

The SONiC Industry Standard CLI Reference Guide

Table Of Commands

aaa accounting login default
aaa authentication failthrough
aaa authentication login default
aaa authorization login default
aaa name-service group
aaa name-service netgroup
aaa name-service passwd
aaa name-service shadow
aaa name-service sudoers
aaa server radius dynamic-author
activate
add-reason
addpath-tx-all-paths
addpath-tx-bestpath-per-as
address-family ipv4
address-family ipv6
address-family l2vpn
advertise ipv4 unicast
advertise ipv6 unicast
advertise-all-vni
advertise-default-gw
advertise-pip
advertise-svi-ip
advertisement-interval
aggregate-address
aging-interval
alarm acknowledge
alarm unacknowledge
alias
allowas-in
always-compare-med
area
as-override
attribute-unchanged
auth-type
authentication command bounce-port ignore
authentication command disable-port ignore
authentication dynamic-vlan enable

authentication event fail action authorize vlan
authentication event fail retry
authentication event no-response action authorize vlan
authentication host-mode
authentication max-users
authentication monitor
authentication open
authentication order
authentication periodic
authentication port-control
authentication priority
authentication rest
authentication telemetry
authentication timer reauthenticate
authorization network radius
auto-cost
autoneg
autort
autostate
bestpath as-path confed
bestpath as-path ignore
bestpath as-path multipath-relax
bestpath compare-routerid
bestpath med
bfd
bfd check-control-plane-failure
bfd profile
bgp as-path-list
bgp community-list
bgp extcommunity-list
binding
buffer init
buffer pool
buffer priority-group
buffer profile
buffer queue
call
capability
capability dynamic
capability extended-nexthop
capability orf prefix-list

channel-group
class
class-map
clear access-list counters
clear audit-log
clear authentication history interface
clear authentication sessions interface
clear bfd peer
clear bgp all
clear bgp ipv4
clear bgp ipv6
clear bgp l2vpn evpn
clear buffer-pool
clear core-files
clear counters
clear counters service-policy
clear counters service-policy interface
clear counters service-policy policy-map
clear counters tam
clear counters vxlan
clear evpn dup-addr
clear ip arp
clear ip arp interface
clear ip dhcp snooping binding
clear ip dhcp snooping statistics
clear ip dhcp snooping statistics detail
clear ip dhcp-relay
clear ip helper-address statistics
clear ip ospf
clear ip pim
clear ip sla
clear ip sla all
clear ipv6 dhcp snooping binding
clear ipv6 dhcp snooping statistics
clear ipv6 dhcp snooping statistics detail
clear ipv6 dhcp-relay
clear ipv6 neighbors
clear ipv6 neighbors interface
clear logging
clear mac address-table dynamic
clear mac dampening-disabled-ports

clear nat
clear priority-group
clear queue
clear radius-server dynamic-author statistics
clear radius-server statistics
client
client-to-client reflection
clock timezone
cluster-id
coalesce-time
collector
compatible
confederation
configure terminal
copp-action
copy
core enable
counter
counters access-list enable
counters access-list interval
counters buffer-drop
counters pg-drop
counters policer
counters port
counters queue
counters rif
counters watermark
crm
dampening
default interface
default interface range
default ipv4-unicast
default local-preference
default show-hostname
default shutdown
default subgroup-pkt-queue-max
default-information
default-metric
default-originate
default-originate ipv4
default-originate ipv6

delay-restore
delete-reason
description
destination
destination CPU
destination erspan
destination-group
detect-multiplier
deterministic-med
df-preference
diag-mode
disable-connected-check
disable-ead-evi-rx
disable-ead-evi-tx
disable-ebgp-connected-route-check
distance
distance bgp
dont-capability-negotiate
dot1p
dot1x pae
dot1x system-auth-control
dot1x timeout quiet-period
dot1x timeout server-timeout
downstream
drop-monitor
dropcounters
dscp
dst-grp
dup-addr-detection
dup-addr-detection freeze
ebgp-multipoint
echo-interval
echo-mode
ecn
enable
encapsulation dot1q vlan-id
enforce-first-as
enforce-multipoint
enterprise-id
entry
errdisable recovery cause

errdisable recovery interval
event profile
evpn ethernet-segment
external-ip
fabric-external
factory
fallback
fast-external-failover
fast-reboot
fast_rate
fec
filter-list
flow-group
frequency
global
graceful-restart
graceful-restart enable
graceful-restart preserve-fw-state
graceful-restart restart-time
graceful-restart stalepath-time
graceful-shutdown
green
group
grpc port
hostname
icmp-echo
ifa
ignore server-key
ignore session-key
import vrf
interface
interface CPU
interface Loopback
interface Management
interface PortChannel
interface breakout
interface media-fec
interface range
interface transceiver diagnostics
interface vxlan
ip

ip access-group
ip access-list
ip address
ip anycast-address
ip anycast-mac-address
ip arp
ip dhcp smart-relay
ip dhcp snooping
ip dhcp snooping trust
ip dhcp snooping verify mac-address
ip dhcp snooping vlan
ip dhcp-relay
ip dhcp-relay link-select
ip dhcp-relay max-hop-count
ip dhcp-relay policy-action
ip dhcp-relay source-interface
ip dhcp-relay vrf-select
ip forward-protocol udp enable
ip forward-protocol udp exclude
ip forward-protocol udp include
ip forward-protocol udp rate-limit
ip host-access-list
ip load-share hash rotate
ip name-server
ip name-server source-interface
ip nht
ip ospf
ip ospf area
ip ospf authentication
ip ospf authentication-key
ip ospf bfd
ip ospf bfd profile
ip ospf cost
ip ospf dead-interval
ip ospf hello-interval
ip ospf message-digest-key
ip ospf mtu-ignore
ip ospf network
ip ospf priority
ip ospf retransmit-interval
ip ospf transmit-delay

ip prefix-list
ip reserve
ip route
ip route vrf
ip route vrf mgmt
ip sla
ip source binding
ip unnumbered
ip vrf
ip vrf forwarding
ip vrf mgmt
ipv6 access-group
ipv6 access-list
ipv6 address
ipv6 anycast-address
ipv6 dhcp snooping
ipv6 dhcp snooping trust
ipv6 dhcp snooping verify mac-address
ipv6 dhcp snooping vlan
ipv6 dhcp-relay
ipv6 dhcp-relay max-hop-count
ipv6 dhcp-relay source-interface
ipv6 dhcp-relay vrf-select
ipv6 enable
ipv6 host-access-list
ipv6 nd
ipv6 neighbor
ipv6 nht
ipv6 ospf6
ipv6 ospf6 area
ipv6 ospf6 bfd
ipv6 ospf6 bfd profile
ipv6 ospf6 cost
ipv6 ospf6 dead-interval
ipv6 ospf6 hello-interval
ipv6 ospf6 mtu-ignore
ipv6 ospf6 network
ipv6 ospf6 priority
ipv6 ospf6 retransmit-interval
ipv6 ospf6 transmit-delay
ipv6 prefix-list

ipv6 route
ipv6 route vrf
ipv6 route vrf mgmt
ipv6 source binding
ipv6 vrf
kdump enable
kdump memory
kdump num-dumps
keepalive-interval
key config-key password-encrypt
link debounce link-up time
link debounce time
link state track
link-error-disable
listen limit
listen range
lldp
lldp enable
lldp med-tlv-select
lldp multiplier
lldp system-description
lldp system-name
lldp timer
lldp tlv-select
lldp tlv-set
local-as
locator-led chassis off
locator-led chassis on
log-adjacency-changes
log-neighbor-changes
logger
logging console
logging level
logging monitor
logging profile
logging sai-level
logging server
logging sonic-level
logging template
mab
mab request format attribute 1 groupsize

```
mab timeout server-timeout  
mac access-group  
mac access-list  
mac address-table  
mac address-table aging-time  
mac address-table dampening-interval  
mac address-table dampening-threshold  
mac move-policy  
mac-address  
map  
match access-group  
match as-path  
match community  
match dei  
match destination-address  
match dscp  
match ethertype  
match evpn  
match ext-community  
match interface  
match ip address prefix-list  
match ip next-hop prefix-list  
match ip protocol  
match ipv6 address prefix-list  
match l4-port  
match local-preference  
match metric  
match origin  
match pcp  
match peer  
match protocol  
match source-address  
match source-protocol  
match source-vrf  
match tag  
match tcp-flags  
match vlan  
max-med  
max-metric  
maximum-paths  
maximum-paths ibgp
```

maximum-prefix
mclag
mclag domain
mclag gateway-mac
mclag-peer-gateway
mclag-separate-ip
mclag-system-mac
min-links
minimum-ttl
mirror
mirror-session
mtu
nat
nat-zone
neigh-suppress
neighbor
neighbor-route
network
network import-check
network-policy
next-hop-self
no
ntp master
ntp server
ntp source-interface
ntp vrf
ospf
ospf abr-type
ospf router-id
ospf6
ospf6 router-id
override-capability
passive
passive-interface
passive-mode
password
path
path-target
pbf next-hop-group
pbf replication-group
pbs

peer
peer-group
peer-ip
peer-link
pfc-priority
ping
ping vrf
ping vrf mgmt
ping6
ping6 vrf
ping6 vrf mgmt
pir
police
policy-map
pool
port
port-group
port-number-base
port-security enable
port-security maximum
port-security violation
portchannel graceful-shutdown
preempt
prefix-list
primary-ip
priority
priority-flow-control
priority-flow-control watchdog action
priority-flow-control watchdog counter-poll
priority-flow-control watchdog detect-time
priority-flow-control watchdog off
priority-flow-control watchdog on
priority-flow-control watchdog polling-interval
priority-flow-control watchdog restore-time
profile
proxy-arp
ptp announce-timeout
ptp domain
ptp domain-profile
ptp ipv6-scope
ptp log-announce-interval

ptp log-min-delay-req-interval
ptp log-sync-interval
ptp mode
ptp network-transport
ptp port add
ptp port del
ptp port master-table
ptp priority1
ptp priority2
ptp two-step
qos map dot1p-tc
qos map dscp-tc
qos map pfc-priority-queue
qos map tc-dot1p
qos map tc-dscp
qos map tc-pg
qos map tc-queue
qos scheduler-policy
qos wred-policy
qos-map dot1p-tc
qos-map dscp-tc
qos-map pfc-priority-queue
qos-map tc-dot1p
qos-map tc-dscp
qos-map tc-pg
qos-map tc-queue
qos-mode
queue
radius-server auth-type
radius-server host
radius-server key
radius-server nas-ip
radius-server retransmit
radius-server statistics
radius-server timeout
rd
read-quanta
reboot
receive-interval
redistribute
refresh

remark
remote-as
remove-private-as
renew dhcp-lease
report-interval
report-type
request-data-size
route-map
route-map delay-timer
route-reflector allow-outbound-policy
route-reflector-client
route-scale
route-scale hosts
route-server-client
route-target
router bgp
router ospf
router ospf6
router-id
run
sampler
scheduler-policy
send-community
sensor-group
seq
server-key
service
service-policy
session
session-timeout
set as-path prepend
set as-path set
set comm-list
set community
set copp-action
set dscp
set extcommunity
set interface
set ip
set ipv6
set ipv6 next-hop

set local-preference
set metric
set mirror-session
set origin
set pcp
set traffic-class
set trap-action
set trap-priority
set trap-queue
set weight
sflow agent-id
sflow collector
sflow default
sflow enable
sflow interface enable
sflow polling-interval
sflow sampling-direction
sflow sampling-rate
sflow source-ip
show PortChannel summary
show Vlan
show aaa
show access-group
show access-list counters
show access-lists summary
show alarm acknowledged
show alarm all
show alarm detail
show alarm id
show alarm recent
show alarm severity
show alarm summary
show audit-log
show authentication
show authentication authentication-history
show authentication clients
show authentication interface
show authentication rest
show authentication telemetry
show bfd peer
show bfd peer counters

show bfd peers
show bfd profile
show bgp all
show bgp as-path-access-list
show bgp community-list
show bgp ext-community-list
show bgp ipv4
show bgp ipv6
show bgp l2vpn evpn route
show bgp l2vpn evpn route detail
show bgp l2vpn evpn route detail type
show bgp l2vpn evpn route rd
show bgp l2vpn evpn route type
show bgp l2vpn evpn route type macip
show bgp l2vpn evpn route type multicast
show bgp l2vpn evpn route type prefix
show bgp l2vpn evpn route vni
show bgp l2vpn evpn summary
show bgp l2vpn evpn vni
show buffer interface
show buffer pool
show buffer profile
show buffer-pool
show class-map
show clock
show config-key password-encrypt
show configuration
show core config
show core info
show core list
show counters configuration
show crm
show database map
show dot1x
show dot1x detail
show dropcounters capabilities
show dropcounters configuration
show dropcounters configuration detail
show errdisable link-flap
show errdisable recovery
show event details

show event id
show event profile
show event recent
show event severity
show event summary
show evpn
show evpn arp-cache vni
show evpn arp-cache vni all
show evpn mac vni
show evpn mac vni all
show evpn next-hops vni
show evpn next-hops vni all
show evpn rmac vni
show evpn rmac vni all
show evpn vni
show evpn vni detail
show grpc
show histogram memory
show hosts
show in-memory-logging
show in-memory-logging count
show in-memory-logging lines
show interface
show interface breakout
show ip access-group
show ip access-lists
show ip arp
show ip arp interface
show ip dhcp smart-relay
show ip dhcp snooping
show ip dhcp snooping binding
show ip dhcp snooping statistics
show ip dhcp snooping statistics detail
show ip dhcp-relay
show ip forward-protocol
show ip helper-address
show ip helper-address statistics
show ip host-access-lists
show ip interfaces
show ip load-share
show ip ospf

show ip prefix-list
show ip route
show ip sla
show ip static-anycast-gateway
show ip vrf
show ip vrf mgmt
show ipv6 access-group
show ipv6 access-lists
show ipv6 dhcp snooping
show ipv6 dhcp snooping binding
show ipv6 dhcp snooping statistics
show ipv6 dhcp snooping statistics detail
show ipv6 dhcp-relay
show ipv6 host-access-lists
show ipv6 interfaces
show ipv6 neighbors
show ipv6 neighbors interface
show ipv6 ospf6
show ipv6 ospf6 vrfs
show ipv6 prefix-list
show ipv6 route
show ipv6 static-anycast-gateway
show kdump files
show kdump log
show kdump memory
show kdump num-dumps
show kdump status
show link state tracking
show lldp neighbor
show lldp statistics
show lldp table
show locator-led
show logging
show logging count
show logging filter
show logging lines
show logging profiles
show logging servers
show mab
show mab interface
show mac access-group

show mac access-lists
show mac address-table
show mac address-table Vlan
show mac address-table address
show mac address-table aging-time
show mac address-table count
show mac address-table dynamic
show mac address-table interface
show mac address-table static
show mac dampening
show mac dampening-disabled-ports
show mac move-policy
show mac move-policy status
show mclag brief
show mclag interface
show mclag mac remote
show mclag mac remote count
show mclag peer-gateway-interfaces
show mclag separate-ip-interfaces
show mirror-session
show nat
show neighbor-suppress-status
show ntp associations
show ntp global
show ntp server
show object-groups
show pbf next-hop-group
show pbf next-hop-group status Switch
show pbf next-hop-group status interface
show pbf replication-group
show pbf replication-group status Switch
show pbf replication-group status interface
show platform environment
show platform fanstatus
show platform firmware
show platform firmware detail
show platform inventory
show platform psustatus
show platform psusummary
show platform sbstatus
show platform syseeprom

```
show platform temperature
show platform temperature detail
show policy-map
show port-group
show port-security
show port-security interface
show priority-flow-control
show priority-group
show ptp
show ptp clock
show ptp parent
show ptp port
show ptp time-property
show qos
show qos map dot1p-tc
show qos map dscp-tc
show qos map pfc-priority-queue
show qos map tc-dot1p
show qos map tc-dscp
show qos map tc-pg
show qos map tc-queue
show qos scheduler-policy
show qos wred-policy
show queue
show radius-server
show radius-server dynamic-author
show radius-server dynamic-author statistics
show reboot-cause
show route-map
show running-configuration
show running-configuration bfd
show running-configuration bgp
show running-configuration bgp as-path-access-list
show running-configuration bgp community-list
show running-configuration bgp extcommunity-list
show running-configuration bgp neighbor
show running-configuration bgp peer-group
show running-configuration class-map
show running-configuration dropcounters
show running-configuration interface
show running-configuration interface Loopback
```

```
show running-configuration interface Management
show running-configuration interface PortChannel
show running-configuration interface Vlan
show running-configuration interface vxlan
show running-configuration ip access-list
show running-configuration ip host-access-list
show running-configuration ip prefix-list
show running-configuration ipv6 access-list
show running-configuration ipv6 host-access-list
show running-configuration ipv6 prefix-list
show running-configuration link state tracking
show running-configuration mac access-list
show running-configuration mclag
show running-configuration mirror-session
show running-configuration nat
show running-configuration ospf
show running-configuration ospf interface
show running-configuration ospf6
show running-configuration ospf6 interface
show running-configuration pac
show running-configuration pbf next-hop-group
show running-configuration pbf replication-group
show running-configuration policy-map
show running-configuration route-map
show running-configuration subinterface
show running-configuration tam
show running-configuration telemetry
show running-configuration vrf
show service-policy
show service-policy interface
show service-policy policy-map
show service-policy summary
show sflow
show sflow counters
show sflow interface
show snmp-server
show snmp-server community
show snmp-server group
show snmp-server host
show snmp-server user
show snmp-server view
```

```
show sonic-log-configuration
show ssh
show ssh-server vrf
show ssh-server vrf
show startup-configuration
show storm-control
show storm-control interface
show subinterfaces status
show switch-profiles
show switch-resource drop-monitor
show switch-resource route-scale
show system
show system cpu
show system cpu history
show system memory
show system processes
show system processes cpu
show system processes mem-usage
show system processes mem-util
show system processes pid
show system status
show system vlan
show tacacs-server
show tacacs-server global
show tacacs-server host
show tam collectors
show tam drop-monitor
show tam drop-monitor sessions
show tam features
show tam flowgroups
show tam ifa
show tam ifa sessions
show tam samplers
show tam switch
show tam tail-stamping
show tam tail-stamping sessions
show tech-support
show techsupport-export
show telemetry data collector details
show telemetry destination-groups
show telemetry global
```

```
show telemetry sensor-group
show telemetry sensor-path
show telemetry subscription
show telemetry transport
show threshold breaches
show uptime
show users
show version
show vrrp
show vrrp6
show vxlan counters
show vxlan interface
show vxlan remote mac
show vxlan remote mac count
show vxlan remote vni
show vxlan remote vni count
show vxlan tunnel
show vxlan tunnel count
show vxlan vlanvnimap
show vxlan vlanvnimap count
show vxlan vrfvnimap
show vxlan vrfvnimap count
show warm-restart
show warm-restart check
show warm-restart config
show warm-restart system
show watermark interval
show watermark telemetry
show ztp-status
shutdown
snmp-server agentaddress
snmp-server community
snmp-server contact
snmp-server enable
snmp-server engine
snmp-server group
snmp-server host
snmp-server location
snmp-server user
snmp-server view
snsr-grp
```

soft-reconfiguration
solo
source-address
source-interface
source-ip
source-port
source-vrf
speed
ssh login_attempts
ssh login_timeout
ssh ports
ssh tcp-forwarding
ssh x11-forwarding
ssh-server vrf
standalone-link-training
static
storm-control broadcast
storm-control unknown-multicast
storm-control unknown-unicast
strict-capability-match
stub-router
subscription
switch-id
switch-resource
switchport access
switchport trunk
system vlan
system-mac
table-map
tacacs-server auth-type
tacacs-server host
tacacs-server key
tacacs-server source-interface
tacacs-server timeout
tail-stamping
tam
tcp-connect
tcp-timeout
tcpdump
tcpdump CPU
tcpdump read

techsupport-export enable
techsupport-export interval
techsupport-export remote-server
telemetry
terminal length
terminal timeout
threshold
threshold buffer-pool
threshold device
threshold priority-group
threshold queue
timeout
timers
tos
traceroute
traceroute vrf
traceroute vrf mgmt
traceroute6
traceroute6 vrf
traceroute6 vrf mgmt
track-interface
traffic-class
transmit-interval
ttl
ttl-security hops
type
udp-timeout
unreliable-los
unsuppress-map
update-delay
update-source
use-es-l3nhg
use-v2-checksum
username
v6only
version
vip
vni
vni-downstream
voice
voice-signaling

vrrp
warm-reboot
warm-restart bgp
warm-restart bgp enable
warm-restart bgp eoui
warm-restart swss
warm-restart swss enable
warm-restart system
warm-restart teamd
warm-restart teamd enable
watermark interval
watermark telemetry
weight
write
write erase install
write-multiplier
write-quanta
yellow
ztp

CLIs By Feature

COREDUMP

core enable
clear core-files
show core list
show core config
show core info

DHCP Snooping

show ip dhcp snooping
ip dhcp snooping
ip dhcp snooping verify mac-address
ipv6 dhcp snooping
ipv6 dhcp snooping verify mac-address

DHCP smart-relay

show ip dhcp smart-relay
ip dhcp smart-relay

ERRDISABLE

show errdisable recovery
show errdisable link-flap
errdisable recovery cause
errdisable recovery interval

HASH

show ip load-share

IP Helper

show ip forward-protocol
show ip helper-address
show ip helper-address statistics
clear ip helper-address statistics
ip forward-protocol udp enable
ip forward-protocol udp include
ip forward-protocol udp exclude
ip forward-protocol udp rate-limit

IPSLA

show ip sla
show configuration
show configuration
show configuration

IPv4 Unnumbered Interface

ip unnumbered
ip unnumbered
ip unnumbered
ip unnumbered

KDUMP

kdump enable
kdump memory
kdump num-dumps
show kdump status
show kdump memory
show kdump num-dumps
show kdump files
show kdump log

LLDP

```
lldp enable
lldp
lldp timer
lldp multiplier
lldp system-name
lldp system-description
lldp tlv-select
lldp
lldp enable
lldp tlv-set
```

LLDP-MED

```
network-policy
lldp med-tlv-select
lldp tlv-select
network-policy
voice
voice-signaling
```

NAT

```
show configuration
```

ND

```
ipv6 nd
ipv6 nd
ipv6 nd
ipv6 nd
```

Nexthop Tracking

```
ipv6 nht
ip nht
```

OSPFv2

```
router ospf
show configuration
area
auto-cost
capability
compatible
graceful-restart
default-information
default-metric
```

distance
log-adjacency-changes
max-metric
network
ospf
ospf abr-type
ospf router-id
passive-interface
redistribute
refresh
timers
write-multiplier
ip ospf
ip ospf area
ip ospf authentication
ip ospf authentication-key
ip ospf bfd
ip ospf bfd profile
ip ospf cost
ip ospf hello-interval
ip ospf message-digest-key
ip ospf mtu-ignore
ip ospf network
ip ospf priority
ip ospf retransmit-interval
ip ospf transmit-delay
ip ospf
ip ospf area
ip ospf authentication
ip ospf authentication-key
ip ospf bfd
ip ospf bfd profile
ip ospf cost
ip ospf hello-interval
ip ospf message-digest-key
ip ospf mtu-ignore
ip ospf network
ip ospf priority
ip ospf retransmit-interval
ip ospf transmit-delay
ip ospf

ip ospf area
ip ospf authentication
ip ospf authentication-key
ip ospf bfd
ip ospf bfd profile
ip ospf cost
ip ospf hello-interval
ip ospf message-digest-key
ip ospf mtu-ignore
ip ospf network
ip ospf priority
ip ospf retransmit-interval
ip ospf transmit-delay
ip ospf
ip ospf area
ip ospf authentication
ip ospf authentication-key
ip ospf bfd
ip ospf bfd profile
ip ospf cost
ip ospf hello-interval
ip ospf message-digest-key
ip ospf mtu-ignore
ip ospf network
ip ospf priority
ip ospf retransmit-interval
ip ospf transmit-delay
ip ospf
ip ospf area
ip ospf authentication
ip ospf authentication-key
ip ospf bfd
ip ospf bfd profile
ip ospf cost
ip ospf hello-interval
ip ospf message-digest-key
ip ospf mtu-ignore
ip ospf network
ip ospf priority
ip ospf retransmit-interval
ip ospf transmit-delay

```
show ip ospf
```

OSPFv3

```
router ospf6
auto-cost
distance
log-adjacency-changes
maximum-paths
ospf6
ospf6 router-id
stub-router
timers
write-multiplier
ipv6 ospf6
ipv6 ospf6 area
ipv6 ospf6 bfd
ipv6 ospf6 bfd profile
ipv6 ospf6 cost
ipv6 ospf6 hello-interval
ipv6 ospf6 mtu-ignore
ipv6 ospf6 network
ipv6 ospf6 priority
ipv6 ospf6 retransmit-interval
ipv6 ospf6 transmit-delay
```

Subinterface

```
show subinterfaces status
show running-configuration subinterface
encapsulation dot1q vlan-id
mtu
shutdown
```

Switch-resource

```
show configuration
```

VRRP

```
show vrrp
show vrrp6
```

ZTP

```
show ztp-status
ztp
```

aaa accounting login default

Description

```
Configures AAA login authorization default list to use tacacs+
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
aaa accounting login default { { [ group { tacacs+ [ local ] } ] } | { [ local { [ group [ tacacs+ ] ] } ] }
```

```
no aaa accounting login default
```

Examples

```
sonic(config)# aaa authorization login default group tacacs+ local
```

aaa authentication failthrough

Description

```
Configures AAA authentication failthrough
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
aaa authentication failthrough <enable>
```

Parameters

Name	Description	Type
enable	enable or disable	Select [enable(enable) disable(disable)]

Examples

```
sonic(config)# aaa authentication failthrough enable
```

aaa authentication login default

Description

```
Configures AAA login authentication default list to authentication first with tacacs+, and, if none respond, (or failthrough is configured) with local next
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
aaa authentication login default { { [ group { { [ radius [ local ] ] } | { [ tacacs+ [ local ] ] } ] } | { [ local { [ group { [ radius ] | [ tacacs+ ] } ] } ] } }
```

```
no aaa authentication login default
```

Examples

```
sonic(config)# aaa authentication login default group tacacs+ local
```

aaa authorization login default

Description

```
Configures AAA login authorization default list to use tacacs+
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
aaa authorization login default { { [ group { tacacs+ [ local ] } ] } | { [ local { [ group [ tacacs+ ] ] } ] }
```

```
no aaa authorization login default
```

Examples

```
sonic(config)# aaa authorization login default group tacacs+ local
```

aaa name-service group

Description

```
Configures AAA group name-service to use ldap
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
aaa name-service group { { [ group ldap ] } | [ local ] | [ login ] }
```

```
no aaa name-service group
```

Examples

```
sonic(config)# aaa name-service group group ldap
```

aaa name-service netgroup

Description

```
Configures AAA netgroup name-service to use ldap
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
aaa name-service netgroup { { [ group ldap ] } | [ local ] }
```

```
no aaa name-service netgroup
```

Examples

```
sonic(config)# aaa name-service netgroup group ldap
```

aaa name-service passwd

Description

```
Configures AAA passwd name-service to use ldap
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
aaa name-service passwd { { [ group ldap ] } | [ local ] | [ login ] }
no aaa name-service passwd
```

Examples

```
sonic(config)# aaa name-service passwd group ldap
```

aaa name-service shadow

Description

```
Configures AAA shadow name-service to use ldap
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
aaa name-service shadow { { [ group ldap ] } | [ local ] | [ login ] }
no aaa name-service shadow
```

Examples

```
sonic(config)# aaa name-service shadow group ldap
```

aaa name-service sudoers

Description

```
Configures AAA sudoers name-service to use ldap
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
aaa name-service sudoers { { [ group ldap ] } | [ local ] }
no aaa name-service sudoers
```

Examples

```
sonic(config)# aaa name-service sudoers group ldap
```

aaa server radius dynamic-author

Description

This command enables DAS functionality and enters dynamic authorization local server configuration mode.

