capability negotiation is performed by default. This command is used to allow compatibility with older BGP versions that have no capability parameters used in open messages between peers.

Use the `no` parameter with this command to enable capability negotiation.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) dont-capability-negotiate

no neighbor (A.B.C.D|X:X::X:X|WORD) dont-capability-negotiate
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.34 dont-capability-negotiate
```

neighbor ebgp-multihop

Use this command to accept and attempt BGP connections to external peers on indirectly connected networks. Multihop is not established if the only route to the multihop peer is a default route. This avoids loop formation.

Use the `no` parameter with this command to return to the default.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) ebgp-multihop

neighbor (A.B.C.D|X:X::X:X|WORD) ebgp-multihop <1-255>

no neighbor (A.B.C.D|X:X::X:X|WORD) ebgp-multihop

no neighbor (A.B.C.D|X:X::X:X|WORD) ebgp-multihop <1-255>
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`<1-255>`	Maximum hop count

Default

If no maximum hop count is set, this is set to 255.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.10.34 remote-as 20
(config-router)#neighbor 10.10.10.34 ebgp-multihop 5
```

neighbor enforce-multihop

Use this command to turn on the enforcement of eBGP neighbors perform multihop.

Use the `no` parameter with this command to turn off this feature.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) enforce-multihop
no neighbor (A.B.C.D|X:X::X:X|WORD) enforce-multihop
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.34 remote-as 20
(config-router)#neighbor 10.10.0.34 enforce-multihop
```

neighbor fall-over bfd

Use this command to configure bidirectional forwarding detection (BFD) for BGP.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) fall-over bfd (multihop|)
no neighbor (A.B.C.D|X:X::X:X|WORD) fall-over bfd (multihop|)
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`multihop`	Enable multihop

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.34 remote-as 20
(config-router)#neighbor 10.10.0.34 fall-over bfd multihop
```

neighbor filter-list

Use this command to set up a BGP filter. This command specifies an access list filter on updates based on the BGP autonomous system paths. Each filter is an access list based on regular expressions.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) filter-list WORD (in|out)
no neighbor (A.B.C.D|X:X::X:X|WORD) filter-list WORD (in|out)
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`WORD`	Name of an autonomous system path access list
`in`	Filter incoming advertised routes
`out`	Filter outgoing advertised routes

Command Mode

Router mode and Address Family mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.34 remote-as 20
(config-router)#neighbor 10.10.0.34 filter-list out in
```

neighbor g-shut

Use this command to start a graceful shutdown for the BGP session of the specified BGP neighbor. The BGP session for this neighbor is shut down after the graceful shutdown timer expires.

If there is no alternate path available for traffic to flow prior the actual shutdown of the BGP session, this path is made available for 60 seconds or for configured time after which the path is no longer available and traffic is dropped.

Use the `no` parameter with this command to bring up the session again for the specified BGP neighbor whose BGP session had been shut down using the `neighbor g-shut` command.

Note: The graceful shutdown capability is not supported on iBGP sessions.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) g-shut
no neighbor (A.B.C.D|X:X::X:X|WORD) g-shut
```

Parameters

`A.B.C.D`	Neighbor IPv4 address
`X:X::X:X|`	Neighbor IPv6 address
`WORD`	Neighbor tag

Default

Disabled

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#neighbor 1.1.1.2 g-shut

#configure terminal
(config)#router bgp 100
(config-router)#no neighbor 1.1.1.2 g-shut
```

neighbor g-shut-timer

Use this command to configure the value of the graceful shutdown timer. After the timer expires, the BGP session initiated for graceful shutdown is shut down.

Use the `no` parameter with this command to revert to the default setting.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) g-shut-timer <10-65535>
no neighbor (A.B.C.D|X:X::X:X|WORD) g-shut-timer <10-65535>
```

Parameters

`A.B.C.D`	Neighbor IPv4 address
`X:X::X:X|`	Neighbor IPv6 address
`WORD`	Neighbor tag
`<10-65535>`	Graceful shutdown timer in seconds

Default

By default, the timer value is set to 60 seconds.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#neighbor 1.1.1.2 g-shut-timer 120
```

neighbor local-as

Use this command to specify an AS (autonomous system) number to use with BGP neighbor.

Use the no parameter with this command to disable this command.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) local-as <1-4294967295>
no neighbor (A.B.C.D|X:X::X:X|WORD) local-as <1-4294967295>
```

Parameters

A.B.C.D	Address of the BGP neighbor in an IPv4 format
X:X::X:X	Address of the BGP neighbor in an IPv6 format
WORD	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies on all peers in the specified group.

<1-4294967295>

Neighbor’s AS number when extended capabilities are configured

Note: The AS number 23456 is a reserved 2-octet AS number. An old BGP speaker (2-byte implementation) should be configured with 23456 as its remote AS number while peering with a non-mappable new BGP speaker (4-byte implementation).

Command Mode

Router mode

Example

```
#configure terminal
(config)#router bgp 11
(config-router)#neighbor 10.10.0.34 local-as 12345
```

neighbor maximum-prefix

Use this command to control the number of prefixes that can be received from a neighbor. This command allows the configuration of a specified number of prefixes that a BGP router is allowed to receive from a neighbor. When the `warning-only` option is not used and extra prefixes are received, the router ends the peering. A terminated peer stays down until the `clear ip bgp command` is used.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) maximum-prefix <1-4294967295>
neighbor (A.B.C.D|X:X::X:X|WORD) maximum-prefix <1-4294967295> <1-100>
neighbor (A.B.C.D|X:X::X:X|WORD) maximum-prefix <1-4294967295> <1-100> warning-only
neighbor (A.B.C.D|X:X::X:X|WORD) maximum-prefix <1-4294967295> warning-only
no neighbor (A.B.C.D|X:X::X:X|WORD) maximum-prefix
no neighbor (A.B.C.D|X:X::X:X|WORD) maximum-prefix <1-4294967295>
no neighbor (A.B.C.D|X:X::X:X|WORD) maximum-prefix <1-4294967295> warning-only
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies on all peers in the specified group.

`<1-4294967295>`	Maximum number of prefixes accepted from this peer
`<1-100>`	Threshold value percent <1-100>
`warning-only`	Only give a warning message when the limit is exceeded

Command Mode

Router mode and Address Family mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.72 maximum-prefix 1244 warning-only
```

neighbor next-hop-self

Use this command to configure the router as the next hop for a BGP-speaking neighbor or peer group. This command allows a BGP router to change the nexthop information that is sent to the iBGP peer. The nexthop information is set to the IP address of the interface used to communicate with the neighbor.

Use the `no` parameter with this command to disable this feature.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) next-hop-self

no neighbor (A.B.C.D|X:X::X:X|WORD) next-hop-self
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode and Address Family mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.72 remote-as 100
(config-router)#neighbor 10.10.0.72 next-hop-self
```

neighbor override-capability

Use this command to override a capability negotiation result.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) override-capability
no neighbor (A.B.C.D|X:X::X:X|WORD) override-capability
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 12
(config-router)#neighbor 10.10.10.10 override-capability
```

neighbor passive

Use this command to set a BGP neighbor as passive.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) passive

no neighbor (A.B.C.D|X:X::X:X|WORD) passive
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 12
(config-router)#neighbor 10.10.10.10 passive
```

neighbor password

Use this command to enable message digest5 (MD5) authentication on a TCP connection between two BGP peers. Configuring MD5 authentication between two BGP peers, means that each segment sent on the TCP connection between the peers is verified. MD5 authentication must be configured with the same password on both BGP peers; otherwise, the connection between them will not be established.

Use the `no` parameter with this command to delete the MD5 authentication

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) password <WORD>
no neighbor (A.B.C.D|X:X::X:X|WORD) password <WORD>
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group
`WORD`	Password (Maximum length is 80 charecters)

Command Mode

Router mode

Example

```
#configure terminal
(config)#router bgp 11
(config-router)#neighbor 10.10.0.73 password ipi

(config-router)#no neighbor 10.10.0.73 password ipi
```

neighbor peer-group

Use this command to add a neighbor to an existing peer group. Neighbors with the same update policies are grouped into peer groups. This facilitates the updates of various policies, such as distribute and filter lists. The peer group is then configured easily with any of the neighbor commands. Any changes made to the peer group affect all members. To create a peer group, use the `neighbor WORD peer-group` command, and then use this command to add neighbors to the group.

Use the no parameter with this command to remove a neighbor from a named peer group.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) peer-group WORD

no neighbor (A.B.C.D|X:X::X:X|WORD) peer-group WORD
```

Parameters

A.B.C.D	Address of the BGP neighbor in an IPv4 format
X:X::X:X	Address of the BGP neighbor in an IPv6 format
WORD	Name of the BGP peer group
WORD	Peer group name

Command Mode

Router mode

Example

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor group1 peer-group
```

neighbor port

Use this command to specify the BGP port number of a neighbor.

Use the `no` parameter with this command to remove a port number from a BGP neighbor.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) port <0-65535>
no neighbor (A.B.C.D|X:X::X:X|WORD) port
no neighbor (A.B.C.D|X:X::X:X|WORD) port <0-65535>
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`<0-65535>`	Port number

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 12
(config-router)#neighbor 10.10.10.10 port 643
```

neighbor prefix-list

Use this command to specify a prefix list for filtering BGP advertisements. Filtering by prefix list matches the prefixes of routes with those listed in the prefix list. If there is a match, the route is used. An empty prefix list permits all prefixes. If a given prefix does not match any entries of a prefix list, the route is denied access. When multiple entries of a prefix list match a prefix, the entry with the smallest sequence number is considered to be a real match.

The router begins the search at the top of the prefix list, with the sequence number 1. Once a match or deny occurs, the router does not need to go through the rest of the prefix list. For efficiency the most common matches or denies are listed at the top. The `neighbor distribute-list` command is an alternative to this command and only one of them can be used for filtering to the same neighbor in any direction.

Use the `no` parameter with this command to remove an entry.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) prefix-list WORD (in|out)
no neighbor (A.B.C.D|X:X::X:X|WORD) prefix-list WORD (in|out)
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`WORD`	Name of an access list
`in`	Apply access list to incoming advertisements
`out`	Apply access list to outgoing advertisements

Command Mode

Router mode and Address Family mode

Examples

```
#configure terminal
(config)#ip prefix-list list1 deny 30.0.0.0/24
(config)#router bgp 12
(config-router)#neighbor 10.10.10.10 prefix-list list1 in
```

neighbor remote-as

This command establishes BGP peering with a customer edge router.

Use this command to specify a neighbor's autonomous system number. If the specified ASN matches the ASN number specified in the router bgp global configuration, the neighbor is identified as internal. If the ASN does no match, it is identified as external to the local AS.

The specified neighbor only exchanges unicast address prefixes, unless the neighbor is also activated using the `neighbor activate` command, which allows the exchange of other routing information.

Use the `no` parameter with this command to delete this peering.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) remote-as <1-65535>

neighbor (A.B.C.D|X:X::X:X|WORD) remote-as <1-4294967295>

no neighbor (A.B.C.D|X:X::X:X|WORD) remote-as <1-65535>

no neighbor (A.B.C.D|X:X::X:X|WORD) remote-as <1-4294967295>
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`<1-4294967295>`

Neighbor's AS number when extended capabilities are configured

Note: `ASNUM` 23456 is a reserved 2-octet AS number. An old BGP speaker (2-byte implementation) should be configured with 23456 as its remote AS number while peering with a non-mappable new BGP speaker (4-byte implementation).

`<1-65535>`	Neighbor's AS number

Command Mode

Router mode and Address Family-vrf mode

Example

```
#configure terminal
(config)#router bgp 11
(config-router)#neighbor 10.10.0.73 remote-as 345
(config-router)#neighbor 11.11.0.74 remote-as 23456
```

Note: The last command in the example above should be used when the local speaker is OBGP and the neighbor is NBGP with a 4-octet ASN.

```
(config)#router bgp 100
(config-router)#address-family ipv4 vrf VRF_A
(config-router-af)#neighbor 10.10.0.1 remote-as 65000
```

```
(config)#router bgp 100
(config-router)#address-family ipv6 vrf VRF_A
(config-router-af)#neighbor 3ffe:15:15:15:15::0 remote-as 65000
```

neighbor remove-private-AS

Use this command to remove the private Autonomous System (AS) number from outbound updates. Private AS numbers are not advertised to the Internet. This command is used with external BGP peers only. The router removes the AS numbers only if the update includes private AS numbers. If the update includes both private and public AS numbers, the system treats it as an error.

Use the `no` parameter with this command to revert to default.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) remove-private-AS

no neighbor (A.B.C.D|X:X::X:X|WORD) remove-private-AS
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Default

Disabled

Command Mode

Router mode and Address Family mode

Example

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.63 remove-private-AS
```

neighbor restart-time

Use this command to set a different restart-time other than the global restart-time configured using the bgp graceful-restart command. This command takes precedence over the restart-time value specified using the bgp graceful-restart command. The restart-time value is the maximum time that a graceful-restart neighbor waits to come back up after a restart. The default value is 120 seconds. Make sure that the restart time specified using this command does not exceed the stalepath-time specified in the Router mode.

Use the `no` parameter with this command to restore the router to its default state.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) restart-time <1-3600>
```

```
no neighbor (A.B.C.D|X:X::X:X|WORD) restart-time <1-3600>
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`<1-3600>`	Delay value

Command Mode

Router mode

Example

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 3.3.3.3 restart-time 45
```

neighbor route-map

Use this command to apply a route map to incoming or outgoing routes. This command filters updates and modifies attributes. A route map is applied to inbound or outbound updates. Only the routes that pass the route map are sent or accepted in updates.

Use the `no` parameter with this command to a route map.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) route-map WORD (in|out)
no neighbor (A.B.C.D|X:X::X:X|WORD) route-map WORD (in|out)
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`WORD`	Name of the route map
`in`	Apply access list to incoming advertisements
`out`	Apply access list to outgoing advertisements

Command Mode

Router mode and Address Family mode

Examples

The following example shows the configuration of the route map named `rmap2` and then the use of this map name in the `neighbor route-map` command.

```
#configure terminal
(config)#route-map rmap2 permit 6
(config-route-map)#match origin incomplete
(config-route-map)#set metric 100
(config-route-map)#exit
(config)#router bgp 12
(config-router)#neighbor 10.10.10.10 route-map rmap2 in
```

neighbor route-reflector-client

Use this command to configure the router as a BGP route reflector and configure the specified neighbor as its client.

Route reflectors are a solution for the explosion of iBGP peering within an autonomous system. By route reflection the number of iBGP peers within an AS is reduced. Use this command to configure the local router as the route reflector and specify neighbors as its client. An AS can have more than one route reflector. One route reflector treats the other route reflector as another iBGP speaker.

Use the `no` parameter with this command to indicate that the neighbor is not a client.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) route-reflector-client
no neighbor (A.B.C.D|X:X::X:X|WORD) route-reflector-client
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode and Address Family mode

Examples

In the following configuration, Router1 is the route reflector for clients 3.3.3.3 and 2.2.2.2; it also has a non-client peer 6.6.6.6.

```
#configure terminal
(config)#router bgp 200
(config-router)#neighbor 3.3.3.3 remote-as 200
(config-router)#neighbor 3.3.3.3 route-reflector-client
(config-router)#neighbor 2.2.2.2 remote-as 200
(config-router)#neighbor 2.2.2.2 route-reflector-client
(config-router)#neighbor 6.6.6.6 remote-as 200
```

neighbor route-server-client

Use this command to configure a neighbor as the route server client.

Use the `no` parameter with this command to remove the configuration of a neighbor as route server client.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) route-server-client
no neighbor (A.B.C.D|X:X::X:X|WORD) route-server-client
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode

Example

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.72 route-server-client

#configure terminal
(config)#router bgp 10
(config-router)#no neighbor 10.10.0.72 route-server-client
```

neighbor send-community

Use this command to specify that a community attribute should be sent to a BGP neighbor. The community attribute groups destinations in a certain community and applies routing decisions according to those communities. On receiving community attributes, the router reannounces them to the neighbor. Only when the `no` parameter is used with this command the community attributes are not reannounced to the neighbor. By default, both `standard` and `extended` community attributes are sent to a neighbor. To explicitly send only the `standard` or `extended` community attribute, run the `bgp config-type` command with the `standard` parameter, before running this command.

Use the `no` parameter with this command to remove the entry. Use the `extended` and `no` parameters to remove extended communities. Specifying no other parameter with `no` removes standard communities only.

See also neighbor send-community in Chapter 4, *BGP Virtual Private Network Commands*.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) send-community

neighbor (A.B.C.D|X:X::X:X|WORD) send-community (both|extended|standard)

no neighbor (A.B.C.D|X:X::X:X|WORD) send-community

no neighbor (A.B.C.D|X:X::X:X|WORD) send-community (both|extended|standard)
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`both`	Send Standard and Extended Community attributes
`extended`	Send Extended Community attributes
`standard`	Send Standard Community attributes

Default

Both `standard` and `extended` community attributes are sent to a neighbor.

Command Mode

Router mode and Address Family mode

Examples

```
#configure terminal
(config)#bgp config-type standard
(config)#router bgp 10
(config-router)#neighbor 10.10.0.72 send-community extended
```

neighbor send-label explicit-null

Use this command to exchange explicit –null label for the specific AF routes advertised between the neighbors. The AFI, SAFI combination of [1, 4] is the associated capability parameter (labelled-unicast) and is enabled by this command. This command is viable only on the ipv4 unicast and ipv4 vrf address families. This command has to be configured on both the neighbors for the capability to be negotiated.

Use the no parameter with this command to disable exchange of labels and remove the associated capability parameter.

Command Syntax

```
Neighbor x.x.x.x send-label explicit-null
No neighbor x.x.x.x send-label explicit-null
```

Parameters

`x.x.x.x`	Address of the BGP neighbor in IPv4 format

Default

By default, only the IPV4 unicast capability is enabled. Only configuration of this command on both neighbors will enable the Labelled Unicast capability.

Command mode

Address family and router mode

Example

```
# configure terminal
(config)# router bgp 100
(config-router)#neighbor 192.168.0.1 send-label explicit-null

# configure terminal
(config)#router bgp 100
(config-router)#address-family ipv4 vrf VRF-1
    (config-router-af)#neighbor 192.168.0.3 send-label explicit-null
```

neighbor shutdown

This command disables a neighbor administratively.

Use this command to terminate any active session for a specified neighbor and clear all related routing information. In case a peer group is specified for shutdown, a large number of peering sessions could be terminated. The `show ip bgp summary` command displays the summary of BGP neighbors and their connections.

Use the `no` parameter with this command to re-enable a neighbor.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) shutdown
no neighbor (A.B.C.D|X:X::X:X|WORD) shutdown
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode and Address Family-vrf mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.72 shutdown

(config)#router bgp 100
(config-router)#address-family ipv6 vrf VRF_A
(config-router-af)#neighbor 3ffe:15:15:15:15::0 shutdown
```

neighbor soft-reconfiguration inbound

Use this command to store updates for inbound soft reconfiguration. Soft-reconfiguration may be used in lieu of BGP route refresh capability. Using this command enables local storage of all the received routes and their attributes. This requires additional memory. When a soft reset (inbound) is done on this neighbor, the locally stored routes are re-processed according to the inbound policy. The BGP neighbor connection is not affected.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) soft-reconfiguration inbound
```

```
no neighbor (A.B.C.D|X:X::X:X|WORD) soft-reconfiguration inbound
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode and Address Family mode

Examples

```
#configure terminal
(config)#router bgp 12
(config-router)#neighbor 10.10.10.10 soft-reconfiguration inbound
```

neighbor strict-capability-match

Use this command to close the BGP connection if capability value does not match the remote peer.

Use the no parameter with this command to disable this function.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) strict-capability-match

no neighbor (A.B.C.D|X:X::X:X|WORD) strict-capability-match
```

Parameters

A.B.C.D	Address of the BGP neighbor in an IPv4 format
X:X::X:X	Address of the BGP neighbor in an IPv6 format
WORD	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 12
(config-router)#neighbor 10.10.10.10 strict-capability-match
```

neighbor timers

Use this command to set the timers for a specific BGP neighbor. Keepalive messages are sent by a router to inform another router that the BGP connection between the two is still active. The keepalive interval is the period of time between each keepalive message sent by the router. The holdtime interval is the time the router waits to receive a keepalive message and if it does not receive a message for this period it declares the neighbor dead.

Use the `no` parameter with this command to clear the timers for a BGP neighbor.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) timers <0-65535> <0-65535>
neighbor (A.B.C.D|X:X::X:X|WORD) timers connect <1-65535>
no neighbor (A.B.C.D|X:X::X:X|WORD) timers
no neighbor (A.B.C.D|X:X::X:X|WORD) timers connect
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`<0-65535>`	Keepalive interval value
`<0-65535>`	Holdtime value
`connect`	BGP connect timer
`<1-65535>`	Connect timer

Defaults

The default keepalive timer value is 180 seconds

The configured holdtime value should be atleast 3 times the keepalive time

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 12
(config-router)#neighbor 10.10.10.10 timers 60 230
(config-router)#neighbor 10.10.10.10 timers connect 10

(config-router)#no neighbor 10.10.10.10 timers
```

neighbor transparent-as

Use this command to specify not to append your AS path number even if the peer is an eBGP peer.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) transparent-as
```

Parameters

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 12
(config-router)#neighbor 10.10.7.1 transparent-as
```

neighbor transparent-nexthop

Use this command to keep the nexthop value of the route even if the peer is an eBGP peer.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) transparent-nexthop
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 12
(config-router)#neighbor 10.10.10.10 transparent-nexthop
```

neighbor unsuppress-map

Use this command to selectively leak more-specific routes to a particular neighbor. When the `aggregate-address` command is used with the `summary-only` option, the more-specific routes of the aggregate are suppressed to all neighbors. Use the `unsuppress-map` command to selectively leak more-specific routes to a particular neighbor.

Use the `no` parameter with this command to restore the setting to the default level.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) unsuppress-map WORD

no neighbor (A.B.C.D|X:X::X:X|WORD) unsuppress-map WORD
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`WORD`	Name of the route map used to select routes to unsuppress

Command Mode

Router mode and Address Family mode

Example

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.73 unsuppress-map mymap

#configure terminal
(config)#router bgp 10
(config-router)#address-family ipv4 unicast
(config-router-af)#neighbor 10.10.0.70 unsuppress-map mymap
```

neighbor update-source

This command allows internal BGP sessions to use any operational interface for TCP connections. Use this command in conjunction with any specified interface on the router. The loopback interface is the interface that is most commonly used with this command. The use of loopback interface eliminates a dependency and BGP does not have to rely on the availability of a particular interface for making TCP connections.

Use the `no` parameter with this command to restore the interface assignment to the closest interface.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) update-source WORD

no neighbor (A.B.C.D|X:X::X:X|WORD) update-source
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`WORD`	Loopback interface name

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor 10.10.0.72 update-source myif
```

neighbor version

Use this command to configure ZebOS-XP to accept only a particular BGP version. By default, the system uses BGP version 4 and on request dynamically negotiates down to version 2. Using this command disables the router's version-negotiation capability and forces the router to use only a specified version with the neighbor.

Use the `no` parameter with this command to use the default version level of a neighbor.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) version (4)

no neighbor (A.B.C.D|X:X::X:X|WORD) version
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`4`	BGP version number

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 12
(config-router)#neighbor 10.10.10.10 version 4

(config)#router bgp 12
(config-router)#no neighbor 10.10.10.10 version
```

neighbor weight

Use this command to specify a weight value, per address-family, to all routes learned from a neighbor. The route with the highest weight gets preference when the same prefix is learned from more than one peer. Unlike the local-preference attribute, the weight attribute is relevant only to the local router. The weights assigned using the `set weight` command override the weights assigned using this command.