Parent Commands (Modes)

configure terminal

Syntax

```
aaa server radius dynamic-author  
no aaa server radius dynamic-author
```

Usage Guidelines

This command enables DAS functionality and enters dynamic authorization local server configuration mode.

Examples

```
sonic-cl# configure terminal  
sonic-cl(config)# aaa server radius dynamic-author  
sonic-cl(config-radius-da)#

```

activate

Description

This command enables/activates a particular address-family for a BGP neighbor

Parent Commands (Modes)

address-family ipv4 unicast

Syntax

```
activate  
no activate
```

Usage Guidelines

Use this command to activate an address-family for a BGP neighbor.
This command can be executed multiple times to enable multiple address families for a BGP neighbor

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 20.20.20.2  
sonic(config-router-bgp-neighbor)# remote-as 300  
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast  
sonic(config-router-bgp-neighbor-af)# activate
```

activate

Description

This command enables/activates a particular address-family for a BGP peer-group

Parent Commands (Modes)

address-family ipv4 unicast

Syntax

```
activate  
no activate
```

Usage Guidelines

Use this command to activate an address-family for a BGP peer-group. This command can be executed multiple times to enable multiple address families for a BGP peer-group

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Ext  
sonic(config-router-bgp Pg)# remote-as 300  
sonic(config-router-bgp Pg)# address-family ipv4 unicast  
sonic(config-router-bgp Pg-af)# activate
```

activate

Description

This command enables/activates a particular address-family for a BGP neighbor

Parent Commands (Modes)

address-family l2vpn evpn

Syntax

```
activate  
no activate
```

Usage Guidelines

Use this command to activate an address-family for a BGP neighbor.
This command can be executed multiple times to enable multiple address families for a BGP neighbor

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)# activate
```

activate

Description

This command enables/activates a particular address-family for a BGP peer-group

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
activate
no activate
```

Usage Guidelines

Use this command to activate an address-family for a BGP peer-group.
This command can be executed multiple times to enable multiple address families for a BGP peer-group

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp-pg)# remote-as 300
sonic(config-router-bgp-pg)# address-family l2vpn evpn
sonic(config-router-bgp-pg-af)# activate
```

activate

Description

Enable the Address Family for this Neighbor

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
activate  
no activate
```

activate

Description

Enable the Address Family for this Neighbor

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
activate  
no activate
```

add-reason

Description

Add dropcounter reason

Parent Commands (Modes)

dropcounters <counter-name>

Syntax

```
add-reason <reason>
```

Parameters

Name	Description	Type
reason	Drop reason list	String

addpath-tx-all-paths

Description

This command enables BGP to advertise all paths to a neighbor.

Parent Commands (Modes)

address-family ipv4 unicast

Syntax

```
addpath-tx-all-paths
no addpath-tx-all-paths
```

Usage Guidelines

Use this command for BGP add path feature configuration. This command allows all BGP paths to be advertised to a neighbor

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# addpath-tx-all-paths
```

addpath-tx-all-paths

Description

This command enables BGP to advertise all paths to neighbors in a peer-group.

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
addpath-tx-all-paths  
no addpath-tx-all-paths
```

Usage Guidelines

Use this command for BGP add path feature configuration. This command allows all BGP paths to be advertised to neighbors in a peer-group

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Ext  
sonic(config-router-bgp Pg)# address-family ipv4 unicast  
sonic(config-router-bgp Pg-af)# addpath-tx-all-paths
```

addpath-tx-all-paths

Description

Use addpath to advertise all paths to a neighbor

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
addpath-tx-all-paths  
no addpath-tx-all-paths
```

addpath-tx-all-paths

Description

Use addpath to advertise all paths to a neighbor

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
addpath-tx-all-paths  
no addpath-tx-all-paths
```

addpath-tx-bestpath-per-as

Description

```
This command enables BGP to advertise best paths to a neighbor.
```

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
addpath-tx-bestpath-per-as  
no addpath-tx-bestpath-per-as
```

Usage Guidelines

```
Use this command for BGP add path feature configuration. This command  
allows best BGP path to be advertised to a neighbor
```

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 20.20.20.2  
sonic(config-router-bgp-neighbor)# remote-as 300  
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast  
sonic(config-router-bgp-neighbor-af)# addpath-tx-bestpath-per-as
```

addpath-tx-bestpath-per-as

Description

```
This command enables BGP to advertise best paths to neighbors in a  
peer-group.
```

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
addpath-tx-bestpath-per-as  
no addpath-tx-bestpath-per-as
```

Usage Guidelines

Use this command for BGP add path feature configuration. This command allows best BGP path to be advertised to neighbors in a peer-group

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Ext  
sonic(config-router-bgp Pg)# address-family ipv4 unicast  
sonic(config-router-bgp Pg-af)# addpath-tx-bestpath-per-as
```

addpath-tx-bestpath-per-as

Description

Use addpath to advertise the bestpath per each neighboring AS

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
addpath-tx-bestpath-per-as  
no addpath-tx-bestpath-per-as
```

addpath-tx-bestpath-per-as

Description

Use addpath to advertise the bestpath per each neighboring AS

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
addpath-tx-bestpath-per-as  
no addpath-tx-bestpath-per-as
```

address-family ipv4

Description

This command enters into IPv4 Unicast address-family configure CLI context

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
address-family ipv4 unicast  
no address-family ipv4 unicast
```

Usage Guidelines

Use this command to switch to IPv4 Unicast address family CLI context to configure parameters specific to this address family

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# address-family ipv4 unicast  
sonic(config-router-bgp-af)#[
```

address-family ipv4

Description

This command enters into IPv4 Unicast address-family configuration CLI context for a BGP neighbor

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
address-family ipv4 unicast  
no address-family ipv4 unicast
```

Usage Guidelines

Use this command to switch to IPv4 Unicast address family CLI context of a BGP neighbor to configure parameters specific to this address family

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# neighbor 30.30.30.3  
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast  
sonic(config-router-bgp-neighbor-af)#[/pre>
```

address-family ipv4

Description

This command switches the CLI context into IPv4 Unicast address-family mode for a BGP peer-group

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
address-family ipv4 unicast  
no address-family ipv4 unicast
```

Usage Guidelines

Use this command to switch to IPv4 Unicast address family CLI context of a BGP peer-group to configure parameters specific to this address family

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# address-family ipv4 unicast
sonic(config-router-bgp Pg-af)#

```

address-family ipv6

Description

This command execution enters into IPv6 Unicast address-family configuration CLI context

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
address-family ipv6 unicast
no address-family ipv6 unicast
```

Usage Guidelines

Use this command to switch to IPv6 Unicast address family CLI context to configure parameters specific to this address family

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# address-family ipv6 unicast
sonic(config-router-bgp-af)#

```

address-family ipv6

Description

This command execution enters into IPv6 Unicast address-family configuration CLI context for a BGP neighbor

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
address-family ipv6 unicast  
no address-family ipv6 unicast
```

Usage Guidelines

Use this command to switch to IPv6 Unicast address family CLI context of a BGP neighbor to configure parameters specific to this address family

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# neighbor 30.30.30.3  
sonic(config-router-bgp-neighbor)# address-family ipv6 unicast  
sonic(config-router-bgp-neighbor-af)#[/pre>
```

address-family ipv6

Description

This command switches the CLI context into IPv6 Unicast address-family mode for a BGP peer-group

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
address-family ipv6 unicast  
no address-family ipv6 unicast
```

Usage Guidelines

Use this command to switch to IPv6 Unicast address family CLI context of a BGP peer-group to configure parameters specific to this address family

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# peer-group PG_Ext  
sonic(config-router-bgp Pg)# address-family ipv6 unicast  
sonic(config-router-bgp Pg-af)#[/pre>
```

address-family l2vpn

Description

This command execution enters into l2vpn evpn address-family configure CLI context

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
address-family l2vpn evpn  
no address-family l2vpn evpn
```

Usage Guidelines

Use this command to switch to l2vpn evpn address family CLI context to configure parameters specific to this address family

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# address-family l2vpn evpn  
sonic(config-router-bgp-af)#[
```

address-family l2vpn

Description

This command execution enters into L2VPN EVPN address-family configuration CLI context for a BGP neighbor

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
address-family l2vpn evpn  
no address-family l2vpn evpn
```

Usage Guidelines

Use this command to switch to L2VPN EVPN address family CLI context of a BGP neighbor to configure parameters specific to this address family

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)#[/pre>
```

address-family l2vpn

Description

This command switches the CLI context into L2VPN EVPN address-family mode for a BGP peer-group

Parent Commands (Modes)

```
peer-group <template-str>[/pre>
```

Syntax

```
address-family l2vpn evpn
no address-family l2vpn evpn[/pre>
```

Usage Guidelines

Use this command to switch to L2VPN EVPN address family CLI context of a BGP peer-group to configure parameters specific to this address family

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# address-family l2vpn evpn
sonic(config-router-bgp Pg-af)#[/pre>
```

advertise ipv4 unicast

Description

This command enables tenant VRFs to announce IPv4 prefixes as EVPN type-5 routes

Parent Commands (Modes)

address-family 12vpn evpn

Syntax

```
advertise ipv4 unicast [ route-map <route-map-name> ]  
no advertise ipv4 unicast [ route-map <route-map-name> ]
```

Parameters

Name	Description	Type
route-map-name	WORD	String

Usage Guidelines

[no] advertise ipv4 unicast

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100 vrf Vrf1  
sonic(config-router-bgp)# address-family 12vpn evpn  
sonic(config-router-bgp-af)# advertise ipv4 unicast
```

advertise ipv6 unicast

Description

This command enables tenant VRFs to announce IPv6 prefixes as EVPN type-5 routes

Parent Commands (Modes)

address-family 12vpn evpn

Syntax

```
advertise ipv6 unicast [ route-map <route-map-name> ]  
no advertise ipv6 unicast [ route-map <route-map-name> ]
```

Parameters

Name	Description	Type
route-map-name	WORD	String

Usage Guidelines

```
[no] advertise ipv6 unicast
```

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100 vrf Vrf1  
sonic(config-router-bgp)# address-family l2vpn evpn  
sonic(config-router-bgp-af)# advertise ipv6 unicast
```

advertise-all-vni

Description

This command enables BGP control plane for all locally-configured VNIs

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
advertise-all-vni  
no advertise-all-vni
```

Usage Guidelines

```
[no] advertise-all-vni
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# advertise-all-vni
```

advertise-default-gw

Description

This command enables gateway advertisement for a particular VNI

Parent Commands (Modes)

```
vni <vninum>
```

Syntax

```
advertise-default-gw
no advertise-default-gw
```

Usage Guidelines

```
[no] advertise-default-gw
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# vni 100
sonic(config-router-bgp-af-vni)# advertise-default-gw
```

advertise-default-gw

Description

This command enables gateways VTEPs to advertise their IP/MAC addresses

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
advertise-default-gw
no advertise-default-gw
```

Usage Guidelines

```
[no] advertise-default-gw
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# advertise-default-gw
```

advertise-pip

Description

This command configure PIP parameters

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
advertise-pip { { [ ip { <advertise-pip-ip> { { [ peer-ip <advertise-pip-peer-ip> ] } ] } ] }
| { [ peer-ip <advertise-pip-peer-ip> ] } } ]
no advertise-pip { { [ ip { [ peer-ip ] ] } ] } | [ peer-ip ] }
```

Parameters

Name	Description	Type
advertise-pip-ip	A.B.C.D	String
advertise-pip-peer-ip	A.B.C.D	String

Usage Guidelines

```
[no] advertise-pip [ip A.B.C.D] [peer-ip A.B.C.D]
[no] advertise-pip [ip A.B.C.D]
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# advertise-pip
sonic(config-router-bgp-af)# advertise-pip ip 1.1.1.1
sonic(config-router-bgp-af)# advertise-pip ip 1.1.1.1 peer-ip 2.2.2.2
```

advertise-svi-ip

Description

This command enables svi mac-ip routes advertisement into EVPN for a particular VNI

Parent Commands (Modes)

```
vni <vninum>
```

Syntax

```
advertise-svi-ip
no advertise-svi-ip
```

Usage Guidelines

```
[no] advertise-svi-ip
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# vni 100
sonic(config-router-bgp-af-vni)# advertise-svi-ip
```

advertise-svi-ip

Description

This command enables svi mac-ip routes advertisement into EVPN

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
advertise-svi-ip  
no advertise-svi-ip
```

Usage Guidelines

```
[no] advertise-svi-ip
```

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# address-family l2vpn evpn  
sonic(config-router-bgp-af)# advertise-svi-ip
```

advertisement-interval

Description

```
Configure advertisement-interval for IPv4 VRRP instance
```

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
advertisement-interval <adv_interval>  
no advertisement-interval
```

Parameters

Name	Description	Type
adv_interval		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4) # vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1) #advertisement-interval 5
```

advertisement-interval

Description

Configure advertisement-interval for IPv6 VRRP instance

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
advertisement-interval <adv_interval>
no advertisement-interval
```

Parameters

Name	Description	Type
adv_interval		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4) # vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1) #advertisement-interval 2
```

advertisement-interval

Description

Configure advertisement-interval for IPv4 VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
advertisement-interval <adv_interval>
no advertisement-interval
```

Parameters

Name	Description	Type
adv_interval		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#advertisement-interval 5
```

advertisement-interval

Description

Configure advertisement-interval for IPv6 VRRP instance

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
advertisement-interval <adv_interval>
no advertisement-interval
```

Parameters

Name	Description	Type
adv_interval		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#advertisement-interval 2
```

advertisement-interval

Description

Configure advertisement-interval for IPv4 VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
advertisement-interval <adv_interval>
no advertisement-interval
```

Parameters

Name	Description	Type
adv_interval		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#advertisement-interval 5
```

advertisement-interval

Description

Configure advertisement-interval for IPv6 VRRP instance

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
advertisement-interval <adv_interval>
no advertisement-interval
```

Parameters

Name	Description	Type
adv_interval		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#advertisement-interval 2
```

advertisement-interval

Description

```
Configure advertisement-interval for IPv4 VRRP instance
```

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
advertisement-interval <adv_interval>
no advertisement-interval
```

Parameters

Name	Description	Type
adv_interval		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#advertisement-interval 5
```

advertisement-interval

Description

Configure advertisement-interval for IPv6 VRRP instance

Parent Commands (Modes)

vrrp ipv6

Syntax

```
advertisement-interval <adv_interval>
no advertisement-interval
```

Parameters

Name	Description	Type
adv_interval		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#advertisement-interval 2
```

advertisement-interval

Description

This command sets the minimum interval between sending BGP routing updates to a neighbor

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
advertisement-interval <tval>
no advertisement-interval
```

Parameters

Name	Description	Type
tval		Integer

Usage Guidelines

Use this command to set the minimum advertisement interval for BGP updates

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# advertisement-interval 10
```

advertisement-interval

Description

This command sets the minimum interval between sending BGP routing updates to neighbors in a peer-group

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
advertisement-interval <tval>
no advertisement-interval
```

Parameters

Name	Description	Type
tval		Integer

Usage Guidelines

Use this command to set the minimum advertisement interval for BGP updates

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp-pg)# advertisement-interval 10
```

aggregate-address

Description

This command configures an aggregate address and enables aggregation of routes that falls in the aggregate address subnet

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
aggregate-address <prefix> { [ as-set ] [ summary-only ] { [ route-map <rtemap> ] } }
no aggregate-address <prefix> { [ as-set ] [ summary-only ] { [ route-map <rtemap> ] } }
```

Parameters

Name	Description	Type
prefix	A.B.C.D/mask	String
rtemap	WORD	String

Usage Guidelines

This command enables user to turn on aggregation of BGP routes. "summary-only" option filters out all the aggregates routes and only the aggregate address will be advertised by BGP. "as-set" option will make sure that AS Path of individual aggregated routes are also included in the resulting aggregate route. "route-map" option gives user a finer control over the route's attributes

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# aggregate-address 17.35.0.0/16
```

aggregate-address

Description

Configure BGP aggregate entries

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
aggregate-address <prefix> { [ as-set ] [ summary-only ] { [ route-map <rtemap> ] } }
no aggregate-address <prefix> { [ as-set ] [ summary-only ] { [ route-map <rtemap> ] } }
```

Parameters

Name	Description	Type
prefix	A::B/mask	String
rtemap	WORD	String

aging-interval

Description

This command configures aging interval.

Parent Commands (Modes)

drop-monitor

Syntax

```
aging-interval <aging_interval>
no aging-interval
```

Parameters

Name	Description	Type
aging_interval	Aging interval	Integer

Usage Guidelines

This command configures aging interval. If the Drop Monitor feature doesn't notice packet drops for this duration, it considers packet drops to have stopped.

Examples

```
sonic# configure terminal
sonic(config)# tam
sonic(config-tam)# drop-monitor
sonic(config-tam-dm)# aging-interval 10
sonic(config-tam-dm)# end
sonic# show tam drop-monitor
Status          : Inactive
Switch ID       : 9876
Number of sessions : 3
Number of collectors : 2
Aging Interval    : 10
sonic#
```

alarm acknowledge

Description

```
Acknowledge an active alarm
```

Syntax

```
alarm acknowledge <id>
```

Parameters

Name	Description	Type
id	String	String

alarm unacknowledge

Description

```
Unacknowledge an active alarm
```

Syntax

```
alarm unacknowledge <id>
```

Parameters

Name	Description	Type
id	String	String

alias

Description

```
Add dropcounter alias
```

Parent Commands (Modes)

```
dropcounters <counter-name>
```

Syntax

```
alias <alias-str>
no alias
```

Parameters

Name	Description	Type
alias-str	String(Max: 24 characters)	String

allowas-in

Description

This command allows BGP neighbor to accept as-path with it's own AS number present in it.

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
allowas-in { [ <value> ] | [ origin ] } ]
no allowas-in { [ <value> ] | [ origin ] } ]
```

Parameters

Name	Description	Type
value		Integer

Usage Guidelines

Accepting own AS in an as-path usually results in AS loop. But sometimes, users add AS number to influence the BGP route selection process. This command enables user to control when a route with as-path containing own AS number should be accepted or not. The command also provides flexibility in terms of maximum number of occurrences of AS number in as-path.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# allowas-in 5
```

allowas-in

Description

This command allows neighbors in a BGP peer-group to accept as-path with it's own AS number present in it.

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
allowas-in { [ <value> ] | [ origin ] } ]
no allowas-in { [ <value> ] | [ origin ] } ]
```

Parameters

Name	Description	Type
value		Integer

Usage Guidelines

Accepting own AS in an as-path usually results in AS loop. But sometimes, users add AS number to influence the BGP route selection process. This command enables user to control when a route with as-path containing own AS number should be accepted or not. The command also provides flexibility in terms of maximum number of occurrences of AS number in as-path.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# remote-as 300
sonic(config-router-bgp Pg)# address-family ipv4 unicast
sonic(config-router-bgp-af)# allowas-in
```

allowas-in

Description

This command allows BGP neighbor to accept as-path with it's own AS number present in it.

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
allowas-in { [ <value> ] | [ origin ] } ]
no allowas-in { [ <value> ] | [ origin ] } ]
```

Parameters

Name	Description	Type
value		Integer

Usage Guidelines

Accepting own AS in an as-path usually results in AS loop. But sometimes, users add AS number to influence the BGP route selection process. This command enables user to control when a route with as-path containing own AS number should be accepted or not. The command also provides flexibility in terms of maximum number of occurrences of AS number in as-path.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)# allowas-in 5
```

allowas-in

Description

This command allows neighbors in a BGP peer-group to accept as-path with it's own AS number present in it.

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
allowas-in { [ <value> ] | [ origin ] } ]
no allowas-in { [ <value> ] | [ origin ] } ]
```

Parameters

Name	Description	Type
value		Integer

Usage Guidelines

Accepting own AS in an as-path usually results in AS loop. But sometimes, users add AS number to influence the BGP route selection process. This command enables user to control when a route with as-path containing own AS number should be accepted or not. The command also provides flexibility in terms of maximum number of occurrences of AS number in as-path.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# remote-as 300
sonic(config-router-bgp Pg)# address-family l2vpn evpn
sonic(config-router-bgp-af)# allowas-in
```

allowas-in

Description

Allow local AS number in as-path

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
allowas-in { [ <value> ] | [ origin ] } ]
no allowas-in { [ <value> ] | [ origin ] } ]
```

Parameters

Name	Description	Type
value		Integer

allowas-in

Description

Allow local AS number in as-path

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
allowas-in { [ <value> ] | [ origin ] } ]  
no allowas-in { [ <value> ] | [ origin ] } ]
```

Parameters

Name	Description	Type
value		Integer

always-compare-med

Description

Always compare the MED on routes, even when they were received from different neighbouring ASes. Setting this option makes the order of preference of routes more defined, and should eliminate MED induced oscillations.

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
always-compare-med  
no always-compare-med
```

Usage Guidelines

Use this command to instruct BGP to always compare MED values for routes even if they are from different ASes

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# always-compare-med
```

area

Description

Configures area parameters within an OSPFv2 router

Parent Commands (Modes)

router ospf [vrf <vrf-name>]

Syntax

```
area <areaid> { { [ authentication [ message-digest ] ] } | { [ default-cost <defaultcost> ] } | { [ filter-list { prefix { <prefixlistname> { in | out } } } ] } | { [ range { <rangeprefix> { { [ advertise { [ cost <metriccost> ] ] } ] } | { [ cost <metriccost> ] } ] | [ not-advertise ] | { [ substitute <rangenetworkprefix> ] ] } ] } | { [ stub [ no-summary ] ] } | { [ virtual-link { <vlinkip> { { [ authentication { [ null ] | [ message-digest ] ] } ] } | { [ authentication-key { <auth-key> [ encrypted ] ] } ] } | { [ message-digest-key { <keyid> { md5 { <md5key> [ encrypted ] } } } ] } | { [ dead-interval <deadinterval> ] } ] | { [ hello-interval <hellointerval> ] } | { [ retransmit-interval <retransmitinterval> ] } ] | { [ transmit-delay <transmitinterval> ] ] } ] } | { [ shortcut { default | disable | enable } ] } ] }
```

```
no area <areaid> { { [ authentication [ message-digest ] ] } | [ default-cost ] } | { [ filter-list { prefix { in | out } } ] } | { [ range { <rangeprefix> { { [ advertise [ cost ] ] } | [ cost ] | [ not-advertise ] | [ substitute ] ] } ] } } | { [ stub [ no-summary ] ] } | { [ virtual-link { <vlinkip> { { [ authentication { [ null ] | [ message-digest ] ] } ] } | { [ message-digest-key { <keyid> md5 } ] } | { [ authentication-key ] | [ dead-interval ] | [ hello-interval ] | [ retransmit-interval ] | [ transmit-delay ] ] } ] } ] | { [ shortcut ] } ] }
```

Parameters

Name	Description	Type
areaid	A.B.C.D or 0..4294967295	String
defaultcost		Integer
prefixlistname		String
rangeprefix	A.B.C.D/mask	String
metriccost		Integer
rangenetworkprefix	A.B.C.D/mask	String
vlinkip	A.B.C.D	String
auth-key		String
keyid		Integer
md5key		String
deadinterval		Integer
hellointerval		Integer
retransmitinterval		Integer
transmitinterval		Integer

Usage Guidelines

Use this command to configure area related parameters within an OSPFv2 router.