Use this command in Router mode to specify a weight value for all address families. Use this command in Address Family mode to specify a weight value per IPv4/IPv6/VPNv4/6PE address family,

When the weight is set for a peer group, all members of the peer group get the same weight. This command can also be used to assign a different weight to an individual peer-group member. When an individually-configured weight of a peer-group member is removed, its weight is reset to its peer group's weight.

Use the `no` parameter with this command to remove a weight assignment.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) weight <0-65535>
no neighbor (A.B.C.D|X:X::X:X|WORD) weight
no neighbor (A.B.C.D|X:X::X:X|WORD) weight <0-65535>
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`<0-65535>`	Weight value

Command Mode

Router mode, Address-Family mode

Examples

```
#configure terminal
(config)#router bgp 12
(config-router)#neighbor 10.10.10.10 weight 60

(config-router)#no neighbor 10.10.10.10 weight 60
```

neighbor WORD peer-group

Use this command to create a peer group.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
neighbor WORD peer-group
no neighbor WORD peer-group
```

Parameters

`WORD`	Name of BGP peer group

Command Mode

Router mode

Example

This example shows how to create a peer group named `group1`.

```
#configure terminal
(config)#router bgp 10
(config-router)#neighbor group1 peer-group
```

network

Use this command to specify the networks to be advertised by the BGP routing process. A unicast network address without a mask is accepted if it falls into the natural boundary of its class. A class-boundary mask is derived if the address matches its natural class-boundary.

Use the `backdoor` parameter to specify a backdoor route to a BGP border router that will provide better information about the network. For data to be advertised by BGP, its routing table must include a route to the specified network. This command specifies the networks to be advertised. The network command works if the network being advertised is known to the router.

The `backdoor` parameter enables a route to be the preferred route even if it has a greater distance. A network that is specified as a backdoor network is dynamically assigned an administrative distance of 200 ensuring that IGP learned routes are preferred. If a backdoor network is not sourced by the local router, the network is learned from the external routers. If the route is learned from eBGP for a backdoor network, the distance is set to 20 or 200.

Use the `no` form of this command to remove a network route entry.

Command Syntax

```
network A.B.C.D (backdoor|)
network A.B.C.D/M (backdoor|)
network A.B.C.D mask A.B.C.D (backdoor|)
network A.B.C.D mask A.B.C.D route-map WORD (backdoor|)
network A.B.C.D route-map WORD (backdoor|)
network A.B.C.D/M route-map WORD (backdoor|)
no network A.B.C.D (backdoor|)
no network A.B.C.D/M (backdoor|)
no network A.B.C.D mask A.B.C.D (backdoor|)
no network A.B.C.D mask A.B.C.D route-map WORD (backdoor|)
no network A.B.C.D route-map WORD (backdoor|)
no network A.B.C.D/M route-map WORD (backdoor|)
```

Parameters

`A.B.C.D`	IP prefix <network>, for example, 35.0.0.0
`A.B.C.D/M`	IP prefix <network>/<length>, for example., 35.0.0.0/8
`backdoor`	BGP backdoor route
`routemap`	Route map used to modify the attributes
`WORD`	Name of the route map
`mask`	Network mask, for example, 255.255.0.0
`A.B.C.D`	Network mask, e.g., 255.255.0.0

Command Mode

Router mode and Address-family mode

Examples

The following example illustrates a Class-A address configured as a network route. The natural Class-A network prefix mask length of 8 is internally derived, that is, 2.0.0.0/8.

```
(config)#router bgp 1
(config-router)#network 2.0.0.0
```

network synchronization

Use this command to enable IGP synchronization for BGP static network routes.

Use this no parameter with this command to disable synchronization of BGP static routes.

Command Syntax

```
network synchronization
```

```
no network synchronization
```

Parameters

None

Command Mode

Router mode and Address Family mode

Examples

The following example enables IGP synchronization of BGP static network routes in the router configuration mode.

```
#configure terminal
(config)#router bgp 11
(config-router)#network synchronization
```

The following example enables IGP synchronization of BGP static network routes in the IPv6 unicast address-family mode.

```
#configure terminal
(config)#router bgp 11
(config)#address-family ipv6 unicast
(config-af)#network synchronization
```

redistribute

Use this command to inject routes from one routing process into another. Redistribution is used by routing protocols to advertise routes that are learned by some other means, such as by another routing protocol or by static routes. Since all internal routes are dumped into BGP, careful filtering is applied to make sure that only routes to be advertised reach the internet, not everything. This command allows redistribution by injecting prefixes from one routing protocol into another routing protocol.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
redistribute [connected|isis|kernel|ospf6|rip|static]
redistribute [connected|isis|kernel|ospf6|rip|static] route-map WORD
no redistribute [connected|isis|kernel|ospf6|rip|static]
no redistribute [connected|isis|kernel|ospf6|rip|static] route-map
no redistribute [connected|isis|kernel|ospf6|rip|static] route-map WORD
```

Parameters

`connected`	Redistribute connected routes
`isis`	Redistribute connected ISO IS-IS routes
`kernel`	Redistribute connected kernel routes
`ospf6`	Redistribute OSPFv2 or OSPFv3 routes
`rip`	Redistribute RIP routes
`static`	Redistribute static routes
`route-map`	Route map reference
`WORD`	Route map entries

Command Mode

Router mode and Address Family-vrf mode

Examples

The following example shows the configuration of the route-map name `rmap1` and then the use of this map name in the `redistribute route-map` command.

```
#configure terminal
(config)#route-map rmap1 permit 1
(config-route-map)#match origin incomplete
(config-route-map)#set metric 100
(config-route-map)#exit
(config)#router bgp 12
(config-router)#redistribute ospf route-map rmap1

(config)#router bgp 100
(config-router)#address-family ipv4 vrf VRF_A
(config-router-af)#redistribute static

(config)#router bgp 100
(config-router)#address-family ipv6 vrf VRF_A
```

```
(config-router-af)#redistribute static
```

restart bgp graceful

Use this command to enable a BGP-speaker router for graceful restart. This command stops the whole BGP process and makes ZebOS-XP retain the BGP routes and mark them as stale. Receiving BGP speakers, retain and mark as stale all BGP routes received from the restarting speaker for all address families received in the Graceful Restart Capability exchange.

Command Syntax

```
    restart bgp graceful
```

Parameters

None

Command Mode

Privileged Exec mode

Examples

```
    #restart bgp graceful
```

router bgp

Use this command to start a BGP process.

Use the no parameter with this command to disable an existing routing process.

Command Syntax

```
router bgp <1-65535>
router bgp <1-4294967295>
no router bgp <1-65535>
no router bgp <1-4294967295>
```

Parameters

`<1-65535>`	Associate the routing process with this autonomous system number
`<1-4294967295>`	Associate the routing process with this autonomous system number

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#router bgp 12
(config-router)#
```

router bgp view

Use this command to configure a BGP routing view.

Use the `no` parameter with this command to disable a routing view.

Command Syntax

```
router bgp <1-65535> view WORD

router bgp <1-4294967295> view WORD

no router bgp <1-65535> view WORD

no router bgp <1-4294967295> view WORD
```

Parameters

`<1-65535>`	Autonomous System number
`<1-4294967295>`	Autonomous System number
`WORD`	BGP view name

Command Mode

Configure mode

Examples

```
#configure terminal
(config)#router bgp 12 view 1
(config-router)#
```

snmp restart bgp

Use this command to restart SNMP in Border Gateway Protocol (BGP)

Command Syntax

```
    snmp restart bgp
```

Parameters

None

Command Mode

Configure mode

Examples

```
    #configure terminal
    (config)#snmp restart bgp
```

synchronization

Use this command to enable IGP synchronization of Internal BGP (iBGP) learned routes with the Internal Gateway Protocol (IGP) system in the router configuration mode or in the address-family configuration mode.

Synchronization is used when a BGP router should not advertise routes learned from iBGP neighbors, unless those routes are also present in an IGP (for example, OSPF). Synchronization may be enabled when all the routers in an autonomous system do not speak BGP, and the autonomous system is a transit for other autonomous systems.

The `no synchronization` command is used when BGP router can advertise routes learned from its iBGP neighbors without waiting for the IGP reachability to be present.

Command Syntax

```
synchronization
no synchronization
```

Parameters

None

Default

IGP synchronization is disabled.

Command Mode

Router mode and Address Family modes

Examples

The following example enables IGP synchronization of iBGP routes in Router mode.

```
#configure terminal
(config)#router bgp 11
(config-router)#synchronization
```

The following example enables IGP synchronization of iBGP routes in the IPv6-Unicast address family.

```
#configure terminal
(config)#router bgp 11
(config-router)#address-family ipv6 unicast
(config-af)#synchronization
```

table-map

Use this command to enable or disable suppression/modification of incoming BGP updates to IP RIB/FIB table installation.

In a dedicated route reflector, all the routes it receives may not be required to be stored or only few selected routes need to be stored, as it may not lie in the data path.

Table maps are particularly useful to attain this restriction. Table-map command can be used in two ways:

- When a simple table-map command is given, the route map referenced in the table-map command shall be used to set certain properties (such as the traffic index) of the routes for installation into the RIB. The route is always downloaded, regardless of whether it is permitted or denied by the route map.
- When the option ‘filter’ is given in the table map command, the route map referenced is used to control whether a BGP route is to be downloaded to the IP RIB (hence the filter). A BGP route is not downloaded to the RIB if it is denied by the route map.

Use this command in Router mode to set the table map rule for all address families. Use this command in Address Family mode to set the table map rule per an IPv4 or IPv6 family.

Use the `no` parameter with this command to remove the table-map rule.

Command Syntax

```
table-map WORD (|filter)
```

Parameter

`WORD`	Specify the route-map name to apply.
`filter`	Filer the routes. If present, the incoming routes are pruned as per the rule specified in route-map-name. If not, it is used to alter the incoming packet information.

Default

All BGP routes will be downloaded to IP RIB.

Command Mode

Router mode, Address Family IPv4 mode, and Address Family IPv6 mode

Examples

The following example shows how to set the table-map command without filter for BGP for all address families.

```
#configure terminal
(config)#router bgp 100
(config-router)#table-map abc
```

The following example shows how to set the table-map command with filter for BGP for all address families.

```
#configure terminal
(config)#router bgp 100
(config-router)#table-map abc filter
```

The following example shows how to set the table-map command without filter for BGP for an IPv6 address family.

```
(config)#router bgp 100
(config-router)#address family ipv6
(config-router-af)# table-map abc
```

The following example shows how to set the table-map command with filter for BGP for an IPv6 address family.

```
(config)#router bgp 100
(config-router)#address family ipv6
(config-router-af)# table-map abc filter
```

timers bgp

Use this command to globally set or reset the keepalive and holdtime values for all the neighbors.

Use the `no` parameter with this command to reset timers to default value.

Command Syntax

```
timers bgp <0-65535> <0-65535>
no timers bgp
no timers bgp <0-65535> <0-65535>
```

Parameters

`<0-65535>`	Frequency with which keepalive messages are sent to the neighbors
`<0-65535>`	Interval after which a neighbor is considered dead if keepalive messages are not received

Default

The default keepalive timer value is 30 seconds.

The default holdtime value is 90 seconds.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 10
(config-router)#timers bgp 40 120
```

undebug bgp

Use this command to disable BGP debugging options.

Command Syntax

```
undebug bgp (all|bfd|dampening|events|filters|fsm|keepalives|mpls|nht|nsm|updates)
```

Parameters

`all`	Disable all debugging for BGP
`bfd`	Disable debugging for BGP Bidirectional Forwarding Detection (BFD)
`dampening`	Disable debugging for BGP dampening
`events`	Disable debugging for BGP events
`filters`	Disable debugging for BGP filters
`fsm`	Disable debugging for BGP Finite State Machine (FSM)
`keepalives`	Disable debugging for BGP keepalives
`mpls`	Disable debugging for BGP MPLS
`nht`	Disable debugging for BGP NHT messages
`nsm`	Disable debugging for NSM messages
`updates`	Disable debugging for BGP updates

Command Mode

Privileged Exec mode

Examples

```
#undebug bgp events
```

CHAPTER 3 BGP4+ Commands

This chapter describes the BGP4+ configuration commands.

- address-family ipv6 (see address-family in Chapter 2, *BGP Commands*)
- aggregate-address X:X::X:X/M
- bgp g-shut (see bgp g-shut in n Chapter 2, *BGP Commands*)
- clear bgp * ipv6
- clear bgp ipv6 (A.B.C.D|X:X::X:X)
- clear bgp ipv6 <1-4294967295>
- clear bgp ipv6 external
- clear bgp ipv6 peer-group
- clear bgp ipv6 unicast dampening
- clear bgp ipv6 unicast flap-statistics
- clear ip bpg ipv6 unicast table-map
- clear ipv6 bgp * vrf
- clear ipv6 bgp X:X::X:X vrf
- neighbor activate (see neighbor activate in Chapter 2, *BGP Commands*)
- neighbor attribute-unchanged (see neighbor attribute-unchanged in Chapter 2, *BGP Commands*)
- neighbor capability dynamic (see neighbor capability dynamic in Chapter 2, *BGP Commands*)
- neighbor capability route-refresh (see neighbor capability route-refresh in Chapter 2, *BGP Commands*)
- neighbor default-originate (see neighbor default-originate in Chapter 2, *BGP Commands*)
- neighbor distribute-list (see neighbor distribute-list in Chapter 2, *BGP Commands*)
- neighbor filter-list (see neighbor filter-list in Chapter 2, *BGP Commands*)
- neighbor maximum-prefix (see neighbor maximum-prefix in Chapter 2, *BGP Commands*)
- neighbor next-hop-self (see neighbor next-hop-self in Chapter 2, *BGP Commands*)
- neighbor peer-group (see neighbor peer-group in Chapter 2, *BGP Commands*)
- neighbor prefix-list (see neighbor prefix-list in Chapter 2, *BGP Commands*)
- neighbor remove-private-AS (see neighbor remove-private-AS in Chapter 2, *BGP Commands*)
- neighbor route-map (see neighbor route-map in Chapter 2, *BGP Commands*)
- neighbor route-reflector-client (see neighbor route-reflector-client in Chapter 2, *BGP Commands*)
- neighbor send-community (see neighbor send-community in Chapter 2, *BGP Commands*)
- neighbor soft-reconfiguration inbound (see neighbor soft-reconfiguration inbound in Chapter 2, *BGP Commands*)
- neighbor unsuppress-map (see neighbor unsuppress-map in Chapter 2, *BGP Commands*)
- network X:X::X:X
- redistribute (see redistribute in Chapter 2, *BGP Commands*)

aggregate-address X:X::X:X/M

Use this command to configure BGP aggregate entries.

Aggregates are used to minimize the size of routing tables. Aggregation combines the characteristics of several different routes and advertises a single route. This command creates an aggregate entry in the BGP routing table if any more-specific BGP routes are available in the specified range. Using the `summary-only` parameter advertises the prefix only, suppressing more-specific routes to neighbors.

The `as-set` parameter creates an aggregate entry advertising the path for this route, consisting of all elements contained in all paths being summarized. Use the `as-set` parameter to reduce the size of path information by listing the AS number only once, even if it was included in multiple paths that were aggregated. The `as-set` parameter is useful when aggregation of information results in an incomplete path information.

Use the `no` parameter with this command to disable this function.

Command Syntax

```
aggregate-address X:X::X:X/M
aggregate-address X:X::X:X/M as-set
aggregate-address X:X::X:X/M as-set summary-only
aggregate-address X:X::X:X/M summary-only
aggregate-address X:X::X:X/M summary-only as-set
no aggregate-address X:X::X:X/M
no aggregate-address X:X::X:X/M as-set
no aggregate-address X:X::X:X/M as-set summary-only
no aggregate-address X:X::X:X/M summary-only
no aggregate-address X:X::X:X/M summary-only as-set
```

Parameters

`X:X::X:X/M`	Aggregate IPv6 prefix
`as-set`	Generate AS set path information
`summary-only`	Filter more specific routes from updates

Command Mode

Address Family mode

Default

Disabled

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#neighbor 2.2.2.2 remote-as 100
(config-router)#neighbor 3.3.3.3 remote-as 200
(config-router)#address-family ipv6
(config-router-af)#aggregate-address 3ffe::/32 as-set summary-only
```

clear bgp * ipv6

Use this command to reset the BGP IPv6 connection for all peers.

Command Syntax

```
clear bgp ipv6 *
clear bgp ipv6 * in
clear bgp ipv6 * in prefix-filter
clear bgp ipv6 * out
clear bgp ipv6 * soft
clear bgp ipv6 * soft in
clear bgp ipv6 * soft out
clear ip bgp * ipv6 unicast in
clear ip bgp * ipv6 unicast out
clear ip bgp * ipv6 unicast soft
clear ip bgp * ipv6 unicast soft in
clear ip bgp * ipv6 unicast soft out
```

Parameters

`in`	Clear incoming advertised routes.
`prefix-filter`	Push out prefix-list ORF and do inbound soft reconfig.
`out`	Clear outgoing advertised routes.
`soft`	Clear both incoming and outgoing routes.
`in`	Soft reconfig inbound update.
`out`	Soft reconfig outbound update.
`unicast`	Unicast prefixes
`in`	Clear incoming advertised routes.
`out`	Soft reconfig outbound update
`soft`	Clear both incoming and outgoing routes
`in`	Soft reconfig inbound update
`out`	Clear outgoing advertised routes

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp * ipv6 unicast soft out
```

clear bgp ipv6 (A.B.C.D|X:X::X:X)

Use this command to reset the BGP neighbor addresses (IPv4 or IPv6) for IPv6 peers.

Command Syntax

```
clear bgp ipv6 (A.B.C.D|X:X::X:X)
clear bgp ipv6 (A.B.C.D|X:X::X:X) in
clear bgp ipv6 (A.B.C.D|X:X::X:X) in prefix-filter
clear bgp ipv6 (A.B.C.D|X:X::X:X) out
clear bgp ipv6 (A.B.C.D|X:X::X:X) soft
clear bgp ipv6 (A.B.C.D|X:X::X:X) soft in
clear bgp ipv6 (A.B.C.D|X:X::X:X) soft out
```

Parameters

`in`	Clear incoming advertised routes
`prefix-filter`	Push out prefix-list ORF and do inbound soft reconfig
`out`	Clear outgoing advertised routes
`soft`	Clear both incoming and outgoing routes
`in`	Soft reconfig inbound update
`out`	Soft reconfig outbound update

Command Mode

Privileged Exec mode

Example

```
#clear bgp ipv6 10.5.2.7
```

clear bgp ipv6 <1-4294967295>

Use this command to reset the BGP connection with a specified AS (Autonomous System) number for IPv6 peers.

Command Syntax

```
clear bgp ipv6 <1-4294967295>
clear bgp ipv6 <1-4294967295> in
clear bgp ipv6 <1-4294967295> in prefix-filter
clear bgp ipv6 <1-4294967295> out
clear bgp ipv6 <1-4294967295> soft
clear bgp ipv6 <1-4294967295> soft in
clear bgp ipv6 <1-4294967295> soft out
```

Parameters

`in`	Clear incoming advertised routes.
`prefix-filter`	
	Push out prefix-list ORF and do inbound soft reconfig.
`out`	Clear outgoing advertised routes.
`soft`	Clear both incoming and outgoing routes.
`in`	Soft reconfig inbound update.
`out`	Soft reconfig outbound update.

Command Mode

Privileged Exec mode

Example

```
#clear bgp ipv6 12345
```

clear bgp ipv6 external

Use this command to reset the BGP connection for all external IPv6 peers or for a specified external IPv6 peer.

Command Syntax

```
clear bgp ipv6 external
clear bgp ipv6 external in prefix-filter
clear bgp ipv6 external soft
clear bgp ipv6 external soft in
clear bgp ipv6 external soft out
clear bgp ipv6 external WORD in
clear bgp ipv6 external WORD out
```

Parameters

`in`	Clear incoming advertised routes
`prefix-filter`	Push out prefix-list ORF and do inbound soft reconfig
`soft`	Clear both incoming and outgoing routes
`in`	Soft reconfig inbound update
`out`	Soft reconfig outbound update
`WORD`	Name of external IPv6 peer
`in`	Soft reconfig inbound update
`out`	Soft reconfig outbound update

Command Mode

Privileged Exec mode

Example

```
#clear bgp ipv6 external soft in
```

clear bgp ipv6 peer-group

Use this command to reset the BGP connection for all members of a peer group.

Command Syntax

```
clear bgp ipv6 peer-group WORD
clear bgp ipv6 peer-group WORD in
clear bgp ipv6 peer-group WORD in prefix-filter
clear bgp ipv6 peer-group WORD out
clear bgp ipv6 peer-group WORD soft
clear bgp ipv6 peer-group WORD soft in
clear bgp ipv6 peer-group WORD soft out
```

Parameters

`in`	Clear incoming advertised routes
`prefix-filter`	Push out prefix-list ORF and do inbound soft reconfig
`out`	Clear outgoing advertised routes
`soft`	Clear both incoming and outgoing routes
`in`	Soft reconfig inbound update
`out`	Soft reconfig outbound update
`WORD`	BGP peer-group name

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp peer-group P1
```

clear bgp ipv6 unicast dampening

Use this command to reset IPv6 BGP dampening information.

Command Syntax

```
clear bgp ipv6 unicast dampening
clear bgp ipv6 unicast dampening X:X::X:X
clear bgp ipv6 unicast dampening X:X::X:X/M
clear ip bgp ipv6 unicast dampening
clear ip bgp ipv6 unicast dampening X:X::X:X
clear ip bgp ipv6 unicast dampening X:X::X:X/M
```

Parameters

`unicast`	Unicast prefixes
`X:X::X:X`	IP prefix (network) for example, 2003::
`X:X::X:X/M`	IP prefix (network and length) for example, 2003::/16

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp ipv6 unicast dampening 1:2::3:4/7
```

clear bgp ipv6 unicast flap-statistics

Use this command to reset IPv6 BGP route flap statistics.