Below command options can be used

```
area <area-id>
    Configures an area within a router by specifying an area-id in positive
    integer or dotted format.

area <area-id> authentication [message-digest]
    Configures area level authentication mode. Authentocation mode can be clear
    text authentication or message-digest authentication.

area <area-id> default-cost <cost-value>
    Configures NSSA or stub-area summary default cost.

area <area-id>> filter-list prefix <prefix-list-name> in|out
    Configures prefix list for inter area prefix filtering. Inter area prefix
    propagation policies are configured using this command. Option 'in'
    is used for filetring incoming prefixes from the area and 'out' is used
    for filtering outgoing prefixes from the area.

area <area-id> range <ip-prefix> [ cost <cost-value> | advertise cost <cost-value> |
    not-advertise | substitute <ip-prefix> ]
    Configures address ranges for inter area address propagation. Inter area
    prefix propagation policies can also be configured using this command.

This command enables to configure prefix advertising rules directly without
using andy prefix list. Cost of an advertized prefix can be modified using
this command. Similarly a advertized prefix can be substituted by another prefix.

area <area-id> stub [ no-summary ]
    Configures area to be stub area with stub options.

area <area-id> shortcut enable|disable|default
    Configures area shortcut options.

area <area-id> virtual-link <remote-router-id>
area <area-id> virtual-link <remote-router-id> authentication [ null | message-digest ]
area <area-id> virtual-link <remote-router-id> authentication-key <key> [ encrpted ]
area <area-id> virtual-link <remote-router-id> message-digest-key <key-id> md5 <key> [
    encrpted ]
area <area-id> virtual-link <remote-router-id> dead-interval <time-value>
area <area-id> virtual-link <remote-router-id> hello-interval <time-value>
area <area-id> virtual-link <remote-router-id> retransmit-interval <time-value>
area <area-id> virtual-link <remote-router-id> transmit-delay <time-value>

    Configures virtual link and its parameters in an area. Virtual link configurations
    are allowed on non-backbone area. Virtual links can have clear text password,
    message-digest based passwords or no password configured at all. When clear text
    and message digest passsword is configured, corresponding authentication-key or
    message-digest-key parameters must be configured. Timer parameters can also be
    configured for any Virtual links. Authentication key or password will be saved
    in encrypted form in configuration. User shall always provide actual password
    while configuring authentication keys. It is not recomended to use encrypted
    option of authentication key.
```

Examples

```
sonic-cli(config-router-ospf)# area 19
sonic-cli(config-router-ospf)# area 19.1.1.19
sonic-cli(config-router-ospf)# area 19 authentication
sonic-cli(config-router-ospf)# area 19 authentication message-digest
sonic-cli(config-router-ospf)# area 19 filter-list prefix plist-area10_in in
sonic-cli(config-router-ospf)# area 19 filter-list prefix plist-area10_out out
sonic-cli(config-router-ospf)# area 19 range 10.1.1.2/24
sonic-cli(config-router-ospf)# area 19 range 10.1.1.0/24 cost 48
sonic-cli(config-router-ospf)# area 19 range 10.2.2.0/24 not-advertise
sonic-cli(config-router-ospf)# area 19 range 10.3.3.0/24 substitute 192.3.3.0/24
```

```

sonic-cli(config-router-ospf)# area 19 stub
sonic-cli(config-router-ospf)# area 19 stub no summary
sonic-cli(config-router-ospf)# area 19 shortcut enable
sonic-cli(config-router-ospf)# area 19 virtual-link 1.1.1.9
sonic-cli(config-router-ospf)# area 19 virtual-link 1.1.1.9 authentication
sonic-cli(config-router-ospf)# area 19 virtual-link 1.1.1.9 authentication null
sonic-cli(config-router-ospf)# area 19 virtual-link 1.1.1.9 authentication message-digest
sonic-cli(config-router-ospf)# area 19 virtual-link 1.1.1.9 authentication-key password
sonic-cli(config-router-ospf)# area 19 virtual-link 1.1.1.9 message-digest-key 19 md5
    md5password
sonic-cli(config-router-ospf)# area 19 virtual-link 1.1.1.9 dead-interval 60
sonic-cli(config-router-ospf)# area 19 virtual-link 1.1.1.9 hello-interval 20
sonic-cli(config-router-ospf)# area 19 virtual-link 1.1.1.9 retransmit-interval 15
sonic-cli(config-router-ospf)# area 19 virtual-link 1.1.1.9 transmit-delay 10

```

Features this CLI belongs to

- OSPFv2 ## area ### Description

Configure OSPFv3 area parameters

Parent Commands (Modes)

`router ospf6 [vrf <vrf-name>]`

Syntax

```

area <areaid> { { [ filter-list { prefix { <prefixlistname> { in | out } } } ] } | { [ stub [
    no-summary ] ] } }
no area <areaid> { { [ filter-list { prefix { in | out } } ] } | { [ stub [ no-summary ] ] } }

```

Parameters

Name	Description	Type
areaid	A.B.C.D or 0..4294967295	String
prefixlistname	String	String

as-override

Description

This command instructs BGP to override AS Numbers in outbound updates if aspath equals remote-as

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
as-override  
no as-override
```

Usage Guidelines

```
Use this command to override the as number in outbound updates
```

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 20.20.20.2  
sonic(config-router-bgp-neighbor)# remote-as 300  
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast  
sonic(config-router-bgp-neighbor-af)# as-override
```

as-override

Description

```
This command instructs BGP to override AS Numbers in outbound updates if  
aspath equals remote-as
```

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
as-override  
no as-override
```

Usage Guidelines

```
Use this command to override the as number in outbound updates
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# address-family ipv4 unicast
sonic(config-router-bgp Pg-af)# as-override
```

as-override

Description

```
Override ASNs in outbound updates if aspath equals remote-as
```

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
as-override
no as-override
```

as-override

Description

```
Override ASNs in outbound updates if aspath equals remote-as
```

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
as-override
no as-override
```

attribute-unchanged

Description

```
This command instructs BGP to propagate route attributes (as-path,
next-hop, med) unchanged to this neighbor
```

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
attribute-unchanged [ as-path ] [ med ] [ next-hop ]  
no attribute-unchanged
```

Usage Guidelines

Use this command to propagate BGP route attributes unchanged to this neighbor. User can control which attributes (as-path, next-hop, med) will be propagated unchanged.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 20.20.20.2  
sonic(config-router-bgp-neighbor)# remote-as 300  
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast  
sonic(config-router-bgp-neighbor-af)# attribute-unchanged as-path next-hop
```

attribute-unchanged

Description

This command instructs BGP to propagate route attributes (as-path, next-hop, med) unchanged to neighbors in a peer-group

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
attribute-unchanged [ as-path ] [ med ] [ next-hop ]  
no attribute-unchanged
```

Usage Guidelines

Use this command to propagate BGP route attributes unchanged to neighbors in a peer-group. User can control which attributes (as-path, next-hop, med) will be propagated unchanged.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# address-family ipv4 unicast
sonic(config-router-bgp Pg-af)# attribute-unchanged as-path next-hop
```

attribute-unchanged

Description

This command instructs BGP to propagate route attributes (as-path, next-hop, med) unchanged to this neighbor

Parent Commands (Modes)

```
address-family 12vpn evpn
```

Syntax

```
attribute-unchanged [ as-path ] [ med ] [ next-hop ]
no attribute-unchanged
```

Usage Guidelines

Use this command to propagate BGP route attributes unchanged to this neighbor. User can control which attributes (as-path, next-hop, med) will be propagated unchanged.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family 12vpn evpn
sonic(config-router-bgp-neighbor-af)# attribute-unchanged as-path next-hop
```

attribute-unchanged

Description

This command instructs BGP to propagate route attributes (as-path, next-hop, med) unchanged to neighbors in a peer-group

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
attribute-unchanged [ as-path ] [ med ] [ next-hop ]  
no attribute-unchanged
```

Usage Guidelines

Use this command to propagate BGP route attributes unchanged to neighbors in a peer-group. User can control which attributes (as-path, next-hop, med) will be propagated unchanged.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Ext  
sonic(config-router-bgp Pg)# address-family l2vpn evpn  
sonic(config-router-bgp Pg-af)# attribute-unchanged as-path next-hop
```

attribute-unchanged

Description

BGP attribute is propagated unchanged to this neighbor

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
attribute-unchanged [ as-path ] [ med ] [ next-hop ]  
no attribute-unchanged
```

attribute-unchanged

Description

BGP attribute is propagated unchanged to this neighbor

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
attribute-unchanged [ as-path ] [ med ] [ next-hop ]
no attribute-unchanged
```

auth-type

Description

This command specifies the type of authorization that the device must use for RADIUS clients.
The client must match the configured attributes for authorization.

Parent Commands (Modes)

```
aaa server radius dynamic-author
```

Syntax

```
auth-type { all | any | session-key }
no auth-type
```

Usage Guidelines

This command configures the accepted authorization types for dynamic RADIUS clients.

Examples

```
sonic-cl(i(config)# aaa server radius dynamic-author
sonic-cl(i(config-radius-da)# auth-type session-key
```

authentication command bounce-port ignore

Description

This command configures the device to ignore a RADIUS server bounce-host-port command. The bounce-host-port command causes a host to flap the link on an authentication port. The link flap causes DHCP renegotiation from one or more hosts connected to this port.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
authentication command bounce-port ignore  
no authentication command bounce-port ignore
```

Usage Guidelines

This command configures the device to ignore a RADIUS server bounce-host-port command.

Examples

```
sonic-cl(i(config)# authentication command bounce-port ignore
```

authentication command disable-port ignore

Description

This command configures the device to ignore a RADIUS server disable-host-port command. The disable-host-port command puts the host port to D-Disabled state with reason as "coa disabled".

The D-Disabled port with reason as ?coa disabled? can be re-enabled either if the auto recovery cause is enabled for coa after the expiry of the auto recovery timer or manually by the administrator by not shutting down the port.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
authentication command disable-port ignore  
no authentication command disable-port ignore
```

Usage Guidelines

This command configures the device to ignore a RADIUS server disable-host-port command.

Examples

```
sonic-cl(i(config)# authentication command disable-port ignore
```

authentication dynamic-vlan enable

Description

This command enables the switch to create VLANs dynamically when a RADIUS-assigned VLAN does not exist in the switch.

Parent Commands (Modes)

configure terminal

Syntax

```
authentication dynamic-vlan enable  
no authentication dynamic-vlan enable
```

Usage Guidelines

Using this command user can enable dynamic vlan creation configuration when a RADIUS-assigned VLAN does not exist in the switch.

Examples

```
sonic(config)# authentication dynamic-vlan enable
```

authentication event fail action authorize vlan

Description

This command configures the unauthenticated VLAN associated with the specified interface or range of interfaces

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
authentication event fail action authorize vlan <vlan-id>  
no authentication event fail action authorize vlan
```

Parameters

Name	Description	Type
vlan-id		Integer

Usage Guidelines

This command allows configuring unauthenticated VLAN associated with the specified interface or range of interfaces.

Examples

```
sonic-cl(i(conf-if-Ethernet0)# authentication event fail action authorize vlan {vlan-id}
```

authentication event fail action authorize vlan

Description

This command configures the unauthenticated VLAN associated with the specified interface or range of interfaces

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
authentication event fail action authorize vlan <vlan-id>
no authentication event fail action authorize vlan
```

Parameters

Name	Description	Type
vlan-id		Integer

Usage Guidelines

This command allows configuring unauthenticated VLAN associated with the specified interface or range of interfaces.

Examples

```
sonic-cli(conf-if-Ethernet0)# authentication event fail action authorize vlan {vlan-id}
```

authentication event fail retry

Description

This command configures the number of times authentication may be reattempted by the client before a port moves to the authentication fail VLAN.
reattempts range is 1 to 5

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
authentication event fail retry <max-attempts>
no authentication event fail retry
```

Parameters

Name	Description	Type
max-attempts	authentication max-attempts	Integer

Usage Guidelines

This command allows configuring the number of times authentication may be reattempted by the client before a port moves to the authentication fail VLAN.

Examples

```
sonic-cli(conf-if-Ethernet0)# authentication event fail retry {max-attempts}
```

authentication event fail retry

Description

This command configures the number of times authentication may be reattempted by the client before a port moves to the authentication fail VLAN.
reattempts range is 1 to 5

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
authentication event fail retry <max-attempts>
no authentication event fail retry
```

Parameters

Name	Description	Type
max-attempts	authentication max-attempts	Integer

Usage Guidelines

This command allows configuring the number of times authentication may be reattempted by the client before a port moves to the authentication fail VLAN.

Examples

```
sonic-clt(conf-if-Ethernet0)# authentication event fail retry {max-attempts}
```

authentication event no-response action authorize vlan

Description

This command configures VLAN as guest vlan on an interface or a range of interfaces. The range is 1 to the maximum VLAN ID supported by the platform or alive server actions. By default, the guest VLAN is 0, i.e. invalid and is not operational.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
authentication event no-response action authorize vlan <vlan-id>
no authentication event no-response action authorize vlan
```

Parameters

Name	Description	Type
vlan-id		Integer

Usage Guidelines

This command allows configuring VLAN as guest vlan on an interface or a range of interfaces.

Examples

```
sonic-cl(i(conf-if-Ethernet0)# authentication event no-response action authorize vlan {vlan-id}
```

authentication event no-response action authorize vlan

Description

This command configures VLAN as guest vlan on an interface or a range of interfaces. The range is 1 to the maximum VLAN ID supported by the platform or alive server actions. By default, the guest VLAN is 0, i.e. invalid and is not operational.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
authentication event no-response action authorize vlan <vlan-id>
no authentication event no-response action authorize vlan
```

Parameters

Name	Description	Type
vlan-id		Integer

Usage Guidelines

This command allows configuring VLAN as guest vlan on an interface or a range of interfaces.

Examples

```
sonic-cli(conf-if-Ethernet0)# authentication event no-response action authorize vlan {vlan-id}
```

authentication host-mode

Description

This command configures the host mode of a port.
configuration on the interface mode takes precedence over the global configuration of this parameter.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
authentication host-mode { single-host | multi-auth | multi-domain | multi-host }  
no authentication host-mode
```

Usage Guidelines

By using this command user can configure authentication host mode on an interface or range of interfaces.

Examples

```
sonic(conf-if-Ethernet0)# authentication host-mode all { multi-auth | multi-domain | multi-host  
| single-host }
```

authentication host-mode

Description

This command configures the host mode of a port.
configuration on the interface mode takes precedence over the global configuration of this parameter.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
authentication host-mode { single-host | multi-auth | multi-domain | multi-host }
no authentication host-mode
```

Usage Guidelines

By using this command user can configure authentication host mode on an interface or range of interfaces.

Examples

```
sonic(conf-if-Ethernet0)# authentication host-mode all { multi-auth | multi-domain | multi-host
| single-host }
```

authentication max-users

Description

This command sets the maximum number of clients supported on an interface or range of interfaces when multi-authentication host mode is enabled on the port. The count value is in the range 1 - 48.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
authentication max-users <maxUsers>
no authentication max-users
```

Parameters

Name	Description	Type
maxUsers		Integer

Usage Guidelines

This command allows configuring the maximum number of clients supported on an interface or range of interfaces. The count value is in the range 1 - 48.

Examples

```
sonic-cli(conf-if-Eth1/1/1)# authentication max-users 37
```

authentication max-users

Description

This command sets the maximum number of clients supported on an interface or range of interfaces when multi-authentication host mode is enabled on the port. The count value is in the range 1 - 48.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
authentication max-users <maxUsers>
no authentication max-users
```

Parameters

Name	Description	Type
maxUsers		Integer

Usage Guidelines

This command allows configuring the maximum number of clients supported on an interface or range of interfaces. The count value is in the range 1 - 48.

Examples

```
sonic-cli(conf-if-Eth1/1/1)# authentication max-users 37
```

authentication monitor

Description

This command enables the Authentication monitor mode on the switch. The purpose of Monitor mode is to help troubleshoot port-based authentication configuration issues without disrupting network access for hosts connected to the switch. In Monitor mode, a host is granted network access to an authentication enforced port even if it fails the authentication process. The results of the process are logged for diagnostic purposes.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
authentication monitor  
no authentication monitor
```

Usage Guidelines

By using this command user can enable monitor mode in the system.

Examples

```
sonic(config)# authentication monitor
```

authentication open

Description

This command configures Open Authentication mode on the port.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
authentication open  
no authentication open
```

Usage Guidelines

By using this command user can enable Open Authentication mode on the port.

Examples

```
sonic-cl(i(conf-if-Ethernet0)# authentication open
```

authentication open

Description

This command configures Open Authentication mode on the port.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
authentication open  
no authentication open
```

Usage Guidelines

By using this command user can enable Open Authentication mode on the port.

Examples

```
sonic-cl(i(conf-if-Ethernet0)# authentication open
```

authentication order

Description

This command sets the order of authentication methods used on the interface.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
authentication order { { dot1x [ mab ] } | { mab [ dot1x ] } }  
no authentication order
```

Usage Guidelines

This command allows configuring order of authentication methods used on the interface.

Examples

```
sonic-cli(conf-if-Ethernet0)# authentication order dot1x mab
```

authentication order

Description

This command sets the order of authentication methods used on the interface.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
authentication order { { dot1x [ mab ] } | { mab [ dot1x ] } }
no authentication order
```

Usage Guidelines

This command allows configuring order of authentication methods used on the interface.

Examples

```
sonic-cli(conf-if-Ethernet0)# authentication order dot1x mab
```

authentication periodic

Description

This command enables periodic reauthentication of the supplicant for the specified interface or range of interfaces.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
authentication periodic  
no authentication periodic
```

Usage Guidelines

By using this command user can configure periodic reauthentication of the supplicant for the specified interface or range of interfaces.

Examples

```
sonic-cli(conf-if-Ethernet0)# authentication periodic
```

authentication periodic

Description

This command enables periodic reauthentication of the supplicant for the specified interface or range of interfaces.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
authentication periodic  
no authentication periodic
```

Usage Guidelines

By using this command user can configure periodic reauthentication of the supplicant for the specified interface or range of interfaces.

Examples

```
sonic-cli(conf-if-Ethernet0)# authentication periodic
```

authentication port-control

Description

This command configures the authentication mode to use on the specified interface or range of interfaces.

configuration on the interface mode takes precedence over the global configuration of this parameter.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
authentication port-control { auto | force-authorized | force-unauthorized }
no authentication port-control
```

Usage Guidelines

By using this command user can configure authentication port-control on the specified interface or range of interfaces.

Examples

```
sonic(conf-if-Ethernet0)# authentication port-control all {auto | force-authorized | force-unauthorized}
```

authentication port-control

Description

This command configures the authentication mode to use on the specified interface or range of interfaces.

configuration on the interface mode takes precedence over the global configuration of this parameter.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
authentication port-control { auto | force-authorized | force-unauthorized }
no authentication port-control
```

Usage Guidelines

By using this command user can configure authentication port-control on the specified interface or range of interfaces.

Examples

```
sonic(conf-if-Ethernet0)# authentication port-control all {auto | force-authorized | force-unauthorized}
```

authentication priority

Description

The priority for the authentication methods used on the interface.

Parent Commands (Modes)

`interface <phy-if-name>`

Syntax

```
authentication priority { { dot1x [ mab ] } | { mab [ dot1x ] } }
no authentication priority
```

Usage Guidelines

This command allows configuring priority for the authentication methods used on the interface.

Examples

```
sonic-clt(conf-if-Ethernet0)# authentication priority dot1x mab
```

authentication priority

Description

The priority for the authentication methods used on the interface.

Parent Commands (Modes)

`interface range iface_range_num`

Syntax

```
authentication priority { { dot1x [ mab ] } | { mab [ dot1x ] } }
no authentication priority
```

Usage Guidelines

This command allows configuring priority for the authentication methods used on the interface.

Examples

```
sonic-cl i(conf-if-Ethernet0) # authentication priority dot1x mab
```

authentication rest

Description

```
rest authentication modes
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
authentication rest <client-auth>
```

Parameters

Name	Description	Type
client-auth	Authentication Types	String

authentication telemetry

Description

```
telemetry authentication modes
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
authentication telemetry <client-auth>
```

Parameters

Name	Description	Type
client-auth	Authentication Types	String

authentication timer reauthenticate

Description

This command is used to configure the period of time after which the Authenticator attempts to reauthenticate a supplicant on the port.

This command also provides an option to specify re-authentication time out value from the server (ex. Radius).

When server option is selected, the server supplied Session time out and Session Termination-action are used by Authenticator to

reauthenticate a supplicant on the port . By default server option is enabled. The reauthenticate seconds value range is 1 to 65535.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
authentication timer reauthenticate { server | <time-period> }  
no authentication timer reauthenticate
```

Parameters

Name	Description	Type
time-period		Integer

Usage Guidelines

This command allows configuring the period of time after which the Authenticator attempts to reauthenticate a supplicant on the port.

Examples

```
sonic-cl i(conf-if-Eth1/1/1) # authentication timer reauthenticate { seconds | server}
```

authentication timer reauthenticate

Description

This command is used to configure the period of time after which the Authenticator attempts to reauthenticate a supplicant on the port.

This command also provides an option to specify re-authentication time out value from the server (ex. Radius).

When server option is selected, the server supplied Session time out and Session Termination-action are used by Authenticator to

reauthenticate a supplicant on the port . By default server option is enabled. The reauthenticate seconds value range is 1 to 65535.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
authentication timer reauthenticate { server | <time-period> }
no authentication timer reauthenticate
```

Parameters

Name	Description	Type
time-period		Integer

Usage Guidelines

This command allows configuring the period of time after which the Authenticator attempts to reauthenticate a supplicant on the port.

Examples

```
sonic-clt(conf-if-Eth1/1/1)# authentication timer reauthenticate { seconds | server}
```

authorization network radius

Description

This command enables the configuration by which RADIUS can assign a VLAN to a client.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
authorization network radius  
no authorization network radius
```

Usage Guidelines

This command enables the configuration by which RADIUS can assign a VLAN to a client.

Examples

```
sonic(config)# authorization network radius
```

auto-cost

Description

Configures interface auto cost reference bandwidth

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
auto-cost reference-bandwidth <ref-bandwidth>  
no auto-cost reference-bandwidth
```

Parameters

Name	Description	Type
ref-bandwidth	1-4294967	Integer

Usage Guidelines

Use this command to configure interface auto cost reference bandwidth. Whenever interface cost is not configured explicitly, reference bandwidth value will be used to calculate the interface cost.

Examples

```
sonic-cli(config-router-ospf)# auto-cost reference-bandwidth 10000
```

Features this CLI belongs to

- OSPFv2 ## auto-cost ### Description

```
Configures interface auto cost reference bandwidth
```

Parent Commands (Modes)

```
router ospf6 [ vrf <vrf-name> ]
```

Syntax

```
auto-cost reference-bandwidth <ref-bandwidth>
no auto-cost reference-bandwidth
```

Parameters

Name	Description	Type
ref-bandwidth	1-4294967	Integer

Usage Guidelines

Use this command to configure interface auto cost reference bandwidth. Whenever interface cost is not configured explicitly, refrence bandwidth value will be used to calculate the interface cost.

Examples

```
sonic-cli(config-router-ospf6)# auto-cost reference-bandwidth 10000
```

Features this CLI belongs to

- OSPFv3 ## autoneg ### Description

```
Configure autoneg
```

Parent Commands (Modes)

```
interface Management <mgmt-if-id>
```

Syntax

```
autoneg
no autoneg
```

autort

Description

```
This command enables automatic derivation of route-distinguisher and route-targets
```

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
autort rfc8365-compatible
no autort rfc8365-compatible
```

Usage Guidelines

```
[no] autort {autort-method}
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# autort rfc8365-compatible
```

autostate

Description

```
Configure Autostate on Vlan interface.
Enable - VLAN Operstatus is dependent on VLAN Member operstatus and VRF-VNI mappings.
Disable - VLAN Operstatus is set as Up once created.
```

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
autostate  
no autostate
```

Usage Guidelines

```
sonic-cl(i)conf-if-Vlan10# autostate
```

Examples

```
sonic-cl(i)config# interface Vlan 10  
sonic-cl(i)conf-if-Vlan10# autostate
```

autostate

Description

```
Configure autostate setting
```

Parent Commands (Modes)

```
interface range create vlan_range_num  
interface range vlan_range_num
```

Syntax

```
autostate  
no autostate
```

bestpath as-path confed

Description

This command specifies that the length of confederation path sets and sequences should be taken into account during the BGP best path decision process.

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
bestpath as-path confed  
no bestpath as-path confed
```

Usage Guidelines

Use this command to consider confederation set and sequence path length for best-path selection process

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# bestpath as-path confed
```

bestpath as-path ignore

Description

This command influences best-path selection algorithm by not comparing as-path attribute

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

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

Usage Guidelines

Use this command to ignore as-path comparison during best-path selection process.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# bestpath as-path ignore
```

bestpath as-path multipath-relax

Description

This command specifies that BGP decision process should consider paths of equal AS_PATH length candidates for multipath computation. Without the knob, the entire AS_PATH must match for multipath computation.

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
bestpath as-path multipath-relax [ as-set ]
no bestpath as-path multipath-relax [ as-set ]
```

Usage Guidelines

Use this command to ignore as-path check for paths for the same prefix thereby making all the paths equal irrespective of their as-path

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# bestpath as-path multipath-relax
```

bestpath compare-routerid

Description

This command influences best-path selection algorithm by comparing router-ids for identical eBGP routes

Ensure that when comparing routes where both are equal on most metrics, including local-pref, AS_PATH length, IGP cost, MED, that the tie is broken based on router-ID.

If this option is enabled, then the already-selected check, where already selected eBGP routes are preferred, is skipped.