Command Syntax

```
clear bgp ipv6 unicast flap-statistics
clear bgp ipv6 unicast flap-statistics X:X::X:X
clear bgp ipv6 unicast flap-statistics X:X::X:X/M
clear ip bgp ipv6 unicast flap-statistics
clear ip bgp ipv6 unicast flap-statistics X:X::X:X
clear ip bgp ipv6 unicast flap-statistics X:X::X:X/M
```

Parameters

`X:X::X:X`	IP prefix (network) for example, 35.0.0.0
`X:X::X:X/M`	IP prefix (network and length) for example, 35.0.0.0/8

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp ipv6 unicast flap-statistics 1:2::3:4/7
```

clear ip bpg ipv6 unicast table-map

Use this command to apply the modified table map or route map rules to the BGP routes in the existing IP routing table.

Command Syntax

```
    clear ip bpg ipv6 unicast table-map (vrf (VRFNAME|all|default))
```

Parameters

`vrf`	Select a VPN Routing/Forwarding Instance.
`VRFNAME`	Specify a VPN Routing/Forwarding instance name.
`all`	Select all VRFs.
`default`	Select default VRFs.

Command Mode

Privileged Exec mode

Examples

```
    #clear ip bgp ipv6 unicast table-map vrf all
```

clear ipv6 bgp * vrf

Use this command to reset an IPv6 BGP connection for all virtual routing forwarding addresses.

Command Syntax

```
clear ipv6 bgp * vrf WORD
clear ipv6 bgp * vrf WORD in
clear ipv6 bgp * vrf WORD out
clear ipv6 bgp * vrf WORD soft
clear ipv6 bgp * vrf WORD soft in
clear ipv6 bgp * vrf WORD soft out
```

Parameters

`in`	Clear incoming advertised routes
`out`	Clear outgoing advertised routes
`soft`	Clear both incoming and outgoing routes
`in`	Soft reconfig inbound update
`out`	Soft reconfig outbound update
`WORD`	BGP peer-group name

Command Mode

Privileged Exec mode

Examples

```
#clear ipv6 bgp * vrf BGPpeer7
```

clear ipv6 bgp X:X::X:X vrf

Use this command to reset the specified VPNv6 Routing/Forwarding (VRF) instance for BGP connections. If the neighbor address is specified with this command, it clears the specified connection. If no address is specified, this command clears all the BGP routes.

Command Syntax

```
clear ipv6 bgp (X:X::X:X) vrf WORD
clear ipv6 bgp X:X::X:X vrf WORD in
clear ipv6 bgp X:X::X:X vrf WORD out
clear ipv6 bgp X:X::X:X vrf WORD soft in
clear ipv6 bgp X:X::X:X vrf WORD soft out
```

Parameters

`vrf`	Specify a VPN routing/forwarding instance for IPv6
`WORD`	VPN routing/forwarding instance name
`in`	Clear incoming advertised routes
`out`	Clear outgoing advertised routes
`soft`	Clear both incoming and outgoing routes
`in`	Soft reconfig inbound update
`out`	Soft reconfig outbound update

Command Mode

Privileged Exec mode

Examples

```
#clear ipv6 bgp vrf 10:10::0:12 vrf vrfname
```

network X:X::X:X

Use this command to specify the networks to be advertised by the BGP routing process. A unicast network address without a mask is accepted if it falls into the natural boundary of its class. A class-boundary mask is derived if the address matches its natural class-boundary.

Use the `backdoor` parameter to specify a backdoor route to a BGP border router that will provide better information about the network. For data to be advertised by BGP, its routing table must include a route to the specified network. This command specifies the networks to be advertised. The network command works if the network being advertised is known to the router.

The `backdoor` parameter enables a route to be the preferred route even if it has a greater distance. A network that is specified as a backdoor network is dynamically assigned an administrative distance of 200 ensuring that IGP learned routes are preferred. If a backdoor network is not sourced by the local router, the network is learned from the external routers. If the route is learned from eBGP for a backdoor network, the distance is set to 20 or 200.

Use the `no` parameter with this command to remove an entry.

Command Syntax

```
network X:X::X:X/M
network X:X::X:X/M backdoor
network X:X::X:X/M route-map WORD (backdoor|)
no network X:X::X:X/M
no network X:X::X:X/M backdoor
no network X:X::X:X/M route-map WORD (backdoor|)
```

Parameters

`X:X::X:X`	IPv6 prefix <network>, for example, 3ffe::
`backdoor`	Specify a BGP backdoor route
`WORD`	Name of the route map

Command Mode

Router mode and Address Family mode

Examples

```
(config)#router bgp 10
(config-router)#network 172.26.0.0/16
```

If Router1 receives updates from 172.10.0.0 via two routing protocols RIP (distance 120) and eBGP (distance 20), router1 chooses the shorter route. Use the `backdoor` parameter to allow Router1 to learn about 172.10.0.0 via RIP.

```
(config)#router rip
(config)#network 172.10.0.0
(config)#router bgp 200
(config)#neighbor 3.3.3.3 remote-as 500
(config)#network 172.10.0.0 backdoor
(config-router)#network 172.16.1.0/24 route-map ipi
```

CHAPTER 4 BGP Virtual Private Network Commands

This chapter describes the BGP Virtual Private Network (VPN) configuration commands.

- address-family (see address-family in Chapter 2, *BGP Commands*)
- bgp inbound-route-filter
- clear ip bgp * vpnv4
- clear ip bgp * vpnv6
- clear ip bgp <1-4294967295> vpnv4
- clear ip bgp <1-4294967295> vpnv6
- clear ip bgp A.B.C.D vpnv4
- clear ip bgp X:X::X:X vpnv6
- debug bgp mpls
- exit-address-family (see exit-address-family in Chapter 2, *BGP Commands*)
- import map
- ip vrf
- neighbor activate (see neighbor activate in Chapter 2, *BGP Commands*)
- neighbor allow-ebgp-vpn
- neighbor allowas-in (see neighbor allowas-in in Chapter 2, *BGP Commands*)
- neighbor as-origination-interval (see neighbor as-origination-interval in Chapter 2, *BGP Commands*)
- neighbor as-override
- neighbor description (see neighbor description in Chapter 2, *BGP Commands*)
- neighbor remote-as (see neighbor remote-as in Chapter 2, *BGP Commands*)
- neighbor send-community (see neighbor send-community in Chapter 2, *BGP Commands*)
- neighbor shutdown (see neighbor shutdown in Chapter 2, *BGP Commands*)
- neighbor send-community
- neighbor soo
- redistribute (see redistribute in Chapter 2, *BGP Commands*)
- rd (route distinguisher)
- route-target

bgp inbound-route-filter

Use this command to enable the MPLS (Multiprotocol Label Switching) VPN/BGP inbound route filter. This command is used to control the installation of routing information into the BGP table.

When a router runs MPLS VPN/BGP PE, it exchanges routing information with a routing distinguisher. By default, ZebOS-XP does not install routing information that does not match the configured routing distinguisher value. When the local box has two VRFs where each routing distinguisher value is 10:100 and 20:200, routing information with routing distinguisher 10:200 is not installed into BGP table.

When `no bgp inbound-route-filter` is configured, all of routing information is installed into the BGP table.

Command Syntax

```
bgp inbound-route-filter
```

```
no bgp inbound-route-filter
```

Parameter

None

Default

Enabled, the router performs the routing distinguisher value check, by default.

Command Mode

Router mode

Examples

```
#configure terminal
(config)#router bgp 100
(config-router)#bgp inbound-route-filter
```

clear ip bgp * vpnv4

Use this command to reset a VPNv4 BGP connection for all peers. This command clears the BGP connection and dynamically resets the outbound routing table. This frees up additional memory required for storing updates to generate new updates.

Command Syntax

```
clear ip bgp * vpnv4 unicast in
clear ip bgp * vpnv4 unicast out
clear ip bgp * vpnv4 unicast soft
clear ip bgp * vpnv4 unicast soft in
clear ip bgp * vpnv4 unicast soft out
```

Parameters

`in`	Clear incoming advertised routes
`out`	Clear outgoing advertised routes
`soft`	Clear both incoming and outgoing routes
`in`	Soft reconfig inbound update
`out`	Soft reconfig outbound update

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp *
#clear ip bgp * vpnv4 unicast out
```

clear ip bgp * vpnv6

Use this command to reset a VPNv6 BGP connection for all peers. This command clears the BGP connection and dynamically resets the outbound routing table. This frees up additional memory required for storing updates to generate new updates.

Command Syntax

```
clear ip bgp * vpnv6 unicast in
clear ip bgp * vpnv6 unicast out
clear ip bgp * vpnv6 unicast soft
clear ip bgp * vpnv6 unicast soft in
clear ip bgp * vpnv6 unicast soft out
```

Parameters

`in`	Clear incoming advertised routes
`out`	Clear outgoing advertised routes
`soft`	Clear both incoming and outgoing routes
`in`	Soft reconfig inbound update
`out`	Soft reconfig outbound update

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp *
#clear ip bgp * vpnv6 unicast out
```

clear ip bgp <1-4294967295> vpnv4

Use this command to reset a BGP connection for all VPN peers in a specified Autonomous System.

Command Syntax

```
clear ip bgp <1-4294967295> vpnv4 unicast in
clear ip bgp <1-4294967295> vpnv4 unicast out
clear ip bgp <1-4294967295> vpnv4 unicast soft
clear ip bgp <1-4294967295> vpnv4 unicast soft in
clear ip bgp <1-4294967295> vpnv4 unicast soft out
```

Parameters

`<1-4294967295>`	Clear peers with this AS number
`in`	Clear incoming advertised routes
`out`	Clear outgoing advertised routes
`soft`	Clear both incoming and outgoing routes
`in`	Soft reconfig inbound update
`out`	Soft reconfig outbound update

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp 500 vpnv4 unicast soft out
```

clear ip bgp <1-4294967295> vpnv6

Use this command to reset a BGP connection for all VPN peers in a specified Autonomous System.

Command Syntax

```
clear ip bgp <1-4294967295> vpnv6 unicast in
clear ip bgp <1-4294967295> vpnv6 unicast out
clear ip bgp <1-4294967295> vpnv6 unicast soft
clear ip bgp <1-4294967295> vpnv6 unicast soft in
clear ip bgp <1-4294967295> vpnv6 unicast soft out
```

Parameters

`<1-4294967295>`	Clear peers with this AS number
`in`	Clear incoming advertised routes
`out`	Clear outgoing advertised routes
`soft`	Clear both incoming and outgoing routes
`in`	Soft reconfig inbound update
`out`	Soft reconfig outbound update

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp 500 vpnv6 unicast soft out
#clear ip bgp 500 vpnv6 unicast out
```

clear ip bgp A.B.C.D vpnv4

Use this command to reset an VPNv4 BGP connection for a specific IPv4 address.

Command Syntax

```
clear ip bgp A.B.C.D vpnv4 unicast in
clear ip bgp A.B.C.D vpnv4 unicast out
clear ip bgp A.B.C.D vpnv4 unicast soft
clear ip bgp A.B.C.D vpnv4 unicast soft in
clear ip bgp A.B.C.D vpnv4 unicast soft out
```

Parameters

`in`	Clear incoming advertised routes
`out`	Clear outgoing advertised routes
`soft`	Clear both incoming and outgoing routes
`in`	Soft reconfig inbound update
`out`	Soft reconfig outbound update

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp 10.10.0.12 soft
#clear ip bgp 10.10.0.10 vpnv4 unicast out
#clear ip bgp 3.3.3.3 vrf VRF1 soft in
```

clear ip bgp X:X::X:X vpnv6

Use this command to reset a VPNv6 BGP connection for a specific IPv6 address.

Command Syntax

```
clear ip bgp X:X::X:X vpnv6 unicast in
clear ip bgp X:X::X:X vpnv6 unicast out
clear ip bgp X:X::X:X vpnv6 unicast soft
clear ip bgp X:X::X:X vpnv6 unicast soft in
clear ip bgp X:X::X:X vpnv6 unicast soft out
```

Parameters

`in`	Clear incoming advertised routes
`out`	Clear outgoing advertised routes
`soft`	Clear both incoming and outgoing routes
`in`	Soft reconfig inbound update
`out`	Soft reconfig outbound update

Command Mode

Privileged Exec mode

Examples

```
#clear ip bgp 3ffe::8 vpnv6 unicast soft in
```

debug bgp mpls

Use this command to enable the display of MPLS related information.

Use the no parameter with this command to disable this function.

Note: This command is available only when vrf option is enabled.

Command Syntax

```
debug bgp mpls
no debug bgp mpls
```

Parameters

None

Default

Disabled

Command Mode

Privileged Exec mode

Examples

```
debug bgp mpls
```

import map

This command assigns a route map to the VRF. This map is applied for routing information imported from another PE or VRF.

Use this command when an application requires finer control over the routes imported into a VRF than provided by the import and export extended communities. You can filter routes that are eligible for import into a VRF through the use of a route map.The route map can deny access to selected routes from a community that is on the import list.

Use the `no` option with this command to remove the map.

Command Syntax

```
import map WORD
```

```
no import map
```

Parameter

`WORD`	Route map

Command Mode

VRF mode

Examples

```
(config)#ip vrf IPI
(config-vrf)#import map set-pref
(config-vrf)#
```

ip vrf

Use this command to assign a VPN Routing Forwarding (VRF) instance.

Use the `no` option with this command to remove the VRF from the instance.