If a route has an ORIGINATOR_ID attribute because it has been reflected, that ORIGINATOR_ID will be used. Otherwise, the router-ID of the peer the route was received from will be used.

The advantage of this is that the route-selection (at this point) will be more deterministic. The disadvantage is that a few or even one lowest-ID router may attract all traffic to otherwise-equal paths because of this check. It may increase the possibility of MED or IGP oscillation, unless other measures were taken to avoid these. The exact behaviour will be sensitive to the iBGP and reflection topology.

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

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

Usage Guidelines

Use this command to compare router-ids as tie-breaker for identical eBGP paths

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# bestpath compare-routerid
```

bestpath med

Description

This command influences best-path selection algorithm by how missing MEDs are treated as well as whether MEDs should be compared for confederation paths.

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
bestpath med { { missing-as-worst [ confed ] } | { confed [ missing-as-worst ] } }  
no bestpath med { { [ missing-as-worst [ confed ] ] } | { [ confed [ missing-as-worst ] ] } } ]
```

Usage Guidelines

Use this command to consider MED for confederation paths for best-path selection process. Also, if MED is missing, should it be considered as worst MED.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# bestpath med missing-as-worst confed
```

bfd

Description

Configure BFD peers

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
bfd
```

bfd

Description

This command enables BFD liveliness check for a BGP neighbor

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
bfd
no bfd
```

Usage Guidelines

Use this command to enable BFD for a BGP neighbor.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# bfd
```

bfd

Description

This command enables BFD liveliness check for BGP neighbors in a peer-group

Parent Commands (Modes)

peer-group <template-str>

Syntax

```
bfd  
no bfd
```

Usage Guidelines

Use this command to enable BFD for a BGP peer-group.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Ext  
sonic(config-router-bgp-pg)# bfd
```

bfd check-control-plane-failure

Description

This command links Data Plane status to the BGP control

Parent Commands (Modes)

neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }

Syntax

```
bfd check-control-plane-failure  
no bfd check-control-plane-failure
```

Usage Guidelines

Use this command to links Data Plane status to the BGP control plane

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# bfd check-control-plane-failure
```

bfd check-control-plane-failure

Description

This command links Data Plane status to the BGP control

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
bfd check-control-plane-failure
no bfd check-control-plane-failure
```

Usage Guidelines

check-control-plane-failure links Data Plane status to the BGP control plane

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# bfd check-control-plane-failure
```

bfd profile

Description

This command enables BFD profile for a BGP neighbor

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
bfd profile <profilename>
no bfd profile
```

Parameters

Name	Description	Type
profilename	WORD	String

Usage Guidelines

Use this command to enable BFD profile for a BGP neighbor.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# bfd profile bgp
```

bfd profile

Description

This command enables BFD profile for BGP neighbors in a peer-group

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
bfd profile <profilename>
no bfd profile
```

Parameters

Name	Description	Type
profilename	WORD	String

Usage Guidelines

Use this command to enable BFD profile for a BGP peer-group.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp-pxg)# bfd profile bgp
```

bgp as-path-list

Description

This command creates BGP AS Path list

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
bgp as-path-list <as-path-list-name> { { deny <regx-id> } | { permit <regx-id> } }
no bgp as-path-list <as-path-list-name> { { [ deny <regx-id> ] } | { [ permit <regx-id> ] } }
```

Parameters

Name	Description	Type
as-path-list-name	WORD	String
regx-id	String	String

Usage Guidelines

Use this command to create BGP AS Path list. User can enter a regular expression of AS Paths that provides flexible and powerful match support. The command also provides any and all options to allow matching all or any entry in as-path-list

Examples

```
sonic# configure terminal
sonic(config)# bgp as-path-list asp_private permit ^65000.*6510565109$
sonic(config)# bgp as-path-list asp_private deny 65107.*65200
```

bgp community-list

Description

This command creates BGP community list

Parent Commands (Modes)

configure terminal

Syntax

```
bgp community-list { { standard { <community-list-name> { deny | permit } { { <aann> [ local-as
] [ no-advertise ] [ no-export ] [ no-peer ] { [ any ] | [ all ] ] } } | { local-as [
<aann> ] [ no-advertise ] [ no-export ] [ no-peer ] { [ any ] | [ all ] ] } } | {
no-advertise [ <aann> ] [ local-as ] [ no-export ] [ no-peer ] { [ any ] | [ all ] ] } } |
{ no-export [ <aann> ] [ local-as ] [ no-advertise ] [ no-peer ] { [ any ] | [ all ] ] } } |
{ no-peer [ <aann> ] [ local-as ] [ no-advertise ] [ no-export ] { [ any ] | [ all ] ] } }
} } } | { expanded { <community-list-name> { deny | permit } { <line> { [ any ] | [ all ]
] } } } }
```

```
no bgp community-list { { standard { <community-list-name> { { [ deny { <aann> | local-as |
no-advertise | no-export | no-peer } ] } } | { [ permit { <aann> | local-as | no-advertise |
no-export | no-peer } ] ] } } } | { expanded { <community-list-name> { { [ deny <line>
] | { [ permit <line> ] ] } } } }
```

Parameters

Name	Description	Type
community-list-name	WORD	String
aann	AA:NN	String
line	Line	String

Usage Guidelines

Use this command to create BGP community list. The command provides options to create expanded or standard community and accepts community in AA:NN, IP:NN and well-known communities format. This command also provides any and all constructs to enable user to design community filters with clause match any or all. For expanded community, user can specify a regular expression of communities.

Examples

```
sonic# configure terminal
sonic(config)# bgp community-list standard CommList_RT 100:200
sonic(config)# bgp community-list standard CommList_RT no-export
sonic(config)# bgp community-list standard CommList_RT no-peer
sonic(config)# bgp community-list standard CommList_RT 65100:3456
```

bgp extcommunity-list

Description

This command creates BGP extended community list entries

Parent Commands (Modes)

configure terminal

Syntax

```
bgp extcommunity-list { { standard { <extcommunity-list-name> { deny | permit } { { rt { <aa> | <ipaddrnn> } { [ any ] | [ all ] ] } } | { soo { <aa> | <ipaddrnn> } { [ any ] | [ all ] ] } } } | { expanded { <extcommunity-list-name> { deny | permit } { <line> { [ any ] | [ all ] ] } } } } }
no bgp extcommunity-list { { standard { <extcommunity-list-name> { { [ deny { { rt { <aa> | <ipaddrnn> } } | { soo { <aa> | <ipaddrnn> } } } ] } | { [ permit { { rt { <aa> | <ipaddrnn> } } | { soo { <aa> | <ipaddrnn> } } ] } } } | { expanded { <extcommunity-list-name> { { [ deny <line> ] } | { [ permit <line> ] } } } } }
```

Parameters

Name	Description	Type
extcommunity-list-name	WORD	String
aa	AA:NN	String
ipaddrnn	A.B.C.D:[1..65535]	String
line	Line	String

Usage Guidelines

Use this command to create BGP extended community list entries. The command provides options to create expanded or standard extended community list entries. For standard extended community, user can create rt or soo type of communities and command will accept communities in AA:NN or IP:NN formats. For expanded extended community, the command will accept a regular expression of communities, which is very flexible and powerful for matching communities in routes. This command also provides option for matching all or any extended communities.

Examples

```
sonic# configure terminal
sonic(config)# bgp extcommunity-list standard ExtComm_AllowInt rt 19.32.56.167:65011 all
sonic(config)# bgp extcommunity-list standard ExtComm_AllowInt rt 31.67.182.214:3001 all
sonic(config)# bgp extcommunity-list standard ExtComm_AllowInt soo 4001:65010 all
sonic(config)# bgp extcommunity-list standard ExtComm_AllowInt soo 98.13.175.21:65101 all
```

binding

Description

Create binding between an ACL and a NAT pool

Parent Commands (Modes)

nat

Syntax

```
binding <binding-name> <pool-name> [ <acl-name> ] [ <natType> ] [ twice-nat-id
    <twice-nat-id-value> ]
no binding <bind-name>
```

Parameters

Name	Description	Type
binding-name	String	String
pool-name	String	String
acl-name	String	String
natType	NAT type	Select [snat dnat]
twice-nat-id-value		Integer

buffer init

Description

QoS buffer init populates the QoS buffer related tables in the Redis ConfigDB by default, the size and xoff values vary based on platform capabilities.

Parent Commands (Modes)

configure terminal

Syntax

```
buffer init  
no buffer init
```

Usage Guidelines

Use this command to configure QoS buffer related entries to default settings.

Examples

```
sonic# configure terminal  
sonic(config)# buffer init
```

buffer pool

Description

This command configures shared head room size for QoS buffer pool ingress_lossless_pool.

Parent Commands (Modes)

configure terminal

Syntax

```
buffer pool <pool-name> { { type { <type-value> { size <size-value> } } } { [  
    shared-headroom-size <shared-headroom-size-value> ] } }  
no buffer pool <name>
```

Parameters

Name	Description	Type
pool-name	String	String
type-value	Direction	Select [ingress(INGRESS) egress(EGRESS)]
size-value		Integer
shared-headroom-size-value		Integer

Usage Guidelines

```
sonic(config)# buffer pool ingress_lossless_pool shared-headroom-size SHARED-HEADROOM-SIZE
```

Examples

```
sonic# configure terminal
sonic(config)# buffer pool ingress_lossless_pool type ingress size 41943040
    shared-headroom-size 12582912
```

buffer priority-group

Description

This command maps QoS priority group (ingress) with QoS buffer profile.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
buffer priority-group <pg-value-range> <profile-name>
no buffer priority-group <pg-value-range>
```

Parameters

Name	Description	Type
pg-value-range		String
profile-name	String	String

Usage Guidelines

```
sonic(conf-if-Ethernet8)# buffer priority-group PG-ID-LIST PROFILE-NAME
```

Examples

```
sonic# configure terminal
sonic(config)# interface Ethernet 8
sonic(conf-if-Ethernet8)# buffer priority-group 2-6 profile_1
```

buffer profile

Description

This command creates the buffer profile for priority groups (ingress) and queue (egress) QoS operations.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
buffer profile <name> { <pool-name> { dynamic-threshold <dynamic-threshold-value> } { [ pause-threshold <pause-threshold-value> ] } }
no buffer profile <name>
```

Parameters

Name	Description	Type
name	String	String
pool-name	String	String
dynamic-threshold-value		Integer
pause-threshold-value		Integer

Usage Guidelines

```
sonic(config)# buffer profile PROFILE-NAME POOL-NAME dynamic-threshold DYNAMIC-THRESHOLD-VALUE
    pause pause-threshold PAUSE-THRESHOLD-VALUE
```

Examples

```
sonic# configure terminal
sonic(config)# buffer profile ingress_lossless_profile ingress_lossless_pool dynamic-threshold
    -6 pause-threshold 441600
sonic(config)# buffer profile profile_1 pool_1 dynamic-threshold -2
```

buffer queue

Description

This command maps QoS queue (egress) with QoS buffer profile.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
buffer queue <queue-value-range> <profile-name>
no buffer queue <queue-value-range>
```

Parameters

Name	Description	Type
queue-value-range		String
profile-name	String	String

Usage Guidelines

```
sonic(conf-if-Ethernet8)# buffer queue Q-ID-LIST PROFILE-NAME
```

Examples

```
sonic# configure terminal
sonic(config)# interface Ethernet 8
sonic(conf-if-Ethernet8)# buffer queue 2-6 profile_1
```

call

Description

Jump to another route-map after match_set

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
call <match-call>
no call
```

Parameters

Name	Description	Type
match-call	WORD	String

capability

Description

```
Configures OSPFv2 Opaque LSA Capability
```

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
capability opaque
no capability opaque
```

Usage Guidelines

```
Use this command to configure Opaque LSA Capability.
```

Examples

```
sonic-clia(config-router-ospf)# capability opaque
```

Features this CLI belongs to

- OSPFv2 ## capability dynamic ### Description

This command allows bgp to advertise dynamic capability to a neighbor

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
capability dynamic  
no capability dynamic
```

Usage Guidelines

Use this command to turn on dynamic capability for a BGP neighbor

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 30.30.30.3  
sonic(config-router-bgp-neighbor)# capability dynamic
```

capability dynamic

Description

This command allows bgp to advertise dynamic capability to neighbors in a peer-group

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
capability dynamic  
no capability dynamic
```

Usage Guidelines

Use this command to turn on dynamic capability for a BGP peer-group

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp- pg)# capability dynamic
```

capability extended-nexthop

Description

This command allows bgp to negotiate the extended-nexthop capability with it's peer. If you are peering over a v6 LL address then this capability is turned on automatically. If you are peering over a v6 Global Address then turning on this command will allow BGP to install v4 routes with v6 nexthops if you do not have v4 configured on interfaces.

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
capability extended-nexthop
no capability extended-nexthop
```

Usage Guidelines

Use this command to turn on extended-nexthop capability for a BGP neighbor

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# capability extended-nexthop
```

capability extended-nexthop

Description

This command allows bgp to negotiate the extended-nexthop capability with it's peer in a peer-group. If you are peering over a v6 LL address then this capability is turned on automatically. If you are peering over a v6 Global Address then turning on this command will allow BGP to install v4 routes with v6 nexthops if you do not have v4 configured on interfaces.

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
capability extended-nexthop  
no capability extended-nexthop
```

Usage Guidelines

Use this command to turn on extended-nexthop capability for a BGP peer-group

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Ext  
sonic(config-router-bgp Pg)# capability extended-nexthop
```

capability orf prefix-list

Description

This command enables BGP to advertise Outbound Route Filtering (ORF) capability to this neighbor

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
capability orf prefix-list { send | receive | both }  
no capability orf prefix-list { [ send ] | [ receive ] | [ both ] } ]
```

Usage Guidelines

Use this command to advertise Outbound Route Filtering capability to neighbor. This capability can be enabled in inbound and outbound direction separately

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# capability orf prefix-list send
```

capability orf prefix-list

Description

This command enables BGP to advertise Outbound Route Filtering (ORF) capability to neighbors in a peer-group

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
capability orf prefix-list { send | receive | both }
no capability orf prefix-list { [ send ] | [ receive ] | [ both ] } ]
```

Usage Guidelines

Use this command to advertise Outbound Route Filtering capability to neighbors in a peer-group. This capability can be enabled in inbound and outbound direction separately

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# address-family ipv4 unicast
sonic(config-router-bgp Pg-af)# capability orf prefix-list send
```

capability orf prefix-list

Description

Advertise prefixlist ORF capability to this neighbor

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
capability orf prefix-list { send | receive | both }
no capability orf prefix-list { [ send ] | [ receive ] | [ both ] } ]
```

capability orf prefix-list

Description

```
Advertise capability to this neighbor
```

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
capability orf prefix-list { send | receive | both }
no capability orf prefix-list { [ send ] | [ receive ] | [ both ] } ]
```

channel-group

Description

```
Configure PortChannel parameters
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
channel-group <lag-id>
no channel-group
```

Parameters

Name	Description	Type
lag-id		Integer

channel-group

Description

Configure PortChannel parameters

Parent Commands (Modes)

interface range iface_range_num

Syntax

```
channel-group <lag-id>
no channel-group
```

Parameters

Name	Description	Type
lag-id		Integer

class

Description

Configures flow match criteria and its actions

Parent Commands (Modes)

policy-map <fbs-policy-name> type { qos | monitoring | forwarding | copp | acl-copp }

Syntax

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
no class <fbs-class-name>
```

Parameters

Name	Description	Type
fbs-class-name	WORD	String
fbs-flow-priority		Integer

class-map

Description

Configures Class-map

Parent Commands (Modes)

configure terminal

Syntax

```
class-map <fbs-class-name> match-type { acl | { fields match-all } | copp }
no class-map <fbs-class-name>
```

Parameters

Name	Description	Type
fbs-class-name	WORD	String

Usage Guidelines

Class-map name can be of maximum 63 characters. The name must begin with A-Z, a-z or 0-9.
Underscore and hypens can be used except as the first character.

Examples

```
sonic(config)# class-map class_permit_ip match-type fields
```

clear access-list counters

Description

Clear access-list counters

Syntax

```
clear access-list counters
```

clear audit-log

Description

```
Clear audit log
```

Syntax

```
clear audit-log
```

clear authentication history interface

Description

```
Clear authentication history of an interface or all interfaces.
```

Syntax

```
clear authentication history interface { all | <port> }
```

Parameters

Name	Description	Type
port	EthernetNUM	

clear authentication sessions interface

Description

```
Clear authentication sessions of an interface or all interfaces.
```

Syntax

```
clear authentication sessions interface { all | <port> }
```

Parameters

Name	Description	Type
port	EthernetNUM	

clear bfd peer

Description

Clears counters of the specific Bidirectional Forwarding detection(BFD) peer with the filters.
BFD packet counters will be reset to 0.

Syntax

```
clear bfd peer { <peer_ipv4> | <peer_ipv6> } [ vrf <vrfname> ] [ multihop ] [ local-address { <local_ipv4> | <local_ipv6> } ] [ interface <interfacename> ] counters
```

Parameters

Name	Description	Type
peer_ipv4	A.B.C.D	String
peer_ipv6	A::B	String
vriname	WORD	String
local_ipv4	A.B.C.D	String
local_ipv6	A::B	String
interfacename	Interface Type - Ranges	

Examples

```
device# clear bfd peer 192.168.2.1 interface Ethernet0 counters
```

clear bgp all

Description

This command clears/resets all BGP information including neighbors, peer-group etc.

Syntax

```
clear bgp all [ vrf <vrf-name> ] { { [ * { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } ] } | { [ <as-num-dot> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } ] } | { [ <ipv4> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } ] } | { [ <ipv6> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } ] } | { [ dampening { [ <ip-addr> ] | [ <ip-prefix> ] ] } ] } | { [ external { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } ] } | { [ interface { Ethernet | PortChannel | Vlan } ] } | { peer-group <peer-group-name> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } ] } }
```

Parameters

Name	Description	Type
vrf-name	WORD	String
as-num-dot	1-4294967295	Integer
ipv4	A.B.C.D	String
ipv6	A::B	String
ip-addr	A.B.C.D	String
ip-prefix	A.B.C.D/mask	String
peer-group-name	WORD	String

Usage Guidelines

Use this command to clear BGP information. Following is a partial list of information with command syntax that can be cleared.

- clear bgp all *

 - This command clears all BGP neighbors in the all address-families activated

- clear bgp all A.B.C.D/A::B
 - Clear peers with address of peer_ip and this address-family activated.
- clear bgp all A.B.C.D/A::B soft {in | out}
 - 'in' option will send route-refresh request unless using 'soft-reconfiguration inbound.'
 - 'out' option will resend all outbound updates

Examples

```
leaf4# clear bgp all 14.14.14.1
```

```
clear bgp ipv4
```

Description

This command clears/resets BGP information including neighbors, peer-group, dampening etc.

Syntax

```
clear bgp ipv4 unicast [ vrf <vrf-name> ] { { * { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } | { <as-num-dot> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } | { external { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } | { <ipv4-addr> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } | { <ip-prefix> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } | { interface { Ethernet | PortChannel | Vlan } { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } } | { peer-group [ <peer-group-name> ] { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } } | { dampening { [ <ip-addr> ] | [ <ip-prefix> ] } } } | in | out | { soft { [ in ] | [ out ] } }
```

Parameters

Name	Description	Type
vrf-name	WORD	String
as-num-dot	1-4294967295	Integer
ipv4-addr	A.B.C.D	String
ipv6-addr	A::B	String
ip-prefix	A.B.C.D/mask	String
peer-group-name	WORD	String
ip-addr	A.B.C.D	String

Usage Guidelines

Use this command to clear BGP information. Following is a partial list of information with command syntax that can be cleared.

- `clear bgp {ipv4 | ipv6} unicast *`
This command clears all BGP neighbors with this address-family and sub-address-family activated
- `clear bgp {ipv4 | ipv6} unicast peer_ip`
Clear peers with address of `peer_ip` and this address-family activated.
- `clear bgp ipv4 unicast dampening { address | prefix | in | out | soft }`
Clear dampened routes.
- `clear bgp {ipv6 | ipv4} unicast soft {in | out}`
`'in'` option will send route-refresh request unless using `'soft-reconfiguration inbound'`.
`'out'` option will resend all outbound updates

Examples

```
leaf4# clear bgp ipv4 unicast 14.14.14.1
```

```
clear bgp ipv6
```

Description

Clear BGP IPv6

Syntax

```
clear bgp ipv6 unicast [ vrf <vrf-name> ] { { * { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } | { <as-num-dot> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } } | { external { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } } | { <ipv4-addr> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } } | { <ip-prefix> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } } | { interface { Ethernet | PortChannel | Vlan } { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } } | { peer-group [ <peer-group-name> ] { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } } | { dampening { [ <ip-addr> ] | [ <ip-prefix> ] } } } | in | out | { soft { [ in ] | [ out ] } }
```

Parameters

Name	Description	Type
vrf-name	WORD	String
as-num-dot	1-4294967295	Integer
ipv4-addr	A.B.C.D	String
ipv6-addr	A::B	String
ip-prefix	A.B.C.D/mask	String
peer-group-name	WORD	String
ip-addr	A.B.C.D	String

clear bgp l2vpn evpn

Description

This command clears/resets BGP information for EVPN address-family on neighbors

Syntax

```
clear bgp l2vpn evpn { { [ <as-num-dot> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } ] } | { [ * { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } ] } } | { [ external { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } ] } } | { [ interface { <ifname> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } ] } } | { [ peer-group { <peer-group name> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } } ] } } | { [ <neighbor-ipv6> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } ] } } | { [ <neighbor-ipv4> { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } ] } } | { [ in ] | [ out ] | { [ soft { [ in ] | [ out ] ] } ] } }
```

Parameters

Name	Description	Type
as-num-dot	1-4294967295	Integer
ifname	String	String
peer-group name	String	String
neighbor-ipv6	A::B	String
neighbor-ipv4	A.B.C.D	String

Usage Guidelines

Use this command to clear BGP information. Following is a partial list of information with command syntax that can be cleared.

- `clear bgp l2vpn evpn *`
This command clears all BGP neighbors with address-family l2vpn evpn activated
- `clear bgp l2vpn evpn {peer_ip} *`
Clear peers with address of peer_ip and address-family l2vpn evpn activated.
- `clear bgp l2vpn evpn soft {in | out}`
'in' option will send route-refresh request unless using 'soft-reconfiguration inbound'.
'out' option will resend all outbound updates

Examples

```
sonic# clear bgp l2vpn evpn *
```

clear buffer-pool

Description

This command is used to clear user/persistent watermark counters recorded by the system.

Syntax

```
clear buffer-pool { watermark | persistent-watermark }
```

Usage Guidelines

Use this command to clear user/persistent watermark counters recorded by the system.

Examples

clear core-files

Description

```
This command removes all core files and also removes core description from systemd-journal. It is to be noted that this operation also removes all the information from the systemd-journal which is not be related to core files.
```

Syntax

```
clear core-files
```

Usage Guidelines

```
sonic# clear core-files
```

Examples

```
sonic# clear core-files
```

Features this CLI belongs to

- COREDUMP

clear counters

Description

```
Clear counters
```

Syntax

```
clear counters interface { rif | all | <port> | { PortChannel <ifId> } }
```

Parameters

Name	Description	Type
port	EthernetNUM	
ifId	Integer	

clear counters service-policy

Description

Clears flow based services applied policy-map applied at Switch/Global level

Syntax

```
clear counters service-policy { Switch | CtrlPlane } [ type { qos | monitoring | forwarding | copp | acl-copp } ]
```

Usage Guidelines

Policy-map type argument is optional. If policy-map type not specified it clear fbs policies counters for given interfaces for all policies matching that interface.

Examples

```
sonic# clear counters service-policy Switch type qos
```

clear counters service-policy interface

Description

Clears flow based services applied policies counters by interface

Syntax

```
clear counters service-policy interface { <eth-if-id> | <po-if-id> | <vlan-if-id> | <eth-sub-if-id> | <po-sub-if-id> | CPU } [ type { qos | monitoring | forwarding | copp | acl-copp } ]
```

Parameters

Name	Description	Type
eth-if-id	EthernetNUM	
po-if-id	PortChannelNUM	
vlan-if-id	VlanNUM	
eth-sub-if-id	EthernetX.Y	
po-sub-if-id	PortChannelNUM	

Usage Guidelines

Policy-map type argument is optional. If policy-map type not specified it clear fbs policies counters for given interfaces for all policies matching that interface.

Examples

```
clear counters service-policy interface Vlan 100 type qos
```

```
clear counters service-policy policy-map
```

Description

Clears flow based services applied policies counters by policy name

Syntax

```
clear counters service-policy policy-map <fbs-policy-name> { { [ interface { <eth-if-id> | <po-if-id> | <vlan-if-id> | <eth-sub-if-id> | <po-sub-if-id> | CPU } ] } | [ Switch ] | [ CtrlPlane ] } }
```

Parameters

Name	Description	Type
fbs-policy-name	WORD	String
eth-if-id	EthernetNUM	
po-if-id	PortChannelNUM	
vlan-if-id	VlanNUM	
eth-sub-if-id	EthernetX.Y	
po-sub-if-id	PortChannelNUM	

Usage Guidelines

Interface argument is optional. If not specified it clear fbs policies counters for given policy-map for all interfaces

Examples

```
clear counters service-policy policy-map policy_vrf interface Vlan 100
```

```
clear counters tam
```

Description

This command is used to clear flowgroup counters.

Syntax

```
clear counters tam { flow-groups { all | <name> } }
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

Use this command to clear flowgroup counters.

Examples

```
sonic-cl# clear counters tam flow-groups all  
sonic-cl# clear counters tam flow-groups flowgroup_name
```

clear counters vxlan

Description

clear command to clear VXLAN tunnel counters per remote VTEP or all remotes.