Command Syntax

```
ip vrf WORD
no ip vrf WORD
```

Parameter

`WORD`	Name of the VRF instance

Command Mode

Configure mode

Command Example

```
(config)#ip vrf IPI
(config-vrf)#
```

neighbor allow-ebgp-vpn

Use this command to allow an eBGP neighbor to be a VPN peer. By default, BGP VPN functionality is allowed only for iBGP peers. Using the `neighbor allow-ebgp-vpn` command allows the VPN connection to be established to an eBGP peer.

Use the `no` parameter with this command to remove the configuration.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) allow-ebgp-vpn

no neighbor (A.B.C.D|X:X::X:X|WORD) allow-ebgp-vpn
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in IPv4 format
`X:X::X:X`	Address of the BGP neighbor in IPv6 format
`WORD`	Name of an existing peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Command Mode

Address Family-vpnv4 mode and Address Family-vpnv6 mode

Examples

```
(config)#router bgp 200
(config-router)#neighbor 66.66.66.66 remote-as 100
(config-router)#neighbor 66.66.66.66 update-source lo
(config-router)#address-family vpnv4 unicast
(config-router-af)#neighbor 66.66.66.66 allow-ebgp-vpn
(config-router-af)#neighbor 66.66.66.66 activate
(config-router-af)#exit-address-family
```

neighbor as-override

Use this command to configure a PE router to override the Autonomous System Number (ASN) of a site with the ASN of a provider. BGP normally ignores the routes from the same AS. However, this command is used to override the customer‘s ASN in BGP, so that the customer CE accepts and installs routes from the same AS.

Typically, this command is used when Customer Edge (CE) routers have the same ASN in some or all sites. As per BGP requirement, a BGP speaker rejects a route that has the same ASN as itself, in the AS_PATH attribute. Thus the CE routers having the same ASN do not accept routes from each other. Configuring the neighbor as-override command on the PE router removes the CE neighbor's ASN from the AS_PATH attribute allowing CE routers with the same ASN to accept routes from each other.

Use the no parameter with this command to remove VPN IPv4 or VPN IPv6 prefixes from a specified router.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) as-override
no neighbor (A.B.C.D|X:X::X:X|WORD) as-override
```

Parameters

A.B.C.D	Address of the BGP neighbor in IPv4 format
X:X::X:X	Address of the BGP neighbor in IPv6 format
WORD	Name of an existing peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

Default

Disabled

Command Mode

Address Family-vrf mode

Examples

```
#configure terminal
(config)#router bgp 7657
(config-router)#address-family ipv4 vrf VRF_A
(config-router-af)#neighbor 10.10.0.1 as-override

#configure terminal
(config)#router bgp 7657
(config-router)#address-family ipv6 vrf VRF_A
(config-router-af)#neighbor 3ffe:15:15:15:15::0 as-override
```

neighbor send-community

Use this command to send the extended-community attribute to a customer edge router. In VPN, route-distinguisher and route-target are encoded in BGP extended-community. This command enables sending of BGP routes with extended community to a neighbor.

See also neighbor send-community in Chapter 2, *BGP Commands*.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) send-community
neighbor (A.B.C.D|X:X::X:X|WORD) send-community (both|extended|standard)
no neighbor (A.B.C.D|X:X::X:X|WORD) send-community
no neighbor (A.B.C.D|X:X::X:X|WORD) send-community (both|extended|standard)
```

Parameters

`A.B.C.D`	Address of the BGP neighbor in an IPv4 format
`X:X::X:X`	Address of the BGP neighbor in an IPv6 format
`WORD`	Name of the BGP peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`both`	Send Standard and Extended Community attributes
`extended`	Send Extended Community attributes
`standard`	Send Standard Community attributes

Default

Disabled. No extended-community attribute is sent to a customer router.

Command Mode

Router mode and Address Family mode

Examples

```
(config)#router bgp 100
(config-router)#address-family ipv4 vrf VRF_A
(config-router-af)#neighbor 10.10.0.1 send-community extended

(config)#router bgp 100
(config-router)#address-family ipv6 vrf VRF_A
(config-router-af)#neighbor 3ffe:15:15:15:15::0 send-community extended
```

neighbor soo

Use this command to enable the site-of-origin (SOO) feature. If the customer AS is multi-homed to the ISP, this command ensures that the PE does not advertise the routes back to the same AS.

Use the `no` parameter with this command to disable this feature.

Command Syntax

```
neighbor (A.B.C.D|X:X::X:X|WORD) soo AS:nn_or_IP:nn
no neighbor (A.B.C.D|X:X::X:X|WORD) soo
```

Parameters

`A.B.C.D` Address of the BGP neighbor in IPv4 format

`X:X::X:X` Address of the BGP neighbor in IPv6 format

`WORD` Name of an existing peer group

Note: For information about how to create peer groups, refer to the neighbor peer-group and neighbor remote-as commands. When this parameter is used with a command, the command applies to all peers in the group.

`ASN:nn_or_IP-address:nn`

Specify an AS number and an arbitrary number (for example, 100:1). Otherwise, specify a 32-bit IP address and an arbitrary number (for example, 192.16.10.1:1).

Command Mode

Address Family VRF mode

Examples

```
(config)#router bgp 100
(config-router)#address-family ipv4 vrf VRF_A
(config-router-af)#neighbor 10.10.0.1 soo 100:1
```

rd (route distinguisher)

Use this command to assign a route distinguisher (RD) for the VRF. The route distinguisher value must be a unique value on the router.

This command creates routing and forwarding tables and specifies the default RD for a VPN. The RD is added to the customer's IPv4 prefixes, changing them into globally unique VPN-IPv4 prefixes.

Command Syntax

```
rd ASN:nn_or_IP-address:nn
```

Parameters

`ASN:nn_or_IP-address:nn`

AS number and an arbitrary number (for example, 100:1). Otherwise, specify a 32-bit IP address and an arbitrary number (for example, 192.16.10.1:1).

Command Mode

VRF mode

Examples

```
(config)#ip vrf VRF_A
(config-vrf)#rd 100:1
```

route-target

Use this command to add a list of import and export route-target extended communities to the VRF.

This command creates lists of import and export route-target extended communities for the VRF. It specifies a target VPN extended community. Execute the command once for each community. All routes with the specific route-target extended community are imported into all VRFs with the same extended community as an import route-target.

Use the `no` parameter with this command to delete a route target.

Command Syntax

```
route-target (import|export|both) ASN:nn_or_IP-address:nn
no route-target (import|export|both) ASN:nn_or_IP-address:nn
```

Parameters

`import`	Import routing information
`export`	Export routing information
`both`	Import and export routing information
`ASN:nn_or_IP-address:nn`	AS number and an arbitrary number (for example, 100:1). Otherwise, specify a 32-bit IP address and an arbitrary number (for example, 192.16.10.1:1).

Command Mode

VRF mode

Examples

```
(config)#ip vrf VRF_A
(config-vrf)#route-target both 100:10

(config)#ip vrf VRF_A
(config-vrf)#route-target import 100:20
```

CHAPTER 5 BGP Show Commands

This chapter describes the BGP show commands.

- show ip bgp vpnv4
- show ip bgp vpnv6
- show ip extcommunity-list
- show ip protocols
- show ip vrf

show bgp

Use this command to display the status of BGP routes.

Command Syntax

```
show bgp
show bgp (ipv6)
show bgp (ipv4|ipv6) (unicast|multicast)
show ip bgp
show ip bgp ipv4 (unicast|multicast)
```

Parameters

`ipv4`	IPv4 routes
`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Examples

```
#show bgp
```

show bgp A.B.C.D

Use this command to display BGP route information for a network.

Command Syntax

```
show bgp (ipv4) (unicast|multicast) A.B.C.D

show ip bgp A.B.C.D

show ip bgp ipv4 (unicast|multicast) A.B.C.D
```

Parameters

`ipv4`	IPv4 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes
`A.B.C.D`	IP prefix (network), for example, 35.0.0.0

Command Mode

Privileged Exec and Exec mode

Examples

```
#show ip bgp 192.10.23.67
BGP table version is 7, local router ID is 80.80.80.80
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
              S Stale
Origin codes: i - IGP, e - EGP,? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
S>i10.70.0.0/24     192.10.23.67             0    100      0 ?
S>i30.30.30.30/32   192.10.23.67             0    100      0 ?
S>i63.63.63.1/32    192.10.23.67             0    100      0 ?
S>i67.67.67.67/32   192.10.23.67             0    100      0 ?
S>i172.22.10.0/24   192.10.23.67             0    100      0 ?
S>i192.10.21.0      192.10.23.67             0    100      0 ?
S>i192.10.23.0      192.10.23.67             0    100      0 ?
Total number of prefixes 7
```

show bgp A.B.C.D/M

Use this command to display BGP route information for a network prefix.

Command Syntax

```
show bgp ipv4 (unicast|multicast) A.B.C.D/M
show ip bgp A.B.C.D/M
show ip bgp ipv4 (unicast|multicast) A.B.C.D/M
```

Parameters

`ipv4`	IPv4 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes
`A.B.C.D/M`	IP prefix (network/length), for example, 35.0.0.0/8

Command Mode

Privileged Exec and Exec mode

Examples

```
#show bgp ipv4 unicast 35.0.0.1/8
```

show bgp community

Use this command to display BGP routes that match a community.

Command Syntax

```
show bgp community

show bgp community [AA:NN|local-AS|no-advertise|no-export] (exact-match|)

show bgp (ipv4|ipv6) (unicast|multicast) community [AA:NN|local-AS|no-advertise|no-
  export] (exact-match|)

show bgp (ipv6) community [AA:NN|local-AS|no-advertise|no-export] (exact-match|)

show ip bgp community

show ip bgp community [AA:NN|local-AS|no-advertise|no-export] (exact-match|)

show ip bgp ipv4 (unicast|multicast) community [AA:NN|local-AS|no-advertise|no-
  export] (exact-match|)
```

Parameters

`ipv4`	IPv4 routes
`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes
`AA:NN`	Community number
`local-AS`	Do not send outside local AS (well-known community)
`no-advertise`	Do not advertise to any peer (well-known community)
`no-export`	Do not export to next AS (well-known community)
`exact-match`	Exact match of the communities

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show bgp community local-as no-export
#show bgp community local-AS exact-match
#show ip bgp ipv4 multicast community 12:34 exact-match
```

show bgp community-list

Use this command to display BGP routes that match a community list.

Command Syntax

```
show bgp community-list WORD (exact-match|)
show bgp (ipv4|ipv6) (unicast|multicast) community-list WORD (exact-match|)
show bgp (ipv6) community-list WORD (exact-match|)
show ip bgp community-list WORD (exact-match|)
show ip bgp ipv4 (unicast|multicast) community-list WORD (exact-match|)
```

Parameters

`WORD`	Community list name
`ipv4`	IPv4 routes
`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes
`exact-match`	Only routes that exactly match the community

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show ip bgp community-list mylist exact-match
#show ip bgp ipv4 multicast community-list mylist exact-match
```

show bgp dampening dampened-paths

Use this command to display detailed information about paths suppressed due to dampening.

Command Syntax

```
show bgp dampening dampened-paths
show bgp (ipv4|ipv6) (unicast|multicast) dampening dampened-paths
show bgp (ipv6) dampening dampened-paths
show ip bgp dampening dampened-paths
show ip bgp ipv4 (unicast|multicast) dampening dampened-paths
```

Parameters

`ipv4`	IPv4 routes
`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show bgp dampening dampened-paths
```

show bgp dampening flap-statistics

Use this command to display BGP dampening flap statistics.

Command Syntax

```
show bgp dampening flap-statistics

show bgp (ipv4|ipv6) (unicast|multicast) dampening flap-statistics

show bgp (ipv6) dampening flap-statistics

show ip bgp dampening flap-statistics

show ip bgp ipv4 (unicast|multicast) dampening flap-statistics
```

Parameters

`ipv4`	IPv4 routes
`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Examples

This sample output shows that the internal route (`i`), has flapped 3 times and is now categorized as history (`h`).