Syntax

```
clear counters vxlan [ <remote_ip_addr> ]
```

Parameters

Name	Description	Type
remote_ip_addr	A.B.C.D	String

Usage Guidelines

```
sonic# clear counters vxlan
```

Examples

```
sonic# clear counters vxlan
sonic# clear counters vxlan 1.2.34.2
```

clear evpn dup-addr

Description

Duplicate address

Syntax

```
clear evpn dup-addr { vni { all | { <vni-number> { { [ mac { <mac-addr> } ] } | { [ ip { <ip-addr> } ] } } } }
```

Parameters

Name	Description	Type
vni-number	VNI	Integer
mac-addr	nn:nn:nn:nn:nn:nn	String
ip-addr	A.B.C.D/A::B	String

clear ip arp

Description

To delete dynamically learned IPv4 entries from the ARP table, use the `clear ip arp` command. To specify the entries to be deleted, enter an interface, port channel, or VLAN, an IPv4 address, or a combination to match.

Use the `show ip arp` command to verify that the ARP entries have been deleted.

Syntax

```
clear ip arp [ <ip-addr> ] [ vrf { <vrfname> | mgmt | all } ]
```

Parameters

Name	Description	Type
ip-addr	A.B.C.D	String
vrfname	VRF name (prefixed by Vrf, Max: 15 characters)	String

Usage Guidelines

```
sonic# clear ip arp [interface {Ethernet < port-number >| PortChannel < number >| Vlan < vlan-id >| Management < port-number >}] [< ipv4-address >]
```

Examples

```
sonic# show ip arp
-----
Address      Hardware address   Interface     Egress Interface
-----
192.168.1.4  00:01:02:03:44:55  Ethernet8      -
192.168.2.4  00:01:02:03:ab:cd  PortChannel1200 -
192.168.3.6  00:01:02:03:04:05  Vlan100       Ethernet4
10.11.48.254 00:01:e8:8b:44:71  eth0          -
10.14.8.102  00:01:e8:8b:44:71  eth0          -
sonic# clear ip arp
sonic# show ip arp
sonic#
```

```
sonic# show ip arp
-----
Address      Hardware address   Interface     Egress Interface
-----
192.168.1.4  00:01:02:03:44:55  Ethernet8      -
192.168.2.4  00:01:02:03:ab:cd  PortChannel1200 -
192.168.3.6  00:01:02:03:04:05  Vlan100       Ethernet4
10.11.48.254 00:01:e8:8b:44:71  eth0          -
10.14.8.102  00:01:e8:8b:44:71  eth0          -
sonic# clear ip arp interface Vlan 100
sonic# show ip arp
-----
Address      Hardware address   Interface     Egress Interface
-----
192.168.1.4  00:01:02:03:44:55  Ethernet8      -
192.168.2.4  00:01:02:03:ab:cd  PortChannel1200 -
10.11.48.254 00:01:e8:8b:44:71  eth0          -
10.14.8.102  00:01:e8:8b:44:71  eth0          -
```

```

sonic# show ip arp
-----
Address      Hardware address   Interface      Egress Interface
-----
192.168.1.4  00:01:02:03:44:55  Ethernet8       -
192.168.2.4  00:01:02:03:ab:cd  PortChannel200 -
192.168.3.6  00:01:02:03:04:05  Vlan100        Ethernet4
10.11.48.254 00:01:e8:8b:44:71  eth0          -
10.14.8.102  00:01:e8:8b:44:71  eth0          -
sonic# clear ip arp 192.168.1.4
sonic# show ip arp
-----
Address      Hardware address   Interface      Egress Interface
-----
192.168.2.4  00:01:02:03:ab:cd  PortChannel200 -
192.168.3.6  00:01:02:03:04:05  Vlan100        Ethernet4
10.11.48.254 00:01:e8:8b:44:71  eth0          -
10.14.8.102  00:01:e8:8b:44:71  eth0          -

```

clear ip arp interface

Description

Clear ARP entries for this interface

Syntax

```
clear ip arp interface { <phy-if-name> | <subif-name> | <mgmt-if-name> | <po-if-name> |
<vlan-if-name> }
```

Parameters

Name	Description	Type
phy-if-name	EthernetNUM	
subif-name	Interface Type	
mgmt-if-name		
po-if-name	PortChannelNUM	
vlan-if-name	VlanNUM	

clear ip dhcp snooping binding

Description

Clear all or a specific dynamic DHCP Snooping binding entries

Syntax

```
clear ip dhcp snooping binding [ <ip-address> { <mac-address> { vlan { <phy-if-name> |  
PortChannel } } } ]
```

Parameters

Name	Description	Type
ip-address	A.B.C.D	String
mac-address	nn:nn:nn:nn:nn:nn	String
phy-if-name	EthernetNUM	

clear ip dhcp snooping statistics

Description

```
Clear Ip DHCP Snooping statistics
```

Syntax

```
clear ip dhcp snooping statistics <iface>
```

Parameters

Name	Description	Type
iface	Interface Type - Ranges	

clear ip dhcp snooping statistics detail

Description

```
Clear Ip DHCP Snooping statistics details
```

Syntax

```
clear ip dhcp snooping statistics detail
```

clear ip dhcp-relay

Description

```
Clear IPv4 dhcp-relay statistics
```

Syntax

```
clear ip dhcp-relay { statistics <ifName> }
```

Parameters

Name	Description	Type
ifName	String	String

clear ip helper-address statistics

Description

```
Clears IP helper statistics on interface.
```

Syntax

```
clear ip helper-address statistics [ <iface> ]
```

Parameters

Name	Description	Type
iface	Interface Type - Ranges	

Usage Guidelines

```
clear ip helper-address statistics [ <interface-name> ]
```

Examples

```
clear ip helper-address statistics
```

```
clear ip helper-address statistics Ethernet0
```

Features this CLI belongs to

- IP Helper

Alternate command

```
clear ip helper_address statistics
```

clear ip ospf

Description

This command clears/resets OSPFv2 sessions on an OSPFv2 interface.

Syntax

```
clear ip ospf { { interface [ <interface-name> ] } | { vrf { <vrf-name> { interface [ <interface-name> ] } } }
```

Parameters

Name	Description	Type
interface-name	Interface Type - Ranges	
vrf-name	String	String

Usage Guidelines

Use this command to clear OSPFv2 sessions. Following is a partial list of information with command syntax that can be cleared.

- clear ip ospf interface IF-NAME
This command clears all OSPFv2 sessions on an OSPFv2 interface
- clear ip ospf vrf VRF-NAME interface IF-NAME
This command clears all OSPFv2 sessions on interface in VRF VRF-NAME

Examples

```
sonic# clear ip ospf interface Ethernet64
sonic# clear ip ospf vrf default interface Ethernet64
```

clear ip pim

Description

These commands are useful for debugging. One resets the PIM interfaces. The other rescans the Outgoing Interface Lists (OILs).

Syntax

```
clear ip pim [ vrf <vrf-name> ] { [ interfaces ] | [ oil ] }
```

Parameters

Name	Description	Type
vrf-name	WORD	String

Usage Guidelines

- * `clear ip pim [vrf <vrf-name>] interfaces` ==> To reset all PIM interfaces of a particular VRF
- * `clear ip pim [vrf <vrf-name>] oil` ==> To rescan PIM OIL (Outgoing Interfaces List) of all multicast entries of a particular VRF

Examples

```
sonic# clear ip pim interfaces
sonic#
-----
sonic# clear ip pim vrf Vrf2 interfaces
sonic#
```

```
sonic# clear ip pim oil
sonic#
-----
sonic# clear ip pim vrf Vrf2 oil
sonic#
```

clear ip sla

Description

```
Clears counters for an IP SLA instances
```

Syntax

```
clear ip sla <id>
```

Parameters

Name	Description	Type
id		Integer

Examples

```
sonic(config)# no clear ip sla 10
```

```
clear ip sla all
```

Description

```
Clears counters for all IP SLA instances
```

Syntax

```
clear ip sla all
```

Examples

```
sonic(config)# no clear ip sla all
```

```
clear ipv6 dhcp snooping binding
```

Description

```
Clear all or a specific IPv6 DHCP Snooping binding entry(s)
```

Syntax

```
clear ipv6 dhcp snooping binding [ <ip-address> { <mac-address> { vlan { <phy-if-name> |  
PortChannel } } } ]
```

Parameters

Name	Description	Type
ip-address	A::B	String
mac-address	nn:nn:nn:nn:nn:nn	String
phy-if-name	EthernetNUM	

clear ipv6 dhcp snooping statistics

Description

```
Clear IPV6 DHCP Snooping statistics
```

Syntax

```
clear ipv6 dhcp snooping statistics <iface>
```

Parameters

Name	Description	Type
iface	Interface Type - Ranges	

clear ipv6 dhcp snooping statistics detail

Description

```
Clear Ipv6 DHCP Snooping statistics details
```

Syntax

```
clear ipv6 dhcp snooping statistics detail
```

clear ipv6 dhcp-relay

Description

Clear IPv6 DHCP relay statistics

Syntax

```
clear ipv6 dhcp-relay { statistics <ifName> }
```

Parameters

Name	Description	Type
ifName	String	String

clear ipv6 neighbors

Description

To delete dynamically learned IPv6 entries from the NDP table, use the 'clear ipv6 neighbor' command.

To specify the entries to be deleted, enter an interface, port channel, or VLAN, an IPv6 address, or a combination to match.

Use the 'show ipv6 neighbors' to verify that the IPv6 entries have been deleted.

Syntax

```
clear ipv6 neighbors [ <ip-addr> ] [ vrf { <vrfname> | mgmt | all } ]
```

Parameters

Name	Description	Type
ip-addr	A::B	String
vrfname	VRF name (prefixed by Vrf, Max: 15 characters)	String

Usage Guidelines

```
sonic# clear ipv6 neighbors [interface {Ethernet < port-number >| PortChannel < number >| Vlan < vlan-id >| Management < port-number >}] [< ipv6-address >]
```

Examples

```
sonic# show ipv6 neighbors
```

Address	Hardware address	Interface	Egress Interface
20::1	00:01:02:03:44:55	Ethernet8	-
20::2	00:01:02:03:ab:cd	PortChannel200	-
20::3	00:01:02:03:04:05	Vlan100	Ethernet4
fe80::e6f0:4ff:fe79:34c7	00:01:e8:8b:44:71	eth0	-

```
sonic# clear ipv6 neighbors
```

```
sonic# show ipv6 neighbors
```

```
sonic# show ipv6 neighbors
```

Address	Hardware address	Interface	Egress Interface
20::1	00:01:02:03:44:55	Ethernet8	-
20::2	00:01:02:03:ab:cd	PortChannel200	-
20::3	00:01:02:03:04:05	Vlan100	Ethernet4
fe80::e6f0:4ff:fe79:34c7	00:01:e8:8b:44:71	eth0	-

Address	Hardware address	Interface	Egress Interface
20::1	00:01:02:03:44:55	Ethernet8	-
20::2	00:01:02:03:ab:cd	PortChannel200	-
fe80::e6f0:4ff:fe79:34c7	00:01:e8:8b:44:71	eth0	-

```
sonic# show ipv6 neighbors
```

Address	Hardware address	Interface	Egress Interface
20::1	00:01:02:03:44:55	Ethernet8	-
20::2	00:01:02:03:ab:cd	PortChannel200	-
20::3	00:01:02:03:04:05	Vlan100	Ethernet4
fe80::e6f0:4ff:fe79:34c7	00:01:e8:8b:44:71	eth0	-

Address	Hardware address	Interface	Egress Interface
20::1	00:01:02:03:44:55	Ethernet8	-
20::2	00:01:02:03:ab:cd	PortChannel200	-
20::3	00:01:02:03:04:05	Vlan100	Ethernet4

clear ipv6 neighbors interface

Description

```
Clear NDP entries for this interface
```

Syntax

```
clear ipv6 neighbors interface { <phy-if-name> | <subif-name> | <mgmt-if-name> | <po-if-name> |  
    <vlan-if-name> }
```

Parameters

Name	Description	Type
phy-if-name	EthernetNUM	
subif-name	Interface Type	
mgmt-if-name		
po-if-name	PortChannelNUM	
vlan-if-name	VlanNUM	

clear logging

Description

```
Clear logging
```

Syntax

```
clear logging
```

clear mac address-table dynamic

Description

```
Clear MAC address-table for dynamic commands
```

Syntax

```
clear mac address-table dynamic { all | { Vlan <vlan-id> } | { interface { Ethernet |  
    PortChannel } } }
```

Parameters

Name	Description	Type
vlan-id		Integer

clear mac dampening-disabled-ports

Description

```
Clear MAC dampening disabled ports
```

Syntax

```
clear mac dampening-disabled-ports { all | PORT_ID | PO_ID }
```

clear nat

Description

```
Clear NAT
```

Syntax

```
clear nat { translations | statistics }
```

clear priority-group

Description

```
This command clears priority-group watermarks, persistent-watermarks and breaches etc.
```

Syntax

```
clear priority-group { { watermark { { headroom { [ interface { <phy-intf-name> } ] } } } | { shared { [ interface { <phy-intf-name> } ] } } } | { persistent-watermark { { headroom { [ interface { <phy-intf-name> } ] } } } | { shared { [ interface { <phy-intf-name> } ] } } }
```

Parameters

Name	Description	Type
phy-intf-name	EthernetNUM	

Usage Guidelines

Use this command to clear priority-group watermarks, persistent-watermarks and breaches etc. There are various CLI options available to clear information for unicast/multicast on priority-group, interface.

- clear priority-group (watermark|persistent-watermark) (headroom|shared) (interface Ethernet [ifname])
This command will display all queue watermarks

Examples

```
sonic# clear priority-group watermark shared interface Ethernet 0
sonic# clear priority-group persistent-watermark headroom
```

clear queue

Description

This command clears queue counters, watermarks, persistent-watermarks and breaches etc.

Syntax

```
clear queue { { counters { [ interface { { <phy-intf-name> { [ queue <queue-id> ] } } ] } | { CPU { [ queue <queue-id> ] } } ] } | { watermark { { unicast { [ interface { <phy-intf-name> } ] } } | { multicast { [ interface { <phy-intf-name> } ] } } } | { persistent-watermark { { unicast { [ interface { <phy-intf-name> } ] } } } | { multicast { [ interface { <phy-intf-name> } ] } } } | { wred-ecn { counters { [ interface { <phy-intf-name> } ] } } }
```

Parameters

Name	Description	Type
phy-intf-name	EthernetNUM	
queue-id	Integer	

Usage Guidelines

Use this command to clear queue counters, WRED ECN counters, watermarks, persistent-watermarks and breaches etc. There are various CLI options available to clear counters for queue, interface and all interfaces.

- clear queue counters (interface Ethernet|CPU [ifname] (queue [id]))
This command will clear all queue counters
- clear queue (watermark|persistent-watermark) (unicast|multicast) (interface Ethernet|CPU [ifname])
This command will clear all queue watermarks or persistent watermark
- clear queue wred-ecn counters (interface Ethernet)
This command will clear all WRED ECN queue counters

Examples

```
sonic# clear queue counters interface Ethernet 0 queue 0
sonic# clear queue watermark unicast
sonic# clear queue persistent-watermark multicast interface Ethernet 0
sonic# clear queue wred-ecn counters interface Ethernet 0
```

clear radius-server dynamic-author statistics

Description

This command clears radius dynamic authorization global counters and per DAS client counters.

Syntax

```
clear radius-server dynamic-author statistics [ client { all | <ipaddr_hostname> } ]
```

Parameters

Name	Description	Type
ipaddr_hostname	WORD	String

Usage Guidelines

This command clears radius dynamic authorization global counters and per DAS client counters.

Examples

```
sonic-cl# clear radius-server dynamic-author statistics [ client { all | ipv4 | ipv6 |
hostname }]
```

clear radius-server statistics

Description

Clear RADIUS statistics

Syntax

```
clear radius-server statistics
```

client

Description

This command configures the IP address or IPV6 address or hostname of the AAA server client. It uses the optional **server-key** keyword at the client level.

Parent Commands (Modes)

```
aaa server radius dynamic-author
```

Syntax

```
client <hostname_or_ipaddr> { [ server-key { <key> [ encrypted ] } ] }  
no client <hostname_or_ipaddr>
```

Parameters

Name	Description	Type
hostname_or_ipaddr	WORD	String
key	WORD	String

Usage Guidelines

This command configures the IP address or IPV6 address or hostname of the AAA server client. It uses the optional **server-key** keyword at the client level.

Examples

```
sonic-cli(config)# aaa server radius dynamic-author  
sonic-cli(config-radius-da)# client 1.1.1.1 server-key  
    U2FsdGVkX18LcQREyGJP/aDuWTi34jJLzdqNy9W5WxI= encrypted  
sonic-cli(config-radius-da)#[/pre>
```

client-to-client reflection

Description

This command configures client to client route reflection

Parent Commands (Modes)

router bgp <as-num-dot> { [vrf <vrf-name>] }

Syntax

```
client-to-client reflection  
no client-to-client reflection
```

Usage Guidelines

Use this command to configure client to client route reflection

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# client-to-client reflection
```

clock timezone

Description

Configure clock timezone

Parent Commands (Modes)

configure terminal

Syntax

```
clock timezone <timezone>  
no clock timezone
```

Parameters

Name	Description	Type
timezone	Standard timezones	

cluster-id

Description

This command configures cluster ID for BGP router

Parent Commands (Modes)

router bgp <as-num-dot> { [vrf <vrf-name>] }

Syntax

```
cluster-id <intval-ip>
no cluster-id
```

Parameters

Name	Description	Type
intval-ip	A.B.C.D or [1..4294967295]	String

Usage Guidelines

A cluster is a collection of route reflectors and their clients, and is used by route reflectors to avoid looping. Use this command to configure cluster-ID (an IP address or a number) on a BGP router

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# cluster-id 23.79.154.17
```

coalesce-time

Description

This command configures subgroup coalesce timer interval in milliseconds

Parent Commands (Modes)

router bgp <as-num-dot> { [vrf <vrf-name>] }

Syntax

```
coalesce-time <coaltime>
no coalesce-time
```

Parameters

Name	Description	Type
coaltime	1-4294967295	Integer

Usage Guidelines

Use this command to configure subgroup coalesce timer interval

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# coalesce-time 2000
```

collector

Description

This command configures collector information.

Parent Commands (Modes)

tam

Syntax

```
collector <name> ip <ip_address> port { <port_number> { [ protocol <protocol_type> ] } } [ vrf <vrf_name> ]
no collector <name>
```

Parameters

Name	Description	Type
name	WORD	String
ip_address	A.B.C.D/A::B	String
port_number		Integer
protocol_type	Transport Protocol Method	Select [UDP(UDP) TCP(TCP)]
vrf_name	WORD	String

Usage Guidelines

This command configures collector information. A collector is typically a machine reachable from the switch, where the telemetry reports are sent.

Examples

```
sonic(config-tam)# collector c2 ip 2.2.2.2 port 7676 protocol UDP vrf Vrf1
sonic(config-tam)#
sonic(config-tam)# show tam collectors
Name          IP Address      Port     Protocol   Vrf
-----        -----          -----    -----      -----
c2            2.2.2.2        7676     UDP       Vrf1
```

compatible

Description

Configures OSPFv2 RFC1583 compatibility

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
compatible rfc1583
no compatible rfc1583
```

Usage Guidelines

Use this command to configure RFC1583 compatibility.

Examples

```
sonic-cli(config-router-ospf)# compatible rfc1583
```

Features this CLI belongs to

- OSPFv2 ## confederation ##### Description

This command configures the list of AS numbers that are part of the confederation. This command also allows user to configure the router's confederation ID

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
confederation { { identifier <id-as> } | { peers <peer-as> } }
no confederation { { identifier <id-as> } | peers }
```

Parameters

Name	Description	Type
id-as	1-4294967295	Integer
peer-as	1-4294967295	Integer

Usage Guidelines

Use this command to create confederation peers and local confederation Id

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# confederation identifier 65000
sonic(config-router-bgp)# confederation peers 65100
sonic(config-router-bgp)# confederation peers 65200
sonic(config-router-bgp)# confederation peers 65300
```

configure terminal

Description

Configure from the terminal

Syntax

```
configure terminal
```

copp-action

Description

Configure CoPP action group

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
copp-action <copp-action-name>
no copp-action <copp-action-name>
```

Parameters

Name	Description	Type
copp-action-name	WORD	String

Examples

```
sonic(config)# classifier copp-system-arp match-type copp
sonic(config-classifier)#[
```

copy

Description

Perform file operations

Syntax

```
copy { { running-configuration { <run_to_home_url> } } | { startup-configuration {  
    <start_to_home_url> } } }
```

Parameters

Name	Description	Type
run_to_home_url	home:	String
start_to_home_url	home:	String

core enable

Description

Enable the capability to generate a core file when an application crash is detected by the kernel.

Parent Commands (Modes)

configure terminal

Syntax

```
core enable
```

Usage Guidelines

```
sonic(config)# core enable
```

Examples

```
sonic# configure terminal  
sonic(config)# core enable
```

Features this CLI belongs to

- COREDUMP

Alternate command

```
config core enable
```

counter

Description

```
Command to configure VXLAN counter polling interval.
```

Parent Commands (Modes)

```
interface vxlan <vxlan-if-name>
```

Syntax

```
counter { polling-interval <poll-timer-value> }
```

Parameters

Name	Description	Type
poll-timer-value		Integer

Usage Guidelines

```
(conf-if-vxlan-vtep)#counter polling-interval POLL_INTERVAL
POLL_INTERVAL - A value between 3 and 30 seconds
```

Examples

```
sonic(config)# interface vxlan vtep1
sonic(conf-if-vxlan-vtep1)# counter polling-interval 3
```

counters access-list enable

Description

```
Enable access-list counter polling
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
counters access-list enable  
no counters access-list enable
```

counters access-list interval

Description

```
Set access-list counter polling interval
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
counters access-list interval <interval>
```

Parameters

Name	Description	Type
interval	1-3600000	Integer

counters buffer-drop

Description

```
counters buffer drop config
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
counters buffer-drop { enable | { interval <interval-value> } }
```

Parameters

Name	Description	Type
interval-value		Integer

counters pg-drop

Description

```
counters priority group drop config
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
counters pg-drop { enable | { interval <interval-value> } }
```

Parameters

Name	Description	Type
interval-value		Integer

counters policer

Description

```
counters policer config
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
counters policer { enable | { interval <interval-value> } }
```

Parameters

Name	Description	Type
interval-value		Integer

counters port

Description

```
counters port config
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
counters port { enable | { interval <interval-value> } }
```

Parameters

Name	Description	Type
interval-value		Integer

counters queue

Description

```
counters queue config
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
counters queue { enable | { interval <interval-value> } }
```

Parameters

Name	Description	Type
interval-value		Integer

counters rif

Description

```
counters rif config
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
counters rif { enable | { interval <interval-value> } }
```

Parameters

Name	Description	Type
interval-value		Integer

counters watermark

Description

```
counters watermark config
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
counters watermark { enable | { interval <interval-value> } }
```

Parameters

Name	Description	Type
interval-value		Integer

crm

Description

Configure critical resource monitoring

Parent Commands (Modes)

configure terminal

Syntax

```
crm { polling { interval <crm-subcmd-data> } } | { thresholds { { all { { type
<crm-subcmd-data> } | { high <crm-subcmd-data> } | { low <crm-subcmd-data> } } } } | { acl {
{ group { { type <crm-subcmd-data> } | { high <crm-subcmd-data> } | { low <crm-subcmd-data>
} } | { entry { { type <crm-subcmd-data> } | { high <crm-subcmd-data> } | { low
<crm-subcmd-data> } } } | { counter { { type <crm-subcmd-data> } | { high <crm-subcmd-data>
} | { low <crm-subcmd-data> } } } } | { table { { type <crm-subcmd-data> } | { high
<crm-subcmd-data> } | { low <crm-subcmd-data> } } } } | { dnat { { type <crm-subcmd-data>
} | { high <crm-subcmd-data> } | { low <crm-subcmd-data> } } } | { snat { { type
<crm-subcmd-data> } | { high <crm-subcmd-data> } | { low <crm-subcmd-data> } } } | { fdb {
{ type <crm-subcmd-data> } | { high <crm-subcmd-data> } | { low <crm-subcmd-data> } } } | {
ipmc { { type <crm-subcmd-data> } | { high <crm-subcmd-data> } | { low <crm-subcmd-data>
} } | { ipv4 { { neighbor { { type <crm-subcmd-data> } | { high <crm-subcmd-data> } | { low
<crm-subcmd-data> } } } | { nexthop { { type <crm-subcmd-data> } | { high <crm-subcmd-data>
} | { low <crm-subcmd-data> } } } | { route { { type <crm-subcmd-data> } | { high
<crm-subcmd-data> } | { low <crm-subcmd-data> } } } } | { ipv6 { { neighbor { { type
<crm-subcmd-data> } | { high <crm-subcmd-data> } | { low <crm-subcmd-data> } } } | {
nexthop { { type <crm-subcmd-data> } | { high <crm-subcmd-data> } | { low <crm-subcmd-data>
} } | { route { { type <crm-subcmd-data> } | { high <crm-subcmd-data> } | { low
<crm-subcmd-data> } } } } | { nexthop { { group { { member { { type <crm-subcmd-data>
} | { high <crm-subcmd-data> } | { low <crm-subcmd-data> } } } } | { object { { type
<crm-subcmd-data> } | { high <crm-subcmd-data> } | { low <crm-subcmd-data> } } } } } } |
{ ipv6 { { neighbor { { type <crm-subcmd-data> } | { high <crm-subcmd-data> } } } | {
nexthop { { type <crm-subcmd-data> } | { high <crm-subcmd-data> } } | { route { { type <crm-subcmd-data>
} | { high <crm-subcmd-data> } } } } | { nexthop { { group { { member { { type <crm-subcmd-data>
} | { high <crm-subcmd-data> } } } } | { object { { type <crm-subcmd-data> } | { high <crm-subcmd-data>
} } } } } } } | { no crm { all | { polling interval } | { thresholds { all | { acl { { group { type | high | low
| { entry { type | high | low } } | { counter { type | high | low } } } } | { table { type
| high | low } } } | { dnat { type | high | low } } | { snat { type | high | low } } } | {
fdb { type | high | low } } | { ipmc { type | high | low } } | { ipv4 { { neighbor { type |
high | low } } | { nexthop { type | high | low } } | { route { type | high | low } } } } } | {
ipv6 { { neighbor { type | high | low } } | { nexthop { type | high | low } } | { route { type |
high | low } } } } | { nexthop { { group { { member { { type | high | low } } } } | { object { { type <crm-subcmd-data> } | { high <crm-subcmd-data> } } } } } } } }
```

Parameters

Name	Description	Type
crm-subcmd-data		Integer

dampening

Description

This command enables BGP route-flap dampening and specifies dampening parameters

Parent Commands (Modes)

address-family ipv4 unicast

Syntax

```
dampening [ <halflife> { [ <reusethr> { <suppresssth> <maxsuppress> } ] } ]  
no dampening
```

Parameters

Name	Description	Type
halflife		Integer
reusethr		Integer
suppresssth		Integer
maxsuppress		Integer

Usage Guidelines

Use this command to configure route flap dampening feature for BGP routes. Route flap dampening algorithm is compatible with RFC 2439. The use of this command is not recommended nowadays. Currently implementation works only for IPv4 address family

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# dampening 10 1200 2000 40
```

default interface

Description

Use this command to restore a selected port to its factory default configuration.

Parent Commands (Modes)

configure terminal

Syntax

```
default interface <iface>
```

Parameters

Name	Description	Type
iface	EthernetNUM	

Usage Guidelines

```
default interface if-name
```

Examples

```
sonic# configure terminal
sonic(config)# default interface Ethernet 8
```

default interface range

Description

Apply default configuration on an interface range

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
default interface range iface_range_num
```

default ipv4-unicast

Description

This command activate IPv4 unicast address-family for a BGP peer by default

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
default ipv4-unicast  
no default ipv4-unicast
```

Usage Guidelines

Use this command to activate IPv4 unicast address family on BGP neighbors by default

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# default ipv4-unicast
```

default local-preference

Description

This command the default value for Local Preference parameter default

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
default local-preference <lprftime>
no default local-preference
```

Parameters

Name	Description	Type
lprftime	(0-4294967295)	Integer

Usage Guidelines

Use this command to set the deafult value of Local Preference parameter

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# default local-preference 200
```

default show-hostname

Description

This command instructs BGP to display hostname in certain display commands

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
default show-hostname
no default show-hostname
```

Usage Guidelines

Use this command to instruct BGP to display hostname in certain display command outputs.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# default show-hostname
```

default shutdown

Description

This command instructs BGP to make newly created neighbors in shutdown state

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
default shutdown
no default shutdown
```

Usage Guidelines

By default, newly created BGP neighbors are in admin enabled state. Use this command to keep newly created BGP neighbors in admin shutdown state.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# default shutdown
```

default subgroup-pkt-queue-max

Description

This command configures the maximum packet queue length for update groups

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
default subgroup-pkt-queue-max <maxval>
no default subgroup-pkt-queue-max
```

Parameters

Name	Description	Type
maxval		Integer

Usage Guidelines

Use this command to set a default maximum packet queue length for update groups

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# default subgroup-pkt-queue-max 50
```

default-information

Description

Configures default information origination policy rules.

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
default-information originate { [ always ] { [ metric <metricval> ] } { [ metric-type
<metrictype> ] } { [ route-map <routemapname> ] } }
no default-information originate { [ always ] [ metric ] [ metric-type ] [ route-map ] }
```

Parameters

Name	Description	Type
metricval		Integer
mertrictype		Integer
routemapname	WORD	String

Usage Guidelines

Use this command to configures default route information origination parameters. Metric value and metric type can be explicitly specified for default routes. Route map rules can also be applied on default routes.

Examples

```
sonic-cl(i(config-router-ospf)# default-information originate always
sonic-cl(i(config-router-ospf)# default-information originate metric 80
sonic-cl(i(config-router-ospf)# default-information originate metric-type 1
sonic-cl(i(config-router-ospf)# default-information originate route-map rmap_droute
```

Features this CLI belongs to

- OSPFv2 ## default-metric ### Description

Configures default metric for redistributed routes.

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
default-metric <defaultmetric>
no default-metric
```

Parameters

Name	Description	Type
defaultmetric		Integer

Usage Guidelines

Use this command to configures metric for redistributed routes.

Examples

```
sonic-cli(config-router-ospf)# default-metric 28
```

Features this CLI belongs to

- OSPFv2 ## default-originate ### Description

```
This command enables BGP to originate default route to this neighbor
```

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
default-originate [ route-map <rtemap> ]
no default-originate [ route-map <rtemap> ]
```

Parameters

Name	Description	Type
rtemap	WORD	String

Usage Guidelines

```
Use this command to originate a default route to this neighbor. User can
optionally use route-map along with this command to specify criteria
to originate default
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# default-originate
```

default-originate

Description

This command enables BGP to originate default route to neighbors in a peer-group

Parent Commands (Modes)

address-family ipv4 unicast

Syntax

```
default-originate [ route-map <rtemap> ]
no default-originate [ route-map <rtemap> ]
```

Parameters

Name	Description	Type
rtemap	WORD	String

Usage Guidelines

Use this command to originate default route to neighbors in a peer-group. User can optionally use route-map along with this command to specify criteria to originate default

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp-pg)# address-family ipv4 unicast
sonic(config-router-bgp-pg-af)# default-originate
```

default-originate

Description

Originate default route to this neighbor

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
default-originate [ route-map <rtemap> ]  
no default-originate [ route-map <rtemap> ]
```

Parameters

Name	Description	Type
rtemap	WORD	String

default-originate

Description

```
Originate default route to this neighbor
```

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
default-originate [ route-map <rtemap> ]  
no default-originate [ route-map <rtemap> ]
```

Parameters

Name	Description	Type
rtemap	WORD	String

default-originate ipv4

Description

```
This command enables border leaf to originate IPv4 default type-5 EVPN routes
```

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
default-originate ipv4  
no default-originate ipv4
```

Usage Guidelines

```
[no] default-originate ipv4
```

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100 vrf Vrf1  
sonic(config-router-bgp)# address-family l2vpn evpn  
sonic(config-router-bgp-af)# default-originate ipv4
```

default-originate ipv6

Description

```
This command enables border leaf to originate IPv6 default type-5 EVPN routes
```

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
default-originate ipv6  
no default-originate ipv6
```

Usage Guidelines

```
[no] default-originate ipv6
```

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100 vrf Vrf1  
sonic(config-router-bgp)# address-family l2vpn evpn  
sonic(config-router-bgp-af)# default-originate ipv6
```

delay-restore

Description

```
Configures MLAG delay restore time in seconds
```

Parent Commands (Modes)

```
mclag domain <mclag-domain-id>
```

Syntax

```
delay-restore <DL>
no delay-restore
```

Parameters

Name	Description	Type
DL	Delay restore time in seconds	Integer

Usage Guidelines

```
Use this command to change the default MLAG delay restore time
```

Examples

```
sonic-clia(config-mclag-domain-100)#delay-restore 180
```

delete-reason

Description

```
Remove dropcounter type
```

Parent Commands (Modes)

```
dropcounters <counter-name>
```

Syntax

```
delete-reason <reason>
```

Parameters

Name	Description	Type
reason	Drop reason list	String

description

Description

```
Add dropcounter description
```

Parent Commands (Modes)

```
dropcounters <counter-name>
```

Syntax

```
description <desc>
no description <desc>
```

Parameters

Name	Description	Type
desc	STRING	String

description

Description

```
Configures policy-map description
```

Parent Commands (Modes)

```
policy-map <fbs-policy-name> type { qos | monitoring | forwarding | copp | acl-copp }
```

Syntax

```
description <description-value>
no description
```

Parameters

Name	Description	Type
description-value	String (Max: 256 characters)	String

Usage Guidelines

A string describing the policy-map max 256 characters. Description should be in double quotes if it has spaces

Examples

```
sonic(config-policy-map)# description "Vrf policy information"
```

description

Description

Configures Class-map description

Parent Commands (Modes)

```
class-map <fb-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
description <description-value>
no description
```

Parameters

Name	Description	Type
description-value	String (Max: 256 characters)	String

Usage Guidelines

Examples

```
sonic(config-class-map)# description"ip match type class-map"
```

description

Description

```
Configures Class-map description
```

Parent Commands (Modes)

```
class-map <fbcs-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
description <description-value>
no description
```

Parameters

Name	Description	Type
description-value	String (Max: 256 characters)	String

Usage Guidelines

Examples

```
sonic(config-class-map)# description"ip match type class-map"
```

description

Description

Configures Class-map description

Parent Commands (Modes)

```
class-map <fbs-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
description <description-value>
no description
```

Parameters

Name	Description	Type
description-value	String (Max: 256 characters)	String

Usage Guidelines

Examples

```
sonic(config-class-map)# description"ip match type class-map"
```

description

Description

Updates description for the flows

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
description <description-value>
no description
```

Parameters

Name	Description	Type
description-value	String (Max: 256 characters)	String

Usage Guidelines

A string describing the flow with max 256 characters. Description should be in double quotes if it has spaces

Examples

```
sonic(config)# policy-map policy_qos type qos
sonic(config-policy-map)# class class_permit_ip priority 10
sonic(config-policy-map-flow)# description"flow to match ip traffic and dscp remark to 10"
```

description

Description

Updates description for the flows

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
description <description-value>
no description
```

Parameters

Name	Description	Type
description-value	String (Max: 256 characters)	String

Usage Guidelines

A string describing the flow with max 256 characters. Description should be in double quotes if it has spaces

Examples

```
sonic(config)# policy-map policy_qos type qos
sonic(config-policy-map)# class class_permit_ip priority 10
sonic(config-policy-map-flow)# description"flow to match ip traffic and dscp remark to 10"
```

description

Description

Updates description for the flows

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
description <description-value>
no description
```

Parameters

Name	Description	Type
description-value	String (Max: 256 characters)	String

Usage Guidelines

A string describing the flow with max 256 characters. Description should be in double quotes if it has spaces

Examples

```
sonic(config)# policy-map policy_qos type qos
sonic(config-policy-map)# class class_permit_ip priority 10
sonic(config-policy-map-flow)# description"flow to match ip traffic and dscp remark to 10"
```

description

Description

Updates description for the flows

Parent Commands (Modes)

class <fbs-class-name> [priority <fbs-flow-priority>]

Syntax

```
description <description-value>
no description
```

Parameters

Name	Description	Type
description-value	String (Max: 256 characters)	String

Usage Guidelines

A string describing the flow with max 256 characters. Description should be in double quotes if it has spaces

Examples

```
sonic(config)# policy-map policy_qos type qos
sonic(config-policy-map)# class class_permit_ip priority 10
sonic(config-policy-map-flow)# description"flow to match ip traffic and dscp remark to 10"
```

description

Description

Updates description for the flows

Parent Commands (Modes)

class <fbs-class-name> [priority <fbs-flow-priority>]

Syntax

```
description <description-value>
no description
```

Parameters

Name	Description	Type
description-value	String (Max: 256 characters)	String

Usage Guidelines

A string describing the flow with max 256 characters. Description should be in double quotes if it has spaces

Examples

```
sonic(config)# policy-map policy_qos type qos
sonic(config-policy-map)# class class_permit_ip priority 10
sonic(config-policy-map-flow)# description"flow to match ip traffic and dscp remark to 10"
```

description

Description

Configures PBF next-hop group description

Parent Commands (Modes)

```
pbf next-hop-group <fbn-nhgrp-name> [ type { ip | ipv6 } ]
```

Syntax

```
description <description-value>
no description
```

Parameters

Name	Description	Type
description-value	String (Max: 256 characters)	String

Usage Guidelines

A string describing the next hop group of max 256 characters. Description should be in double quotes if it has spaces

Examples

```
sonic(config-pbf-ipv6-nh-group)# description"example description"
```

description

Description

Configures PBF next-hop group description

Parent Commands (Modes)

```
pbf next-hop-group <fbn-nhgrp-name> [ type { ip | ipv6 } ]
```

Syntax

```
description <description-value>
no description
```

Parameters

Name	Description	Type
description-value	String (Max: 256 characters)	String

Usage Guidelines

A string describing the next hop group of max 256 characters. Description should be in double quotes if it has spaces

Examples

```
sonic(config-pbf-ipv6-nh-group)# description"example description"
```

description

Description

Configures PBF replication group description

Parent Commands (Modes)

```
pbf replication-group <fbn-replgrp-name> [ type { ip | ipv6 } ]
```

Syntax

```
description <description-value>
no description
```

Parameters

Name	Description	Type
description-value	String (Max: 256 characters)	String

Usage Guidelines

A string describing the replication group of max 256 characters. Description should be in double quotes if it has spaces

Examples

```
sonic(config-pbf-ipv6-nh-group)# description"example description"
```

description

Description

Configures PBF replication group description

Parent Commands (Modes)

```
pbf replication-group <fbn-replgrp-name> [ type { ip | ipv6 } ]
```

Syntax

```
description <description-value>
no description
```

Parameters

Name	Description	Type
description-value	String (Max: 256 characters)	String

Usage Guidelines

A string describing the replication group of max 256 characters. Description should be in double quotes if it has spaces

Examples

```
sonic(config-pbf-ipv6-nh-group)# description"example description"
```

description

Description

Set description to a link state tracking group.

Parent Commands (Modes)

```
link state track <grp-name>
```

Syntax

```
description <grp-descr>
no description
```

Parameters

Name	Description	Type
grp-descr	String	String

Examples

```
sonic(config-link-track)# description "Example description"
```

Alternate command

```
admin@sonic:~$ sudo config linktrack update <name> --description <description>
```

description

Description

```
Textual description
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
description <desc>
no description
```

Parameters

Name	Description	Type
desc	STRING	String

description

Description

```
Textual description
```

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
description <desc>
no description
```

Parameters

Name	Description	Type
desc	STRING	String

description

Description

```
Textual description
```

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
description <desc>
no description
```

Parameters

Name	Description	Type
desc	STRING	String

description

Description

```
Textual description
```

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
description <desc>
no description
```

Parameters

Name	Description	Type
desc	STRING	String

description

Description

```
Textual description
```

Parent Commands (Modes)

```
interface Management <mgmt-if-id>
```

Syntax

```
description <desc>
no description
```

Parameters

Name	Description	Type
desc	String	String

description

Description

Textual description

Parent Commands (Modes)

interface range iface_range_num

Syntax

```
description <desc>
no description
```

Parameters

Name	Description	Type
desc	STRING	String

description

Description

Textual description

Parent Commands (Modes)

interface range create vlan_range_num
interface range vlan_range_num

Syntax

```
description <desc>
no description
```

Parameters

Name	Description	Type
desc	STRING	String

description

Description

Textual description

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
interface range po_range_num
```

Syntax

```
description <desc>
no description
```

Parameters

Name	Description	Type
desc	STRING	String

description

Description

This command configures a display string for a BGP neighbor

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
description <String>
no description
```

Parameters

Name	Description	Type
String	STRING	String

Usage Guidelines

Use this command to configure a descriptive string for a BGP neighbor

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# remote-as 65100
sonic(config-router-bgp-neighbor)# description to_nyc_dc1
```

description

Description

This command configures a display string for a BGP peer-group

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
description <String>
no description
```

Parameters

Name	Description	Type
String	STRING	String

Usage Guidelines

Use this command to configure a descriptive string for a BGP peer-group

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# description My_PG_East_Cost_Nbrs
```

destination

Description

Configure SPAN mirror session. Supports mirroring to physical interface. Source interface can be either port or portchannel. Supports mirroring in rx/tx or both directions. Mirror session can be used in ACL configurations.

Parent Commands (Modes)

```
mirror-session <session-name>
```

Syntax

```
destination <phy-if-id> [ source { <phy-if-name> | <po-if-name> } ] [ direction <sess-direction> ]
```

Parameters

Name	Description	Type
phy-if-id	EthernetNUM	
phy-if-name	Interface Ethernet Range	
po-if-name	Interface PortChannel Range	
sess-direction	Port mirror session direction	Select [rx(rx) tx(tx) both(both)]

Examples

```
sonic(config)# mirror-session Mirror1
sonic(config-mirror-Mirror1)# destination Ethernet0 source Ethernet4 direction rx
Success
sonic(config-mirror-Mirror1)# exit
```

destination CPU

Description

Configure SPAN mirror session to CPU port.
Source interface can be either port or portchannel.
Supports mirroring in rx/tx or both directions.
Mirror session can be used in ACL configurations.

Parent Commands (Modes)

```
mirror-session <session-name>
```

Syntax

```
destination CPU [ source { <phy-if-name> | <po-if-name> } ] [ direction { { [ rx { { [ <filter-in-val> { [ tx <filter-out-val> ] } ] } | { [ tx <filter-out-val> ] } ] } } | { [ tx { [ <filter-out-val> ] | { [ rx <filter-in-val> ] } ] } { [ rx <filter-in-val> ] } ] }
```

Parameters

Name	Description	Type
phy-if-name	Interface Ethernet Range	
po-if-name	Interface PortChannel Range	
filter-in-val	String(Max: 72 characters)	String
filter-out-val	String(Max: 72 characters)	String

Examples

```
sonic(config)# mirror-session Mirror1
sonic(config-mirror-Mirror1)# destination Ethernet0 source Ethernet4 direction rx
Success
sonic(config-mirror-Mirror2)#

```

destination erspan

Description

Configure ERSPAN mirror session. Supports mirroring to any destination IP. Source interface can be either port or portchannel. Supports mirroring in rx/tx or both directions. Mirror session can be used in ACL configurations.

Parent Commands (Modes)

```
mirror-session <session-name>
```

Syntax

```
destination erspan [ dst-ip <dst_ip> ] [ src-ip <src_ip> ] [ dscp <ip_dscp> ] [ gre <ip_gre> ]  
[ ttl <ip_ttl> ] [ queue <queue_val> ] [ source { <phy-if-name> | <po-if-name> } ] [  
direction { { [ rx { { [ <filter-in-val> { [ tx <filter-out-val> ] } ] } | { [ tx  
<filter-out-val> ] } ] } | { [ tx { [ <filter-out-val> ] | { [ rx <filter-in-val> ] } ]  
} { [ rx <filter-in-val> ] } ] } ] ]
```

Parameters

Name	Description	Type
dst_ip	A.B.C.D/A::B	String
src_ip	A.B.C.D/A::B	String
ip_dscp		Integer
ip_gre	Hexadecimal Type	String
ip_ttl		Integer
queue_val		Integer
phy-if-name	Interface Ethernet Range	
po-if-name	Interface PortChannel Range	
filter-in-val	String(Max: 72 characters)	String
filter-out-val	String(Max: 72 characters)	String

Examples

```
sonic(config)# mirror-session Mirror2  
sonic(config-mirror-Mirror2)# destination erspan dst-ip 10.1.1.1 src-ip 11.1.1.1 dscp 10 ttl 10  
gre 0x88ee queue 10 source Ethernet4 direction rx filter MIRROR_ACL  
Success  
sonic(config-mirror-Mirror2)#[/pre>
```

destination-group

Description

This command is used to create a new telemetry destination group.

Parent Commands (Modes)

telemetry

Syntax

```
destination-group <dg-id>
no destination-group <name>
```

Parameters

Name	Description	Type
dg-id	WORD	String

Usage Guidelines

Use this command to create a new telemetry destination group.

Examples

```
sonic# configure terminal
sonic(config)# telemetry
sonic(config-telemetry)# destination-group dg1
sonic(conf-tm-dest-dg1)#

```

detect-multiplier

Description

Configure detection multiplier for Bidirectional Forwarding detection(BFD) peer for timeout.

Parent Commands (Modes)

```
peer <peer_ipv4>
peer <peer_ipv6>
peer [ interface ] <interfacename>
peer [ local-address ] <local_ipv4>
peer [ local-address ] <local_ipv6>
peer [ multihop ]
peer [ vrf ] <vrfname>
```

Syntax

```
detect-multiplier <multiplier>
```

Parameters

Name	Description	Type
multiplier		Integer

Usage Guidelines

Default value is 3.

Examples

```
device()#configure terminal
device(config)#bfd
device(conf-bfd)# peer 192.168.0.5 interface Ethernet0
device(conf-bfd-peer)# detect-multiplier 5
```

detect-multiplier

Description

Configure detection multiplier for Bidirectional Forwarding detection(BFD) peer for timeout.

Parent Commands (Modes)

```
profile <profilename>
```

Syntax

```
detect-multiplier <multiplier>
```

Parameters

Name	Description	Type
multiplier		Integer

Usage Guidelines

Default value is 3.

Examples

```
device()#configure terminal
device(config)#bfd
device(conf-bfd)# profile fast
device(conf-bfd-profile)# detect-multiplier 5
```

deterministic-med

Description

This command enables to carry out route-selection in a way that produces deterministic results locally, even in the face of MED and the lack of a well-defined order of preference it can induce on routes.

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

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

Usage Guidelines

Carry out route-selection in a way that produces deterministic answers locally, even in the face of MED and the lack of a well-defined order of preference it can induce on routes. Without this option the preferred route with MED may be determined largely by the order that routes were received in.

Setting this option will have a performance cost that may be noticeable when there are many routes for each destination. Currently in BGP it is implemented in a way that scales poorly as the number of routes per destination increases.

By default deterministic-med is disabled

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# deterministic-med
```

df-preference

Description

Configure DF selection preference

Parent Commands (Modes)

```
evpn ethernet-segment <es_id>
```

Syntax

```
df-preference <df_pref>
no df-preference
```

Parameters

Name	Description	Type
df_pref		Integer

diag-mode

Description

Configure port diagnostic mode

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
diag-mode <diag-mode>
no diag-mode
```

Parameters

Name	Description	Type
diag-mode	Set port diag mode	Select [on off]

diag-mode

Description

```
Configure port diagnostic mode
```

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
diag-mode <diag-mode>
no diag-mode
```

Parameters

Name	Description	Type
diag-mode	Set port diag mode	Select [on off]

disable-connected-check

Description

```
This command disables the restriction that eBGP peers must be directly
connected
```

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
disable-connected-check  
no disable-connected-check
```

Usage Guidelines

Use this command to allow peerings between directly connected eBGP peers using loopback addresses.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 30.30.30.3  
sonic(config-router-bgp-neighbor)# disable-connected-check
```

disable-connected-check

Description

This command disables the restriction that eBGP peers must be directly connected

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
disable-connected-check  
no disable-connected-check
```

Usage Guidelines

Use this command to allow peerings between directly connected eBGP peers using loopback addresses.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Ext  
sonic(config-router-bgp-Pg)# disable-connected-check
```

disable-ead-evi-rx

Description

This command enables activation of PE on EAD-ES even if EAD-EVI is not received

Parent Commands (Modes)

address-family l2vpn evpn

Syntax

```
disable-ead-evi-rx  
no disable-ead-evi-rx
```

Usage Guidelines

[no] disable-ead-evi-rx

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# address-family l2vpn evpn  
sonic(config-router-bgp-af)# disable-ead-evi-rx
```

disable-ead-evi-tx

Description

This command disables advertising EAD-EVI for local ESs

Parent Commands (Modes)

address-family l2vpn evpn

Syntax

```
disable-ead-evi-tx  
no disable-ead-evi-tx
```

Usage Guidelines

[no] disable-ead-evi-tx

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# disable-ead-evi-tx
```

disable-ebgp-connected-route-check

Description

This command disables eBGP connected route check

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
disable-ebgp-connected-route-check
no disable-ebgp-connected-route-check
```

Usage Guidelines

Use this command to disable checking if next-hop is connected on ebgp sessions. When BGP peering is between the loopback interfaces, user should enable this option.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# disable-ebgp-connected-route-check
```

distance

Description

Configures distance value to OSPFv2 routes.

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
distance { [ <admindistance> ] | { [ ospf { { [ external <extdistance> ] } | { [ inter-area <interdistance> ] } | { [ intra-area <intradistance> ] } ] } } } no distance [ ospf { [ external ] | [ inter-area ] | [ intra-area ] } ]
```

Parameters

Name	Description	Type
admindistance		Integer
extdistance		Integer
interdistance		Integer
intradistance		Integer

Usage Guidelines

Use this command to configure route distance value to OSPFv2 routes. Distance value can also be configured based on route types link inter-area routes, intra-area routes and external routes.

Examples

```
sonic-cli(config-router-ospf)# distance ospf 40  
sonic-cli(config-router-ospf)# distance intra-area 10 inter-area 20 external 30
```

Features this CLI belongs to

- OSPFv2 ## distance ### Description

Configures distance value to OSPFv3 routes.

Parent Commands (Modes)

```
router ospf6 [ vrf <vrf-name> ]
```

Syntax

```
distance { [ <admindistance> ] | { [ ospf6 { { [ external { <extdistance> { [ inter-area <interdistance> ] } { [ intra-area <intradistance> ] } ] } | { [ inter-area { <interdistance> { [ external <extdistance> ] } { [ intra-area <intradistance> ] } ] } | { [ intra-area { <intradistance> { [ external <extdistance> ] } { [ inter-area <interdistance> ] } ] } ] } } } no distance [ ospf6 ]
```

Parameters

Name	Description	Type
admindistance		Integer
extdistance		Integer
interdistance		Integer
intradistance		Integer

Usage Guidelines

Use this command to configure route distance value to OSPFv3 routes. Distance value can also be configured based on route types link inter-area routes, intra-area routes and external routes.