```
#show ip bgp dampening flap-statistics
BGP table version is 1, local router ID is 30.30.30.77
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network            From                 Flaps  Duration  Reuse   Path
hi1.1.1.0/24     10.100.0.62          3  00:01:20         i
```

show bgp dampening parameters

Use this command to display the BGP dampening parameters.

Command Syntax

```
show bgp dampening parameters
show bgp (ipv4|ipv6) (unicast|multicast) dampening parameters
show bgp (ipv6) dampening parameters
show ip bgp dampening parameters
show ip bgp ipv4 (unicast|multicast) dampening parameters
```

Parameters

`ipv4`	IPv4 routes
`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Examples

```
#show ip bgp dampening parameters
```

show bgp filter-list

Use this command to display routes that match a regular expression filter list.

Command Syntax

```
show bgp filter-list WORD
show bgp (ipv4|ipv6) (unicast|multicast) filter-list WORD
show bgp (ipv6) filter-list WORD
show ip bgp filter-list WORD
show ip bgp ipv4 (unicast|multicast) filter-list WORD
```

Parameters

`WORD`	Regular-expression filter list
`ipv4`	IPv4 routes
`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show ip bgp filter-list mylist
```

show bgp inconsistent-as

Use this command to display routes with inconsistent AS paths.

Command Syntax

```
show bgp inconsistent-as
show bgp (ipv4|ipv6) (unicast|multicast) inconsistent-as
show bgp (ipv6) inconsistent-as
show ip bgp inconsistent-as
show ip bgp ipv4 inconsistent-as
show ip bgp ipv4 (unicast|multicast) inconsistent-as
```

Parameters

`ipv4`	IPv4 routes
`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Examples

```
#show bgp inconsistent-as
```

show bgp ipv6

Use this command to display the status of IPv6 BGP connections.

Command Syntax

```
show bgp (ipv6) (unicast|multicast|labeled|)
show bgp (ipv6) (unicast|multicast|labeled|) X:X::X:X/M
```

Parameters

`X:X::X:X/M`	IPv6 prefix <network>/<length>, for example, 3ffe:a::/64
`multicast`	IPv6 multicast address prefixes
`unicast`	IPv6 unicast address prefixes
`labeled`	Labeled IPv6 routes

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show bgp ipv6 labeled 3ffe:a::/64
           BGP routing table entry for 3ffe:a::/64
           Paths: (1 available, best #1, table Default-IP-Routing-Table)
           Not advertised to any peer
           Local
           ::ffff:114:1414 from 20.20.20.1 (92.92.92.92)
        Origin incomplete metric 0, localpref 100, label   5420,
           valid, internal, best
           Last update: Mon May 26 17:48:18 2008
```

show bgp neighbors

Use this command to display information about TCP and BGP neighbor connections.

Command Syntax

```
show bgp neighbors
show bgp (ipv4|ipv6) (unicast|multicast|) neighbors
show bgp (ipv6) neighbors
show ip bgp ipv4 (unicast|multicast) neighbors
show ip bgp neighbors
show ip bgp neighbors (A.B.C.D|X:X::X:X)
show ip bgp ipv4 (unicast|multicast) neighbors (A.B.C.D|X:X::X:X)
```

Parameters

`ipv4`	IPv4 neighbor connection information
`ipv6`	IPv6 neighbor connection information
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes
`A.B.C.D`	IPv4 neighbor
`X:X::X:X`	IPv6 neighbor

Command Mode

Privileged Exec and Exec modes

Examples

```
#show ip bgp neighbors
```

show bgp neighbors advertised-routes

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

Command Syntax

```
show bgp neighbors (A.B.C.D|X:X::X:X) advertised-routes
show bgp (ipv4|ipv6) (unicast|multicast|) neighbors (A.B.C.D|X:X::X:X) advertised-
  routes
show ip bgp neighbors (A.B.C.D|X:X::X:X) advertised-routes
show ip bgp ipv4 (unicast|multicast) neighbors (A.B.C.D|X:X::X:X) advertised-routes
```

Parameters

`A.B.C.D`	IPv4 neighbor
`X:X::X:X`	IPv6 neighbor
`ipv4`	IPv4 addresses
`ipv6`	IPv6 addresses
`multicast`	Multicast prefixes
`unicast`	Unicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show ip bgp ipv4 multicast neighbors 1.2.3.4 advertised-routes
```

show bgp neighbors received prefix-filter

Use this command to display the prefix list filter.

Command Syntax

```
show bgp neighbors (A.B.C.D|X:X::X:X) received prefix-filter

show bgp (ipv4|ipv6) (unicast|multicast|) neighbors (A.B.C.D|X:X::X:X) received
  prefix-filter

show ip bgp neighbors (A.B.C.D|X:X::X:X) received prefix-filter

show ip bgp ipv4 (unicast|multicast) neighbors (A.B.C.D|X:X::X:X) received prefix-
  filter
```

Parameters

`A.B.C.D`	IPv4 address
`X:X::X:X`	IPv6 address
`ipv4`	IPv4 addresses
`ipv6`	IPv6 addresses
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show bgp neighbors received prefix-filter
```

show bgp neighbors received-routes

Use this command to display the received routes from a neighbor.

To display all the received routes from a neighbor, perform a BGP soft reconfigure first.

Command Syntax

```
show bgp neighbors (A.B.C.D|X:X::X:X) received-routes

show bgp (ipv4|ipv6) (unicast|multicast|) neighbors (A.B.C.D|X:X::X:X) received-
  routes

show ip bgp ipv4 (unicast|multicast) neighbors (A.B.C.D|X:X::X:X) received-routes

show ip bgp neighbors (A.B.C.D|X:X::X:X) received-routes
```

Parameters

`A.B.C.D`	IPv4 address
`X:X::X:X`	IPv6 address
`ipv4`	IPv4 addresses
`ipv6`	IPv6 addresses
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show bgp neighbors 10.10.10.2 received-routes
```

show bgp neighbors routes

Use this command to display all accepted routes learned from neighbors.

Command Syntax

```
show bgp neighbors (A.B.C.D|X:X::X:X) routes
show bgp (ipv4|ipv6) (unicast|multicast|) neighbors (A.B.C.D|X:X::X:X) routes
show ip bgp neighbors (A.B.C.D|X:X::X:X) routes
show ip bgp ipv4 (unicast|multicast) neighbors (A.B.C.D|X:X::X:X) routes
```

Parameters

`A.B.C.D`	IPv4 address
`X:X::X:X`	IPv6 address
`ipv4`	IPv4 addresses
`ipv6`	IPv6 addresses
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Example

The following output displays detailed information about the neighbor.

```
#show bgp neighbors 10.10.10.2 routes
BGP neighbor is fe80::203:47ff:feb0:d72b, remote AS 10, local AS 10, internal
link
  BGP version 4, remote router ID 10.10.10.50
  BGP state = Established, up for 00:02:01
  Last read 00:00:01, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received (old and new)
    Address family IPv4 Unicast: advertised and received
    Address family IPv6 Unicast: advertised and received
  Received 3 messages, 0 notifications, 0 in queue
  Sent 5 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Minimum time between advertisement runs is 5 seconds
 For address family: IPv4 Unicast
  Community attribute sent to this neighbor (both)
  0 accepted prefixes
  0 announced prefixes
```

show bgp nexthop-tracking

Use this command to display BGP nexthop-tracking status.

Command Syntax

```
show bgp nexthop-tracking
```

Parameters

None

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show bgp nexthop-tracking
Configured NHT: ENABLED
NHT Delay time-interval : 6
BGP VRF: (Default) VRF_ID 0
BGP Instance: (Default), AS: 100, router-id 4.4.4.40
NHT is Enabled
Recvd Msg count from NSM: 0
NHT delay-timer remaining seconds: 0
BGP nexthop(s):
Total number of IPV4 nexthops : 0
Total number of IPV6 nexthops : 0

BGP VRF: VRF_A VRF_ID 2
BGP Instance: (Default), AS: 100, router-id 4.4.4.40
NHT is Enabled
Recvd Msg count from NSM: 0
NHT delay-timer remaining seconds: 0
BGP nexthop(s):
Total number of IPV4 nexthops : 0
Total number of IPV6 nexthops : 0
```

show bgp nexthop-tree-details

Use this command to display BGP nexthop-tree details.

Command Syntax

```
show bgp nexthop-tree-details
```

Parameters

None

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show bgp nexthop-tree-details
Configured NHT: ENABLED
NHT Delay time-interval : 6
BGP VRF: (Default) VRF_ID 0
BGP Instance: (Default), AS: 100, router-id 4.4.4.40
NHT is Enabled
Recvd Msg count from NSM: 0
NHT delay-timer remaining seconds: 0
BGP nexthop(s):
Total number of IPV4 nexthops : 0
Total number of IPV6 nexthops : 0

BGP VRF: VRF_A VRF_ID 2
BGP Instance: (Default), AS: 100, router-id 4.4.4.40
NHT is Enabled
Recvd Msg count from NSM: 0
NHT delay-timer remaining seconds: 0
BGP nexthop(s):
Total number of IPV4 nexthops : 0
Total number of IPV6 nexthops : 0

TSUP40#
TSUP40#show bgp nexthop-tree-details
BGP Instance: (Default), AS: 100, router-id 4.4.4.40
AFI_IP Nexthop count : 0
AFI_IP6 Nexthop count : 0

BGP Instance: (Default), AS: 0, router-id 0.0.0.0
AFI_IP Nexthop count : 0
AFI_IP6 Nexthop count : 0

BGP Instance: (Default), AS: 100, router-id 4.4.4.40
```

show bgp paths

Use this command to display BGP path information.

Command Syntax

```
show bgp paths

show bgp (ipv4|ipv6) (unicast|multicast|) paths

show bgp (ipv6) paths

show ip bgp paths

show ip bgp ipv4 (unicast|multicast) paths
```

Parameters

`ipv4`	IPv4 routes
`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show ip bgp ipv4 unicast paths

Address       Refcnt Path
[0x81fa578:0] (239)
```

show bgp prefix-list

Use this command to display routes matching the prefix-list.

Command Syntax

```
show bgp prefix-list WORD
show bgp (ipv4|ipv6) (unicast|multicast|) prefix-list WORD
show ip bgp prefix-list WORD
show ip bgp ipv4 (unicast|multicast) prefix-list WORD
```

Parameters

`WORD`	Name of the IP prefix list
`ipv4`	IPv4 routes
`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show bgp prefix-list mylist
```

show bgp quote-regexp

Use this command to display route matching an AS path quoted regular expression.

Command Syntax

```
show bgp quote-regexp WORD
show bgp (ipv4|ipv6) (unicast|multicast|) quote-regexp WORD
show ip bgp quote-regexp WORD
show ip bgp ipv4 (unicast|multicast) quote-regexp WORD
```

Parameters

`WORD`	A regular expression to match the AS paths. Use quotes to enclose the regular expression.
`ipv4`	IPv4 route information
`ipv6`	IPv6 route information
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show bgp quote-regexp "IPI"
```

show bgp regexp

Use this command to display routes matching the AS path regular expression.

Command Syntax

```
show bgp regexp LINE
show bgp (ipv4|ipv6) (unicast|multicast) regexp LINE
show bgp (ipv6) regexp LINE
show ip bgp regexp LINE
show ip bgp ipv4 (unicast|multicast) regexp LINE
```

Parameters

`ipv4`	IPv4 routes
`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes
`LINE`	A regular expression to match the AS paths

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show bgp regexp IPI
```

show bgp route-map

Use this command to display routes that match the specified route map.

Command Syntax

```
show bgp route-map WORD
show bgp (ipv4|ipv6) (unicast|multicast) route-map WORD
show bgp (ipv6) route-map WORD
show ip bgp route-map WORD
show ip bgp ipv4 (unicast|multicast) route-map WORD
```

Parameters

`WORD`	Routes matching the route-map
`ipv4`	IPv4 routes
`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show bgp route-map IPI
```

show bgp summary

Use this command to display a summary of BGP neighbor status.

Command Syntax

```
show bgp summary

show bgp (ipv4|ipv6) (unicast|multicast|) summary

show ip bgp summary

show ip bgp ipv4 (unicast|multicast) summary
```

Parameters

`ipv4`	IPv4 routes
`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Example

This is a sample output from the `show ip bgp summary` command displaying a summary of BGP neighbor status.