Examples

```
sonic-cli(config-router-ospf6)# distance ospf6 40  
sonic-cli(config-router-ospf6)# distance intra-area 10 inter-area 20 external 30
```

Features this CLI belongs to

- OSPFv3 ## distance bgp ### Description

This command changes distance value of BGP. The command allows finer control to change the distance values for external routes, internal routes and local routes separately

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
distance bgp <external> { <internal> <local> }  
no distance bgp
```

Parameters

Name	Description	Type
external		Integer
internal		Integer

Name	Description	Type
local		Integer

Usage Guidelines

Use this command to configure administrative distance for BGP route types

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# distance bgp 100 50 10
```

distance bgp

Description

Define an administrative distance

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
distance bgp <external> { <internal> <local> }
no distance bgp
```

Parameters

Name	Description	Type
external		Integer
internal		Integer
local		Integer

dont-capability-negotiate

Description

This command suppresses sending Capability Negotiation as OPEN message optional parameter to the peer. This command only affects the peer is configured other than IPv4 unicast configuration.

When remote peer does not have capability negotiation feature, remote peer will not send any capabilities at all. In that case, bgp configures the peer with configured capabilities.

You may prefer locally configured capabilities more than the negotiated capabilities even though remote peer sends capabilities. If the peer is configured by override-capability, BGP ignores received capabilities then override negotiated capabilities with configured values.

Additionally the user should be reminded that this feature fundamentally disables the ability to use widely deployed BGP features - BGP unnumbered, hostname support, AS4, Addpath, Route Refresh, ORF, Dynamic Capabilities, and graceful restart.

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
dont-capability-negotiate  
no dont-capability-negotiate
```

Usage Guidelines

Use this command to disable capability negotiation for a BGP neighbor

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 30.30.30.3  
sonic(config-router-bgp-neighbor)# dont-capability-negotiate
```

dont-capability-negotiate

Description

This command suppresses sending Capability Negotiation as OPEN message optional parameter to neighbors in a peer-group. This command only affects the peer is configured other than IPv4 unicast configuration.

When remote peer does not have capability negotiation feature, remote peer will not send any capabilities at all. In that case, bgp configures the peer with configured capabilities.

You may prefer locally configured capabilities more than the negotiated capabilities even though remote peer sends capabilities. If the peer is configured by override-capability, BGP ignores received capabilities then override negotiated capabilities with configured values.

Additionally the user should be reminded that this feature fundamentally disables the ability to use widely deployed BGP features - BGP unnumbered, hostname support, AS4, Addpath, Route Refresh, ORF, Dynamic Capabilities, and graceful restart.

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
dont-capability-negotiate  
no dont-capability-negotiate
```

Usage Guidelines

Use this command to disable capability negotiation for BGP neighbors in a peer-group

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Ext  
sonic(config-router-bgp Pg)# dont-capability-negotiate
```

dot1p

Description

This command to add DOT1P to Traffic class entry in map.

Parent Commands (Modes)

```
qos map dot1p-tc <name>
```

Syntax

```
dot1p <dot1p_list> { traffic-class <tc> }  
no dot1p <dot1p_list>
```

Parameters

Name	Description	Type
dot1p_list		String
tc		Integer

Usage Guidelines

Use this command to add entry to map DOT1P to Traffic class.

Examples

```
sonic# configure terminal
sonic(config)# dot1p 1 traffic-class 0
sonic(config)# dot1p 2 traffic-class 0
sonic(config)# dot1p 3 traffic-class 1
```

dot1x pae

Description

This command configures the port's dot1x pae role. The port can serve as either an authenticator or none.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
dot1x pae { authenticator | none }
no dot1x pae
```

Usage Guidelines

By using this command user can configure port's dot1x role.

Examples

```
sonic-clis(conf-if-Eth1/1/1)# dot1x pae { authenticator | none }
```

dot1x pae

Description

This command configures the port's dot1x pae role. The port can serve as either an authenticator or none.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
dot1x pae { authenticator | none }
no dot1x pae
```

Usage Guidelines

By using this command user can configure port's dot1x role.

Examples

```
sonic-cli(conf-if-Eth1/1/1)# dot1x pae { authenticator | none }
```

dot1x system-auth-control

Description

This command enables the dot1x authentication support on the switch. When disabled, the dot1x configuration is retained and can be changed, but is not activated.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
dot1x system-auth-control
no dot1x system-auth-control
```

Usage Guidelines

By using this command user can enable dot1x authentication support on the switch.

Examples

```
sonic(config)# dot1x system-auth-control
```

dot1x timeout quiet-period

Description

Time period in seconds that the port remains in the wait state following a failed authentication exchange with the client, before reattempting authentication.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
dot1x timeout quiet-period <time-period>
no dot1x timeout quiet-period
```

Parameters

Name	Description	Type
time-period		Integer

Usage Guidelines

Configure the dot1x quiet period that the device remains in the quiet state following a failed authentication.

Examples

```
sonic-cli(conf-if-Eth1/1/1)# dot1x timeout quiet-period
```

dot1x timeout quiet-period

Description

Time period in seconds that the port remains in the wait state following a failed authentication exchange with the client, before reattempting authentication.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
dot1x timeout quiet-period <time-period>
no dot1x timeout quiet-period
```

Parameters

Name	Description	Type
time-period		Integer

Usage Guidelines

Configure the dot1x quiet period that the device remains in the quiet state following a failed authentication.

Examples

```
sonic-clt(conf-if-Eth1/1/1)# dot1x timeout quiet-period
```

dot1x timeout server-timeout

Description

The time period in seconds that the device waits for a response from the authentication server. If the server does not send a response to an 802.1X packet within the specified period, the packet is sent again.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
dot1x timeout server-timeout <time-period>
no dot1x timeout server-timeout
```

Parameters

Name	Description	Type
time-period		Integer

Usage Guidelines

Configure the dot1x server timeout that the device waits for a response from the authentication server.

Examples

```
sonic-clt(conf-if-Eth1/1/1)# dot1x timeout server-timeout
```

dot1x timeout server-timeout

Description

The time period in seconds that the device waits for a response from the authentication server. If the server does not send a response to an 802.1X packet within the specified period, the packet is sent again.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
dot1x timeout server-timeout <time-period>
no dot1x timeout server-timeout
```

Parameters

Name	Description	Type
time-period		Integer

Usage Guidelines

Configure the dot1x server timeout that the device waits for a response from the authentication server.

Examples

```
sonic-cli(conf-if-Eth1/1/1)# dot1x timeout server-timeout
```

downstream

Description

Configure all mclags as downstream interfaces.

Parent Commands (Modes)

```
link state track <grp-name>
```

Syntax

```
downstream all-mclag  
no downstream all-mclag
```

Examples

```
sonic(config-link-track)# downstream all-mclag
```

Alternate command

```
admin@sonic:~$ sudo config linktrack update <name> --downstream all-mclag
```

drop-monitor

Description

This CLI is used to create flow configuration for drop-monitor

Parent Commands (Modes)

```
switch-resource
```

Syntax

```
drop-monitor flows <flows-name>
```

Parameters

Name	Description	Type
flows-name	min or none	Select [min(min) none(None)]

Usage Guidelines

This command is used to flow configurations for Drop-Monitor in the EM Entry Table

Examples

```
sonic(config-switch-resource)# drop-monitor flows min
```

drop-monitor

Description

Configure drop-monitor feature

Parent Commands (Modes)

```
tam
```

Syntax

```
drop-monitor
```

dropcounters

Description

Configure dropcounters

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
dropcounters <counter-name>
no dropcounters <counter-name>
```

Parameters

Name	Description	Type
counter-name	WORD	String

dscp

Description

This command to add DSCP to Traffic class entry in map.

Parent Commands (Modes)

```
qos map dscp-tc <name>
```

Syntax

```
dscp <dscp_list> { traffic-class <tc> }
no dscp <dscp_list>
```

Parameters

Name	Description	Type
dscp_list		String
tc		Integer

Usage Guidelines

Use this command to add entry to map DSCP to Traffic class.

Examples

```
sonic# configure terminal
sonic(config)# dscp 1 traffic-class 0
sonic(config)# dscp 2 traffic-class 0
sonic(config)# dscp 3 traffic-class 1
```

dst-grp

Description

This command is used to assign an existing destination group to the current subscription.

Parent Commands (Modes)

```
subscription <sub-id>
```

Syntax

```
dst-grp <dst_grp_val>
```

Parameters

Name	Description	Type
dst_grp_val	WORD	String

Usage Guidelines

Use this command to assign an existing destination group to the current subscription.

Examples

```
sonic(config-telemetry)# subscription s1
sonic(conf-tm-sub-s1)# dst-grp dg1
```

dup-addr-detection

Description

This command allows to set the threshold for address moves, including maximum moves allowed and maximum time interval

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
dup-addr-detection [ max-moves { <nummoves> { time <timevalue> } } ]  
no dup-addr-detection [ max-moves { <nummoves> { time <timevalue> } } ]
```

Parameters

Name	Description	Type
nummoves	Number of moves 2-1000, default 5	Integer
timevalue	Time in seconds 2-1800, default 180	Integer

Usage Guidelines

```
[no] dup-addr-detection max-moves {max-moves-number} time {timer-value}
```

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# address-family l2vpn evpn  
sonic(config-router-bgp-af)# dup-addr-detection max-moves 10 time 1200
```

dup-addr-detection freeze

Description

This command allows to specify the action to be taken on duplicate address detection.
It allows to configure freezing the address permanently or for a specified duration

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
dup-addr-detection freeze { permanent | <time> }  
no dup-addr-detection freeze { permanent | <time> }
```

Parameters

Name	Description	Type
time	Time in seconds 30-3600, default 180	Integer

Usage Guidelines

```
[no] dup-addr-detection freeze permanent|{freeze-time}
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# dup-addr-detection freeze permanent
```

ebgp-multipath

Description

```
This command enable multihop attribute for EBGP neighbors
```

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
ebgp-multipath [ <hop-count> ]
no ebgp-multipath [ <hop-count> ]
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

EBGP neighbors that are multiple hops away need this configuration. User can optionally set the maximum hops that BGP neighbors can be apart.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# remote-as 65100
sonic(config-router-bgp-neighbor)# ebgp-multipath 10
```

ebgp-multipath

Description

This command enables multipath for EBGP peer-group

Parent Commands (Modes)

peer-group <template-str>

Syntax

```
ebgp-multipath [ <hop-count> ]
no ebgp-multipath [ <hop-count> ]
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

peer-group with eBGP neighbors as members that are multiple hops away need this configuration. User can optionally set the maximum hops that BGP neighbors in peer-group can be apart.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp-pg)# ebgp-multipath 10
```

echo-interval

Description

Configure desired echo packet transmit interval for Bidirectional Forwarding detection(BFD) peer.

Parent Commands (Modes)

```
peer <peer_ipv4>
peer <peer_ipv6>
peer [ interface ] <interfacename>
peer [ local-address ] <local_ipv4>
peer [ local-address ] <local_ipv6>
peer [ multihop ]
peer [ vrf ] <vrfname>
```

Syntax

```
echo-interval <echo_interval>
```

Parameters

Name	Description	Type
echo_interval		Integer

Usage Guidelines

Default value is 50 milliseconds.

Examples

```
device()#configure terminal
device(config)#bfd
device(conf-bfd)# peer 192.168.0.5 interface Ethernet0
device(conf-bfd-peer)# echo-interval 200
```

echo-interval

Description

```
Configure desired echo packet transmit interval for Bidirectional Forwarding detection(BFD) peer.
```

Parent Commands (Modes)

```
profile <profilename>
```

Syntax

```
echo-interval <echo_interval>
```

Parameters

Name	Description	Type
echo_interval		Integer

Usage Guidelines

```
Default value is 50 milliseconds.
```

Examples

```
device()#configure terminal
device(config)#bfd
device(conf-bfd)# profile fast
device(conf-bfd-profile)# echo-interval 200
```

echo-mode

Description

```
Enable echo-mode for Bidirectional Forwarding detection(BFD) peer.
```

Parent Commands (Modes)

```
peer <peer_ipv4>
peer <peer_ipv6>
peer [ interface ] <interfacename>
peer [ local-address ] <local_ipv4>
peer [ local-address ] <local_ipv6>
peer [ multihop ]
peer [ vrf ] <vrfname>
```

Syntax

```
echo-mode
no echo-mode
```

Usage Guidelines

This command can be used to enable echo mode for BFD single-hop peer, echo mode is not supported for multi-hop peers.

Examples

```
device()#configure terminal
device(config)#bfd
device(conf-bfd)# peer 192.168.0.5 interface Ethernet0
device(conf-bfd-peer)# echo-mode
```

echo-mode

Description

Enable echo-mode for Bidirectional Forwarding detection(BFD) peer.

Parent Commands (Modes)

```
profile <profilename>
```

Syntax

```
echo-mode
no echo-mode
```

Usage Guidelines

This command can be used to enable echo mode for BFD single-hop peer, echo mode is not supported for multi-hop peers.

Examples

```
device()#configure terminal  
device(config)#bfd  
device(conf-bfd)# profile fast  
device(conf-bfd-profile)# echo-mode
```

ecn

Description

This command to enable ECN based on color.
This commands support options as none/green/all.
Option "none" to disable ECN. Option "green" for ECN for green color alone.
Option "all" to enable ECN on all configured colors.

Parent Commands (Modes)

```
qos wred-policy <name>
```

Syntax

```
ecn <ecn>  
no ecn
```

Parameters

Name	Description	Type
ecn	ECN Options	Select [none(ecn_none) green(ecn_green) yellow(ecn_yellow) green_yellow(ecn_green_yellow)]

Usage Guidelines

Use this command to configure ECN for color.

Examples

```
sonic(conf-wred-wred-policy)# ecn green  
sonic(conf-wred-wred-policy)# ecn yellow  
sonic(conf-wred-wred-policy)# ecn green_yellow
```

enable

Description

Enable dropcounter

Parent Commands (Modes)

dropcounters <counter-name>

Syntax

enable
no enable

enable

Description

Enable NAT feature

Parent Commands (Modes)

nat

Syntax

enable
no enable

enable

Description

This command activates IFA feature on the switch.

Parent Commands (Modes)

ifa

Syntax

enable
no enable

Usage Guidelines

This command activates IFA feature on the switch. IFA activated switches act as intermediate nodes for all IFA-tagged flows transiting the switch.

Examples

```
sonic(config)# tam
sonic(config-tam)# ifa
sonic(config-tam-ifla)# enable
sonic(config-tam-ifla)# end
sonic# show tam ifa
Status : Active
Switch ID : 9876
Enterprise ID : 8798
Version : 2.0
Number of sessions : 0
Number of collectors : 0
sonic#
```

enable

Description

This command activates Drop Monitor feature on the switch.

Parent Commands (Modes)

```
drop-monitor
```

Syntax

```
enable
no enable
```

Usage Guidelines

This command activates Drop Monitor feature on the switch.

Examples

```
sonic(config-tam-dm)# enable
sonic(config-tam-dm)# end
sonic# show tam drop-monitor
Status : Active
Switch ID : 9876
Number of sessions : 2
Number of collectors : 1
Aging Interval : 30
sonic#
```

enable

Description

This command activates Tail Stamping feature on the switch.

Parent Commands (Modes)

tail-stamping

Syntax

```
enable  
no enable
```

Usage Guidelines

This command activates Tail Stamping feature on the switch.

Examples

```
sonic# configure terminal  
sonic(config)# tam  
sonic(config-tam)# tail-stamping  
sonic(config-tam-ts)# enable  
sonic(config-tam-ts)# end  
sonic# show tam tail-stamping  
Status : Active  
Switch ID : 9876  
Number of sessions : 0  
sonic#
```

encapsulation dot1q vlan-id

Description

Configures vlan tag for routed subinterface

Parent Commands (Modes)

```
interface <phy-sub-if-name>  
interface PortChannel <lag-id-subid>
```

Syntax

```
encapsulation dot1q vlan-id <vlanid>
```

Parameters

Name	Description	Type
vlanid		Integer

Usage Guidelines

```
[no] encapsulation dot1q vlan-id <vlan-tag>
```

Examples

```
sonic-cl# configure terminal  
sonic-cl(config)# interface Ethernet 0.10  
sonic-cl(conf-subif-Ethernet0.10)# encapsulation dot1q vlan-id 100  
sonic-cl(config)#
```

```
sonic-cl# configure terminal  
sonic-cl(config)# interface Ethernet 0.10  
sonic-cl(conf-subif-Ethernet0.10)# no encapsulation  
sonic-cl(config)#
```

Features this CLI belongs to

- Subinterface

enforce-first-as

Description

This command enforces that first AS in as-path of a route received from BGP peer must be peer's AS number

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
enforce-first-as  
no enforce-first-as
```

Usage Guidelines

Use this command to enforce that first AS in as-path of route from eBGP peer must be peer's local AS number

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 30.30.30.3  
sonic(config-router-bgp-neighbor)# enforce-first-as
```

enforce-first-as

Description

This command enforces that first AS in as-path of a route received from BGP peer must be peer's AS number

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
enforce-first-as  
no enforce-first-as
```

Usage Guidelines

Use this command to enforce that first AS in as-path of route from eBGP peer must be peer's local AS number

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Ext  
sonic(config-router-bgp Pg)# enforce-first-as
```

enforce-multiphop

Description

This command enforces that eBGP neighbors perform multihop

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
enforce-multiphop  
no enforce-multiphop
```

Usage Guidelines

Use this command to enforce eBGP neighbors perform multihop

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 30.30.30.3  
sonic(config-router-bgp-neighbor)# enforce-multiphop
```

enforce-multiphop

Description

This command enforces that eBGP neighbors in a peer-group perform multihop

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
enforce-multiphop  
no enforce-multiphop
```

Usage Guidelines

Use this command to enforce eBGP neighbors in a peer-group perform multihop

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp-vg)# enforce-multipath
```

enterprise-id

Description

This command configures a 32-bit identifier that is used in the IPFIX telemetry reports.

Parent Commands (Modes)

tam

Syntax

```
enterprise-id <id>
no enterprise-id
```

Parameters

Name	Description	Type
id	1-4294967295	Integer

Usage Guidelines

This command configures a 32-bit identifier that is used in the IPFIX telemetry reports. When not configured, the last 16-bits from the system mac address are used.

Examples

```
sonic# configure terminal
sonic(config)# tam
sonic(config-tam)# enterprise-id 5678
sonic(config-tam)# exit
sonic(config)# exit
sonic# show tam switch
TAM Device information
-----
Switch ID      : 1234
Enterprise ID  : 5678
sonic#
```

entry

Description

Configure next-hop group member at specified index

Parent Commands (Modes)

pbf next-hop-group <fbn-nhgrp-name> [type { ip | ipv6 }]

Syntax

```
entry <entry-index> { next-hop <ip-address> { [ vrf { <vrf-name> | default } ] } { [ recursive
] | [ non-recursive ] | [ overlay ] ] } }
no entry <entry-index>
```

Parameters

Name	Description	Type
entry-index		Integer
ip-address	A.B.C.D	String
vrf-name	VRF name (prefixed by Vrf, Max: 15 characters)	String

entry

Description

Configure next-hop group member at specified index

Parent Commands (Modes)

pbf next-hop-group <fbn-nhgrp-name> [type { ip | ipv6 }]

Syntax

```
entry <entry-index> { next-hop <ip-address> { [ vrf { <vrf-name> | default } ] } { [ recursive
] | [ non-recursive ] | [ overlay ] ] } }
no entry <entry-index>
```

Parameters

Name	Description	Type
entry-index		Integer
ip-address	A::B	String
vrf-name	VRF name (prefixed by Vrf, Max: 15 characters)	String

entry

Description

```
Configure replication group member at specified index
```

Parent Commands (Modes)

```
pbf replication-group <fbn-replgrp-name> [ type { ip | ipv6 } ]
```

Syntax

```
entry <entry-index> { next-hop <ip-address> { [ vrf { <vrf-name> | default } ] } { [ recursive
] | [ non-recursive ] | [ overlay ] ] } [ single-copy ] }
no entry <entry-index>
```

Parameters

Name	Description	Type
entry-index		Integer
ip-address	A.B.C.D	String
vrf-name	VRF name (prefixed by Vrf, Max: 15 characters)	String

entry

Description

```
Configure replication group member at specified index
```

Parent Commands (Modes)

```
pbf replication-group <fbn-replgrp-name> [ type { ip | ipv6 } ]
```

Syntax

```
entry <entry-index> { next-hop <ip-address> { [ vrf { <vrf-name> | default } ] } { [ recursive  
] | [ non-recursive ] | [ overlay ] ] } [ single-copy ] }  
no entry <entry-index>
```

Parameters

Name	Description	Type
entry-index		Integer
ip-address	A::B	String
vrf-name	VRF name (prefixed by Vrf, Max: 15 characters)	String

errdisable recovery cause

Description

Enables error disable recovery for the given cause.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
errdisable recovery cause { [ bpduguard ] | [ link-flap ] }  
no errdisable recovery cause { [ bpduguard ] | [ link-flap ] }
```

Usage Guidelines

Use this command to enable error disable recovery for the given cause.

Examples

```
sonic-clis(config)# errdisable recovery cause udld
```

Features this CLI belongs to

- ERRDISABLE ## errdisable recovery interval ### Description

Configures error disable recovery interval.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
errdisable recovery interval <interval>
no errdisable recovery interval
```

Parameters

Name	Description	Type
interval		Integer

Usage Guidelines

Use this command to set errdisable recovery interval. Default value is 300.

Examples

```
sonic-clia(config)# errdisable recovery interval 200
```

Features this CLI belongs to

- ERRDISABLE ## event profile

Description

Select an event profile as active

Syntax

```
event profile <filename>
```

Parameters

Name	Description	Type
filename	String	String

evpn ethernet-segment

Description

Configure ethernet-segment

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
evpn ethernet-segment <es_id>
no evpn ethernet-segment <es_id>
```

Parameters

Name	Description	Type
es_id	nn:nn:nn:nn:nn:nn:nn:nn:nn or (1-16777215)	String

external-ip

Description

Command to set the external IPv4 address

Parent Commands (Modes)

```
interface vxlan <vxlan-if-name>
```

Syntax

```
external-ip <ExtIP>
no external-ip
```

Parameters

Name	Description	Type
ExtIP	A.B.C.D	String

Usage Guidelines

```
(conf-if-vxlan-vtep)# external-ip SOURCEIP  
SOURCEIP - external IPv4 address
```

Examples

```
sonic(config)# interface vxlan vtep1  
sonic(conf-if-vxlan-vtep1)# external-ip 1.1.1.2
```

fabric-external

Description

This command configures a BGP neighbor as fabric-external neighbor.

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
fabric-external  
no fabric-external
```

Usage Guidelines

Use this command to enable next-hop rewrite for fabric-external neighbor in case of multi-site.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 20.20.20.2  
sonic(config-router-bgp-neighbor)# remote-as 300  
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn  
sonic(config-router-bgp-neighbor-af)# fabric-external
```

fabric-external

Description

This command configures BGP neighbors in a peer-group as fabric-external neighbor.

Parent Commands (Modes)

address-family l2vpn evpn

Syntax

```
fabric-external  
no fabric-external
```

Usage Guidelines

Use this command to enable next-hop rewrite for fabric-external peer-group in case of multi-site.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Int  
sonic(config-router-bgp Pg)# address-family l2vpn evpn  
sonic(config-router-bgp-af)# fabric-external
```

factory

Description

This command is used to set the factory default configuration profile of a SONiC switch. This command removes the currently running switch configuration and creates a new startup configuration file using the input default configuration profile name. The newly created startup configuration is also applied as part of this command. This command may result in a loss of switch connectivity as it results in a restart of all SONiC application services.

Parent Commands (Modes)

configure terminal

Syntax

```
factory default profile <config_profile> [ confirm ]
```

Parameters

Name	Description	Type
config_profile	String	String

Usage Guidelines

```
factory default profile profile-name
```

Examples

```
sonic# configure terminal
sonic(config)# factory default profile 12
```

fallback

Description

Configure fallback mode for the portchannel

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
fallback
no fallback
```

fast-external-failover

Description

This command causes bgp to take down ebgp peers immediately when a link flaps.

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
fast-external-failover  
no fast-external-failover
```

Usage Guidelines

Use this command to control how sensitive eBGP neighborship is to the underlying link failure.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# fast-external-failover
```

fast-reboot

Description

```
fast-reboot
```

Syntax

```
fast-reboot [ force <_force> ] [ verbose <_verbose> ] [ ignore <_ignore> ] [ sbin-reboot  
<_sbin_reboot> ] [ noprompt <_noprompt> ]
```

Parameters

Name	Description	Type
_force	[yes/no]	Select [yes no]
_verbose	[yes/no]	Select [yes no]
_ignore	[yes/no]	Select [yes no]
_sbin_reboot	[yes/no]	Select [yes no]
_noprompt	[yes/no]	Select [yes no]

fast_rate

Description

Configure LACP fast rate mode for the portchannel

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
fast_rate  
no fast_rate
```

fec

Description

Configure FEC (forward error correction)

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
fec <fec>  
no fec
```

Parameters

Name	Description	Type
fec	FEC (forward error correction) mode	Select [fc rs none]

fec

Description

Configure FEC (forward error correction)

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
fec <fec>
no fec
```

Parameters

Name	Description	Type
fec	FEC (forward error correction) mode	Select [fc rs none]

filter-list

Description

This command configures a filter list for a BGP neighbor

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
filter-list <fname> { in | out }
no filter-list <fname> { in | out }
```

Parameters

Name	Description	Type
fname	WORD	String

Usage Guidelines

Use this command to define policy (route filtering) for a BGP neighbor in outbound or/and inbound direction.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# filter-list fl_allow_remote in
```

filter-list

Description

This command configures a filter list for BGP peer-group

Parent Commands (Modes)

address-family ipv4 unicast

Syntax

```
filter-list <fname> { in | out }
no filter-list <fname> { in | out }
```

Parameters

Name	Description	Type
fname	WORD	String

Usage Guidelines

Use this command to define policy (route filtering) for a BGP peer-group in outbound or/and inbound direction.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp-pg)# address-family ipv4 unicast
sonic(config-router-bgp-pg-af)# filter-list fl_allow_remote in
```

filter-list

Description

Establish BGP filters

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
filter-list <fname> { in | out }
no filter-list <fname> { in | out }
```

Parameters

Name	Description	Type
fname	WORD	String

filter-list

Description

Establish BGP filters

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
filter-list <fname> { in | out }
no filter-list <fname> { in | out }
```

Parameters

Name	Description	Type
fname	WORD	String

flow-group

Description

This command configures flow group by specifying a set of match criteria that defines a set of flows that are of interest.