```
#show ip bgp summary
BGP router identifier 10.10.15.50, local AS number 65000
1 BGP AS-PATH entries
0 BGP community entries

Neighbor V    AS MsgRcvd MsgSent   TblVer InQ OutQ Up/Down State/PfxRcd
10.10.9.50    4 65000     460      595    0    0              00:17:48
3
10.10.14.51   4   100      93      120    0       0    0      00:42:16
0

Total number of neighbors 2
```

show bgp view

Use this command to display information for a BGP view.

Command Syntax

```
show bgp ipv6 view WORD

show ip bgp view WORD

show ip bgp view WORD A.B.C.D

show ip bgp view WORD A.B.C.D/M

show ip bgp view WORD ipv4 (unicast|multicast) summary

show ip bgp view WORD neighbors

show ip bgp view WORD neighbors (A.B.C.D|X:X::X:X)

show ip bgp view WORD summary
```

Parameters

`ipv6`	IPv6 addresses
`WORD`	BGP view name
`A.B.C.D`	Network in the BGP routing table
`A.B.C.D/M`	IP prefix <network>/<length>, e.g., 35.0.0.0/8, in the BGP routing table
`ipv4`	IPv4 addresses
`multicast`	Multicast prefixes
`unicast`	Unicast prefixes
`summary`	Summary of BGP neighbor status
`neighbors`	Detailed information on TCP and BGP neighbor connections
`A.B.C.D`	IPv4 neighbor
`X:X::X:X`	IPv6 neighbor
`summary`	Summary of BGP neighbor status

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show ip bgp view I2
BGP table version is 0, local router ID is 10.10.10.50
Status codes: s suppressed, d damped, h history, p stale, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*>i100.156.70.0/24  10.10.10.52                     0      0 i
*>i100.156.71.0/24  10.10.10.52                     0      0 i
*>i100.156.72.0/24  10.10.10.52                     0      0 i
*>i100.156.73.0/24  10.10.10.52                     0      0 i
*>i100.156.74.0/24  10.10.10.52                     0      0 i
```

```
Total number of prefixes 5
```

show bgp X:X::X:X

Use this command to display BGP network information in an IPv6 environment.

Command Syntax

```
show bgp X:X::X:X
show bgp (ipv6) X:X::X:X
show bgp (ipv6) (unicast|multicast) X:X::X:X
```

Parameters

`ipv6`	IPv6 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes
`X:X::X:X`	IPv6 prefix (network), for example, 2003::

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show bgp ipv6 3ffe::8
```

show bgp X:X::X:X/M longer prefixes

Use this command to display BGP network information along with mask information.

Command Syntax

```
show bgp X:X::X:X/M longer-prefixes
```

Parameters

`X:X::X:X/M`	IPv6 prefix (network/length), for example, 2003::/16

Command Mode

Privileged Exec mode and Exec mode

Examples

```
#show bgp 3ffe::8/8 longer-prefixes
```

show debugging bgp

Use this command to display BGP debugging options.

Command Syntax

```
show debugging bgp
```

Parameters

None

Command Mode

Privileged Exec mode and Exec mode

Example

This is a sample output from the show debugging bgp command.

```
#show debugging bgp
BGP debugging status:
 BGP debugging is on
 BGP events debugging is on
 BGP updates debugging is on
 BGP fsm debugging is on
```

show ip bgp

Use this command to display BGP process information.

Command Syntax

```
show ip bgp
show ip bgp ipv4 (unicast|multicast)
```

Parameters

`ipv4`	IPv4 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Examples

```
#show ip bgp
```

show ip bgp attribute-info

Use this command to show internal attribute hash information.

Command Syntax

```
    show ip bgp attribute-info
```

Parameters

None

Command Mode

Privileged Exec mode and Exec mode

Example

This is a sample output from the `show ip bgp attribute-info` command displaying internal attribute information.

```
    #show ip bgp attribute-info
    attr[1] nexthop 0.0.0.0
    attr[1] nexthop 10.10.10.10
```

show ip bgp cidr-only

Use this command to display routes with non-natural network masks.

Command Syntax

```
show ip bgp cidr-only
show ip bgp ipv4 (unicast|multicast) cidr-only
```

Parameters

`ipv4`	IPv4 routes
`unicast`	Unicast prefixes
`multicast`	Multicast prefixes

Command Mode

Privileged Exec mode and Exec mode

Example

This is a sample output from the `show ip bgp cidr-only` command.

```
#show ip bgp cidr-only
BGP table version is 0, local router ID is 10.10.10.50
Status codes: s suppressed, d damped, h history, p stale, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 3.3.3.0/24       10.10.10.10                            0 11 i
Total number of prefixes 2
```

show ip bgp community-info

Use this command to list all BGP community information.

Command Syntax

```
show ip bgp community-info
```

Parameters

None

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show ip bgp community-info

Address Refcnt Community
```

show ip bgp scan

Use this command to display BGP scan status.

Command Syntax

```
show ip bgp scan
```

Parameters

None

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show ip bgp scan
BGP scan is running
BGP scan interval is 60
BGP instance: AS is 11,DEFAULT
Current BGP nexthop cache:
BGP connected route:
 10.10.10.0/24
 10.10.11.0/24
```

show ip bgp vpnv4

Use this command to display information relating to VPNv4.

Command Syntax

```
show ip bgp vpnv4 all
show ip bgp vpnv4 all A.B.C.D
show ip bgp vpnv4 all neighbors
show ip bgp vpnv4 all neighbors A.B.C.D
show ip bgp vpnv4 all summary
show ip bgp vpnv4 all tags
show ip bgp vpnv4 rd WORD
show ip bgp vpnv4 rd WORD A.B.C.D
show ip bgp vpnv4 rd WORD label
show ip bgp vpnv4 rd WORD neighbors
show ip bgp vpnv4 rd WORD neighbors A.B.C.D
show ip bgp vpnv4 rd WORD summary
show ip bgp vpnv4 view WORD all
show ip bgp vpnv4 vrf NAME
show ip bgp vpnv4 vrf NAME A.B.C.D
show ip bgp vpnv4 vrf NAME label
show ip bgp vpnv4 vrf NAME summary
```

Parameters

`all`	Displays information about all VPNv4 NLRIs
`A.B.C.D`	Network
`neighbors`	TCP and BGP neighbor connections
`A.B.C.D`	Network
`summary`	Summary display
`tags`	BGP tags for prefixes
`rd`	Route distinguisher
`WORD`	BGP view name
`A.B.C.D`	Network
`label`	MPLS Labels for prefixes
`neighbors`	TCP and BGP neighbor connections
`A.B.C.D`	Network
`summary`	Summary display
`view`	VPNv4 NLRI-specific information
`WORD`	BGP view name

`vrf`	VRF VPNv4 NLRIs
`NAME`	VPN Routing/Forwarding instance name
`A.B.C.D`	Network
`label`	MPLS Labels for prefixes
`summary`	Summary display

Command Mode

Privileged Exec mode and Exec mode

Example

This is a sample output from the `show ip bgp vpnv4 all` command displaying VPNv4 specific information

```
#show ip bgp vpnv4 all
   Network           Next Hop            Metric LocPrf Weight Path
Route Distinguisher: 100:1 (VRF1)
* i 10.10.9.0/24     10.10.0.1               0   141      0  65000 ?
*>  10.10.9.0/24     10.10.14.50             0            0  65000 ?
*>  10.10.10.0/24    10.10.14.50                          0  65000 ?
* i 10.10.15.0/24    10.10.0.1                   141      0  65000 ?
*>  10.10.15.0/24    10.10.14.50             0            0  65000 ?
```

show ip bgp vpnv6

Use this command to display information relating to VPNv6 Network Layer Reachability Information (NLRI).

Command Syntax

```
show ip bgp vpnv6 all
show ip bgp vpnv6 all A:B::C:D
show ip bgp vpnv6 all neighbors
show ip bgp vpnv6 all neighbors (A.B.C.D|X:X::X:X)
show ip bgp vpnv6 all summary
show ip bgp vpnv6 all tags
show ip bgp vpnv6 rd WORD
show ip bgp vpnv6 rd WORD label
show ip bgp vpnv6 rd WORD neighbors
show ip bgp vpnv6 rd WORD neighbors (A.B.C.D|X:X::X:X)
show ip bgp vpnv6 rd WORD summary
show ip bgp vpnv6 rd WORD X:X::X:X
show ip bgp vpnv6 view WORD all
show ip bgp vpnv6 vrf NAME
show ip bgp vpnv6 vrf NAME label
show ip bgp vpnv6 vrf NAME summary
show ip bgp vpnv6 vrf NAME X:X::X:X
```

Parameters

`all`	Displays information about all VPNv4 NLRIs
`A:B::C:D`	IPv4 network
`neighbors`	TCP and BGP neighbor connections
`A:B::C:D`	IPv4 network
`X:X::X:X`	IPv6 network
`summary`	Summary display
`tags`	BGP tags for prefixes
`rd`	Route distinguisher
`WORD`	Route distinguisher name
`label`	MPLS labels for prefixes
`neighbors`	TCP and BGP neighbor connections
`A:B::C:D`	IPv4 network
`X:X::X:X`	IPv6 network
`summary`	Summary display
`X:X::X:X`	IPv6 network

`view`	VPNv4 NLRI-specific information
`WORD`	View name
`all`	All VPNv4 NLRIs
`vrf`	VRF VPNv4 NLRIs
`NAME`	VPN Routing/Forwarding instance name
`label`	MPLS labels for prefixes
`summary`	Summary display
`X:X::X:X`	IPv6 network

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show ip bgp vpnv6 all
```

show ip extcommunity-list

Use this command to display BGP routes that match an extended community list.

Command Syntax

```
show ip extcommunity-list
show ip extcommunity-list (<1-199>|WORD)
```

Parameters

<1-199>	Number of extended community list
WORD	Name of extended community list

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show ip extcommunity-list 33
```

show ip protocols

Use this command to display information about the IP protocols such as IP routing process parameters and statistics.

Command Syntax

```
show ip protocols
show ip protocols bgp
```

Parameters

`bgp`	BGP information

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show ip protocols bgp
Routing Protocol is "bgp 100"
Sending updates every 30 seconds with +/-50%, next due in 12 seconds
Timeout after 180 seconds, garbage collect after 120 seconds
Outgoing update filter list for all interface is not set
Incoming update filter list for all interface is not set
Default redistribution metric is 1
Redistributing: connected static
Default version control: send version 2, receive version 2
Interface        Send  Recv   Key-chain
   eth0                  2     2
Routing for Networks:
  10.10.0.0/24
Routing Information Sources:
  Gateway          BadPackets BadRoutes  Distance Last Update
 Distance: (default is 120
```

show ip vrf

Use this command to display the routing information of the VRF, such as interface, route distinguisher, route-target, and so on.

Command Syntax

```
show ip vrf
show ip vrf WORD
```

Parameter

`WORD`	VRF name

Command Mode

Privileged Exec mode and Exec mode

Example

```
#show ip vrf VRF_A
VRF VRF_A; (table=1)
```

Appendix A Regular Expressions

Table A-1 shows the regular expression special characters used in BGP commands. You can use these characters in combination to build complex regular expressions.

Table A-1: Regular expression characters

Symbol	Character	Meaning
^	Caret	Matches the beginning of the input string. When used at the beginning of a string of characters, it negates a pattern match.
$	Dollar sign	Matches the end of the input string.
.	Period	Matches a single character (including white spaces).
*	Asterisk	Matches none or more sequences of a pattern.
+	Plus sign	Matches one or more sequences of a pattern.
?	Question mark	Matches none or one occurrence of a pattern.
_	Underscore	Matches spaces, commas, braces, parenthesis, or the beginning and end of an input string.
[]	Brackets	A range of single-characters.
-	Hyphen	Separates the end points of a range.

Index