Parent Commands (Modes)

tam

Syntax

```
flow-group <name> [ src-ip <src_ip> ] [ dst-ip <dst_ip> ] [ l4-src-port <l4_src_port> ] [ l4-dst-port <l4_dst_port> ] [ priority <priority_value> ] [ protocol <protocol_value> ] [ src-mac <src_mac> ] [ dst-mac <dst_mac> ] [ vlan-id <vlan> ] [ ethertype { ETHERTYPE_IPV4 | ETHERTYPE_IPV6 | ETHERTYPE_ARP | ETHERTYPE_VLAN | ETHERTYPE_LLDP | ETHERTYPE_MPLS | ETHERTYPE_ROCE } ]
no flow-group <name>
```

Parameters

Name	Description	Type
name	WORD	String
src_ip	A.B.C.D/mask or A::B/mask	String
dst_ip	A.B.C.D/mask or A::B/mask	String
l4_src_port		Integer
l4_dst_port		Integer
priority_value		Integer
protocol_value	Transport Protocol Method	Select [UDP(IP_UDP) TCP(IP_TCP)]
src_mac	MACADDRESS	String
dst_mac	MACADDRESS	String
vlan		Integer

Usage Guidelines

This command configures flow group by specifying a set of match criteria that defines a set of flows that are of interest.

Examples

```
sonic(config-tam)# flow-group f9 type ipv4 src-ip 192.1.2.3 dst-ip 172.6.5.4
sonic# show tam flowgroups
Flow Group Name      : f9
  Id                : 60
  Priority          : 100
  SRC IP            : 192.1.2.3/32
  DST IP            : 172.6.5.4/32
Packet Count         : 5432
Flow Group Name      : DEMO
  Id                : 1
  Priority          : 100
  SRC IP            : 1.1.1.1/32
  DST IP            : 4.4.4.4/32
Packet Count         : 454
sonic#
```

flow-group

Description

This command attaches an existing flow group to an interface.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
flow-group <name>
no flow-group <name>
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

This command attaches an existing flow group to an interface.

Examples

```

sonic# configure terminal
sonic(config)# interface Ethernet 46
sonic(conf-if-Ethernet46)# flow-group f1
sonic(conf-if-Ethernet46)# end
sonic# show tam flowgroups
Flow Group Name      : DEMO
  Id                 : 1
  Priority          : 100
  SRC MAC           : 11:22:33:44:55:66
  SRC IP            : 1.1.1.1/32
  DST IP            : 4.4.4.4/32
Packet Count         :
Flow Group Name      : f88
  Id                 : 409
  Priority          : 100
  SRC IP            : 1.1.1.1/32
  DST IP            : 5.5.5.5/32
Packet Count         :
Flow Group Name      : f1
  Id                 : 717
  Priority          : 100
  SRC MAC           : 00:00:00:00:00:01
  DST MAC           : 00:00:00:00:00:02
  SRC IP            : 3.3.3.3/32
  DST IP            : 4.4.4.4/32
  Ingress Intf     : Ethernet46
Packet Count         :
sonic#

```

frequency

Description

Configure frequency of probe for an IP SLA instance

Parent Commands (Modes)

`ip sla <sla-id>`

Syntax

```

frequency <freq-value>
no frequency

```

Parameters

Name	Description	Type
freq-value		Integer

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-10)# frequency 45
```

global

Description

This command is used to configure the telemetry dialout client global configuration.

Parent Commands (Modes)

```
telemetry
```

Syntax

```
global <encoding> <retry_interval> <src_ip> <unidirectional>
no global
```

Parameters

Name	Description	Type
encoding	WORD	String
retry_interval		Integer
src_ip	A.B.C.D/A::B	String
unidirectional	Boolean choice	Select [true(1) false(0)]

Usage Guidelines

Use this command to configure the telemetry dialout client global.

Examples

```
sonic(config-telemetry)# global JSON_IETF 30 1.1.1.1 true
```

graceful-restart

Description

Configures OSPFv2 Graceful Restart.

Parent Commands (Modes)

router ospf [vrf <vrf-name>]

Syntax

```
graceful-restart { { [ grace-period <grace-period-val> ] } | { [ helper { { enable [ <neighbor-ip> ] } | [ strict-lsa-checking ] | [ planned-only ] | { [ supported-grace-time <supported-grace-time-val> ] } } ] } }
no graceful-restart { { [ grace-period [ <grace-period-val> ] ] } | { [ helper { { enable [ <neighbor-ip> ] } | [ strict-lsa-checking ] | [ planned-only ] | { [ supported-grace-time [ <supported-grace-time-val> ] ] } } ] } }
```

Parameters

Name	Description	Type
grace-period-val		Integer
neighbor-ip	A.B.C.D	String
supported-grace-time-val		Integer

Usage Guidelines

Configure Graceful Restart (RFC 3623).

Examples

```
sonic-cli(config-router-ospf)# graceful-restart
sonic-cli(config-router-ospf)# graceful-restart grace-time 120
sonic-cli(config-router-ospf)# graceful-restart helper enable
sonic-cli(config-router-ospf)# graceful-restart helper enable 192.168.1.1
sonic-cli(config-router-ospf)# graceful-restart helper strict-lsa-checking
sonic-cli(config-router-ospf)# graceful-restart helper planned-only
sonic-cli(config-router-ospf)# graceful-restart helper supported-grace-time 120
```

Features this CLI belongs to

- OSPFv2 ## graceful-restart enable ### Description

This command enables Graceful Restart for an instance of BGP

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
graceful-restart enable  
no graceful-restart enable
```

Usage Guidelines

Use this command to enable BGP Graceful Restart globally in an instance of BGP. Changing the Graceful restart parameter will take effect only on the fly will not take effect immediately. It will require all the BGP neighbors to be reset to take effect. This is because Graceful Restart capability must be negotiated with neighbors to make this feature functional.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# graceful-restart enable
```

graceful-restart preserve-fw-state

Description

This command instructs BGP to preserve forwarding state during Graceful Restart for an instance of BGP

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
graceful-restart preserve-fw-state  
no graceful-restart preserve-fw-state
```

Usage Guidelines

Use this command to enable BGP to preserve forwarding state of BGP during Graceful Restart.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# graceful-restart preserve-fw-state
```

graceful-restart restart-time

Description

This command configures restart timer interval for BGP

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
graceful-restart restart-time <restart-time>
no graceful-restart restart-time
```

Parameters

Name	Description	Type
restart-time		Integer

Usage Guidelines

Use this command to configure BGP restart timer interval in seconds. This is optional parameter and determines how long peer routers will wait to delete stale routes before a BGP open message is received. The default value is 120 seconds.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# graceful-restart restart-time 180
```

graceful-restart stalepath-time

Description

This command configures stale path timer interval for BGP

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
graceful-restart stalepath-time <stalepath-time>
no graceful-restart stalepath-time
```

Parameters

Name	Description	Type
stalepath-time		Integer

Usage Guidelines

This command is used to set the maximum time to hold on to the stale paths of a gracefully restarted peer. All stale paths are deleted after the expiration of this timer. This is an optional parameter. The default is 360 seconds

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# graceful-restart stalepath-time 300
```

graceful-shutdown

Description

Enable graceful shutdown for the portchannel

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [
    fallback ] [ fast_rate ]
```

Syntax

```
graceful-shutdown  
no graceful-shutdown
```

graceful-shutdown

Description

```
Enable graceful shutdown for the portchannels
```

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]  
interface range po_range_num
```

Syntax

```
graceful-shutdown  
no graceful-shutdown
```

graceful-shutdown

Description

```
This command enables Graceful shutdown feature
```

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
graceful-shutdown  
no graceful-shutdown
```

Usage Guidelines

```
Use this command to gracefully remove a BGP router from service. This  
command will instruct BGP to enter into graceful shutdown mode by  
resending routes with GSHUT community to all it's neighbors. This will  
enable all it's neighbor to route traffic around it so that router can  
be taken out of service without impact data forwarding
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# graceful-shutdown
```

green

Description

This command to configure WRED minimum, maximum thresholds and drop probability for color green.

Parent Commands (Modes)

```
qos wred-policy <name>
```

Syntax

```
green minimum-threshold { <min-threshold> { maximum-threshold <max-threshold> {
    drop-probability <max-drop-rate> } } }
```

Parameters

Name	Description	Type
min-threshold		Integer
max-threshold		Integer
max-drop-rate		Integer

Usage Guidelines

Use this command to configure WRED minimum, maximum and drop probability for green color packets.

Examples

```
sonic(conf-wred-wred-green)# color green minimum-threshold 100 maximum-threshold 200
    drop-probability 50
```

group

Description

```
Add dropcounter group
```

Parent Commands (Modes)

```
dropcounters <counter-name>
```

Syntax

```
group <group-str>
no group
```

Parameters

Name	Description	Type
group-str	WORD	String

grpc port

Description

```
Sets the GRPC port.
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
grpc port <port_val>
```

Parameters

Name	Description	Type
port_val		Integer

Usage Guidelines

```
sonic-cli(config)# grpc port <value>
```

Examples

```
sonic-cli(config)# grpc port 10000
sonic-cli(config)#
```

hostname

Description

Configures hostname of the switch.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
hostname <hostname-val>
no hostname
```

Parameters

Name	Description	Type
hostname-val	WORD	String

Usage Guidelines

```
sonic-cli(config)# hostname <host-name>
```

Examples

```
sonic-cli(config)# hostname R1
sonic-cli(config)#
```

icmp-echo

Description

```
Configure operation type as ICMP and destination IP for an IP SLA instance
```

Parent Commands (Modes)

```
ip sla <sla-id>
```

Syntax

```
icmp-echo <addr>
no icmp-echo
```

Parameters

Name	Description	Type
addr	String	String

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-10)# icmp-echo 30.30.1.2
```

ifa

Description

```
Configure inband flow analyzer feature
```

Parent Commands (Modes)

```
tam
```

Syntax

```
ifa
```

ignore server-key

Description

This command configures the device to ignore the server key.

Parent Commands (Modes)

aaa server radius dynamic-author

Syntax

```
ignore server-key  
no ignore server-key
```

Usage Guidelines

This command configures the device to ignore the server key.

Examples

```
sonic-cl(i(config)# aaa server radius dynamic-author  
sonic-cl(i(config-radius-da)# ignore server-key  
sonic-cl(i(config-radius-da)#

```

ignore session-key

Description

This command configures the device to ignore the session key.

Parent Commands (Modes)

aaa server radius dynamic-author

Syntax

```
ignore session-key  
no ignore session-key
```

Usage Guidelines

This command configures the device to ignore the session key.

This command execution fails when the auth-type set to "session-key" as the authentication can happen only based on the session-key attribute.

Examples

```
sonic-cl(i(config)# aaa server radius dynamic-author
sonic-cl(i(config-radius-da)# ignore session-key
sonic-cl(i(config-radius-da)#

```

import vrf

Description

This command imports all routes or selective routes based on route-map from other VRF.

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
import vrf { { route-map <route-map-name> } | <import-vrf-name> }
no import vrf { { [ route-map <route-map-name> ] } | [ <import-vrf-name> ] } ]
```

Parameters

Name	Description	Type
route-map-name	WORD	String
import-vrf-name	Comma separated words(Without space)	String

Usage Guidelines

Use this command to import routes from other VRF.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# import vrf Vrf1
      or
sonic(config-router-bgp-af)# import vrf route-map map1
```

import vrf

Description

```
Import routes from another VRF
```

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
import vrf { { route-map <route-map-name> } | <import-vrf-name> }
no import vrf { { [ route-map <route-map-name> ] } | [ <import-vrf-name> ] }
```

Parameters

Name	Description	Type
route-map-name	WORD	String
import-vrf-name	Comma separated words(Without space)	String

interface

Description

```
Configure interfaces
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
interface { <phy-if-name> | <vlan-if-name> | <phy-sub-if-name> }
no interface { <vlan_range_num> | <po_range_num> | <phy-sub-if-name> }
```

Parameters

Name	Description	Type
phy-if-name	EthernetNUM	
vlan-if-name	VlanNUM	

Name	Description	Type
phy-sub-if-name	EthernetX.Y	

interface CPU

Description

CPU Interface commands

Parent Commands (Modes)

configure terminal

Syntax

interface CPU

interface Loopback

Description

Loopback interface configuration

Parent Commands (Modes)

configure terminal

Syntax

interface Loopback <lo-id>
no interface Loopback <lo-id>

Parameters

Name	Description	Type
lo-id		Integer

interface Management

Description

Management interface commands

Parent Commands (Modes)

configure terminal

Syntax

```
interface Management <mgmt-if-id>
```

Parameters

Name	Description	Type
mgmt-if-id		Integer

interface PortChannel

Description

PortChannel interface configuration

Parent Commands (Modes)

configure terminal

Syntax

```
interface PortChannel { { <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ] } | <lag-id-subid> }
```

Parameters

Name	Description	Type
lag-id		Integer
PoMode	PortChannel Mode	Select [active on]
min-links-value		Integer
lag-id-subid	X.Y	String

interface breakout

Description

Breakout a port

Parent Commands (Modes)

configure terminal

Syntax

```
interface breakout port <slotport> mode { <breakout_mode> { [ total-lane-count <lanecount> ] } }  
no interface breakout port <slotport>
```

Parameters

Name	Description	Type
slotport	Front-panel port (slot/port)	Select [1x100G 1x40G 2x100G 2x50G 4x100G 4x25G 4x10G 1x400G 2x200G 4x50G 8x50G 8x25G 8x10G 2x40G 1x50G 1x25G 1x20G 1x10G]
breakout_mode	Port breakout mode	Integer

Usage Guidelines

Use this command breakout a port or undo a breakout configuration.
interface breakout port front-panel-port mode breakout-mode
no interface breakout port front-panel-port

Examples

```
sonic# configure terminal  
sonic(config)# interface breakout port 1/1 mode 4x25  
Dynamic Port Breakout in-progress, use 'show interface breakout port 1/1' to check status.  
sonic(config)# no interface breakout port 1/1  
Dynamic Port Breakout in-progress, use 'show interface breakout port 1/1' to check status.  
sonic-cli(config)# interface breakout port 1/31 mode 2x50G total-lane-count 4  
Dynamic Port Breakout in-progress, use 'show interface breakout port 1/31' to check status.
```

interface media-fec

Description

Configure Media FEC on a port

Parent Commands (Modes)

configure terminal

Syntax

```
interface media-fec port <slotport> mode <media-fec-mode>
no interface media-fec port <slotport>
```

Parameters

Name	Description	Type
slotport media-fec-mode	Front-panel port (slot/port) Media FEC (forward error correction) mode	Select [ieee custom]

Usage Guidelines

Use this command to configure media-fec a port or undo media-fec configuration.
interface media-fec port front-panel-port mode media-fec-mode
no interface media-fec port front-panel-port

Examples

```
sonic# configure terminal
sonic(config)# interface media-fec port 1/1 mode custom
media-fec on port in-progress, use 'show interface interface-name' to check status.
sonic(config)# no interface media-fec port 1/1
media-fec on port in-progress, use 'show interface interface-name' to check status.
```

interface range

Description

Config interface range

Parent Commands (Modes)

configure terminal

Syntax

```
interface range { iface_range_num | vlan_range_num | po_range_num | { create { vlan_range_num |  
    { po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [  
        fast_rate ] } } } }
```

Parameters

Name	Description	Type
PoMode	PortChannel Mode	Select [active on]
min-links-value		Integer

interface transceiver diagnostics

Description

Transceiver diagnostics configuration

Parent Commands (Modes)

configure terminal

Syntax

```
interface transceiver diagnostics { { loopback { [ host-side-input ] | [ host-side-output ] | [  
    media-side-input ] | [ media-side-output ] ] } } | { pattern { [ checker-host ] | [  
    checker-media ] | [ generator-host ] | [ generator-media ] ] } } } [ <phy-if-name> ]  
no interface transceiver diagnostics { { loopback { [ host-side-input ] | [ host-side-output ]  
    | [ media-side-input ] | [ media-side-output ] ] } } | { pattern { [ checker-host ] | [  
    checker-media ] | [ generator-host ] | [ generator-media ] ] } } } [ <phy-if-name> ]
```

Parameters

Name	Description	Type
phy-if-name	Interface Ethernet Range	

interface vxlan

Description

Command to enter VXLAN configuration mode.

Parent Commands (Modes)

configure terminal

Syntax

```
interface vxlan <vxlan-if-name>
no interface vxlan <vxlan-if-name>
```

Parameters

Name	Description	Type
vxlan-if-name		String

Usage Guidelines

```
(config)# interface vxlan VTEPNAME
VTEPNAME - string prefixed with 'vtep' with max size of 10 chars.
```

Examples

```
sonic(config)# interface vxlan vtep1
sonic(config-if-vtep1)#[
```

ip

Description

This command is used to configure an existing telemetry destination group.

Parent Commands (Modes)

destination-group <dg-id>

Syntax

```
ip address <address_val> port <port_val> [ protocol <prtocol_val> ] [ encoding <encoding_val> ]
    [ src-ip <src_ip> ] [ vrf { mgmt } ]
```

Parameters

Name	Description	Type
address_val	A.B.C.D/A::B	String
port_val		Integer
prtocol_val	WORD	String
encoding_val	WORD	String
src_ip	A.B.C.D/A::B	String

Usage Guidelines

Use this command to configure the address and port (protocol and encoding not yet supported, parameters present for compatibility) of a destination group.

Examples

```
sonic(config-telemetry)# destination-group dg1
sonic(conf-tm-dest-dg1)# ip address 1.1.1.1 port 1111 protocol gRPC encoding GPB src-ip
    192.168.122.11 vrf mgmt
```

ip access-group

Description

Apply IPv4 ACL to an interface.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip access-group <access-list-name> { in | out }
no ip access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	WORD	String

Usage Guidelines

ACL must be created first and must be of type IPv4 to be applied. Only 1 ACL of a given type can be applied per interface and per direction.

Examples

```
sonic(conf-if-po1)# ip access-group ipacl-example out
```

ip access-group

Description

Apply IPv4 ACL to an interface.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip access-group <access-list-name> { in | out }
no ip access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	WORD	String

Usage Guidelines

ACL must be created first and must be of type IPv4 to be applied. Only 1 ACL of a given type can be applied per interface and per direction.

Examples

```
sonic(conf-if-po1)# ip access-group ipacl-example out
```

ip access-group

Description

Apply IPv4 ACL to an interface.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip access-group <access-list-name> { in | out }
no ip access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	WORD	String

Usage Guidelines

ACL must be created first and must be of type IPv4 to be applied. Only 1 ACL of a given type can be applied per interface and per direction.

Examples

```
sonic(conf-if-po1)# ip access-group ipacl-example out
```

ip access-group

Description

Apply IPv4 ACL to an interface.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip access-group <access-list-name> { in | out }
no ip access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	WORD	String

Usage Guidelines

ACL must be created first and must be of type IPv4 to be applied. Only 1 ACL of a given type can be applied per interface and per direction.

Examples

```
sonic(conf-if-po1)# ip access-group ipacl-example out
```

ip access-group

Description

Apply IPv4 ACL globally

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip access-group <access-list-name> { in | out }
no ip access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	String(Max: 72 characters)	String

Usage Guidelines

ACL must be created first and must be of type IPv4 to be applied. Only 1 ACL of a given type can be applied globally per direction.

Examples

```
sonic(config)# ip access-group ipacl-example in
```

ip access-list

Description

Create IPv4 ACL

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip access-list <access-list-name>
no ip access-list <access-list-name>
```

Parameters

Name	Description	Type
access-list-name	String(Max: 72 characters)	String

Usage Guidelines

ACL name can be of maximum 63 characters. The name must begin with A-Z, a-z or 0-9. Underscore and hypens can be used except as the first character. ACL name must be unique across all ACL types.

Examples

```
sonic(config)# ip access-list ipacl-example
```

ip address

Description

IP address

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip address <addr> { [ secondary ] }  
no ip address [ <addr> { [ secondary ] } ]
```

Parameters

Name	Description	Type
addr	A.B.C.D/mask	String

ip address

Description

IP address

Parent Commands (Modes)

```
interface Management <mgmt-if-id>
```

Syntax

```
ip address <addr> { { [ gwaddr <gw_addr> ] } | [ secondary ] }  
no ip address [ <addr> { [ secondary ] } ]
```

Parameters

Name	Description	Type
addr	A.B.C.D/mask	String
gw_addr	A.B.C.D	String

ip address

Description

IP address

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip address <addr> { [ secondary ] }  
no ip address [ <addr> { [ secondary ] } ]
```

Parameters

Name	Description	Type
addr	A.B.C.D/mask	String

ip address

Description

IP address

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip address <addr> { [ secondary ] }  
no ip address [ <addr> { [ secondary ] } ]
```

Parameters

Name	Description	Type
addr	A.B.C.D/mask	String

ip address

Description

IP address

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ip address <addr> { [ secondary ] }  
no ip address [ <addr> { [ secondary ] } ]
```

Parameters

Name	Description	Type
addr	A.B.C.D/mask	String

ip address

Description

IP address

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip address <addr> { [ secondary ] } }
no ip address [ <addr> { [ secondary ] } ]
```

Parameters

Name	Description	Type
addr	A.B.C.D/mask	String

ip anycast-address

Description

Configures IPv4 Static Anycast Gateway Address for an Interface.

Parent Commands (Modes)

interface <vlan-if-name>

Syntax

```
ip anycast-address <anycast-addr>
no ip anycast-address <anycast-addr>
```

Parameters

Name	Description	Type
anycast-addr	A.B.C.D/mask	String

Examples

```
sonic(conf-if-Vlan5)# ip anycast-address 50.0.0.1/24
```

ip anycast-address

Description

Configures IPv4 Static Anycast Gateway Address for an Ethernet sub-interface.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip anycast-address <anycast-addr>
no ip anycast-address <anycast-addr>
```

Parameters

Name	Description	Type
anycast-addr	A.B.C.D/mask	String

Examples

```
sonic(conf-if-Eth5.1)# ip anycast-address 50.0.0.1/24
```

ip anycast-address

Description

Enable/Disable IPv4 Static Anycast Gateway functionality. By default it is enabled.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip anycast-address { enable | disable }
```

Examples

```
sonic(config)# ip anycast-address enable
```

```
sonic(config)# ip anycast-address disable
```

ip anycast-mac-address

Description

Configures MAC address for all the Static Anycast Gateway Addresses on the system.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip anycast-mac-address <anycast-mac>
no ip anycast-mac-address <anycast-mac>
```

Parameters

Name	Description	Type
anycast-mac	nn:nn:nn:nn:nn:nn	String

Examples

```
sonic(conf-if-Vlan5)# ip anycast-mac-address 00:22:33:44:55:66
```

ip arp

Description

Configures time in seconds for ARP cache entry to timeout.

Parent Commands (Modes)

configure terminal

Syntax

```
ip arp timeout <value>
no ip arp timeout
```

Parameters

Name	Description	Type
value	value in seconds	Integer

Usage Guidelines

Use this command to configure IPv4 ARP cache entry timeout.

Examples

```
sonic-cli(config)# ip arp timeout 200
```

ip arp

Description

Configure static ARP.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
ip arp <static-ip> <neigh>
no ip arp <static-ip> <neigh>
```

Parameters

Name	Description	Type
static-ip	A.B.C.D	String
neigh	nn:nn:nn:nn:nn:nn	String

Usage Guidelines

Use this command to configure static ARP.

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# arp ip_addr mac
```

ip arp

Description

Configure static ARP.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip arp <static-ip> <neigh>
no ip arp <static-ip> <neigh>
```

Parameters

Name	Description	Type
static-ip	A.B.C.D	String
neigh	nn:nn:nn:nn:nn:nn	String

Usage Guidelines

Use this command to configure static ARP.

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# arp ip_addr mac
```

ip arp

Description

Configure static ARP.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip arp <static-ip> <neigh>
no ip arp <static-ip> <neigh>
```

Parameters

Name	Description	Type
static-ip	A.B.C.D	String
neigh	nn:nn:nn:nn:nn:nn	String

Usage Guidelines

Use this command to configure static ARP.

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# arp ip_addr mac
```

ip arp

Description

Configure static ARP.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip arp <static-ip> <neigh>
no ip arp <static-ip> <neigh>
```

Parameters

Name	Description	Type
static-ip	A.B.C.D	String
neigh	nn:nn:nn:nn:nn:nn	String

Usage Guidelines

Use this command to configure static ARP.

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# arp ip_addr mac
```

ip dhcp smart-relay

Description

Enable DHCPv4 smart-relay globally

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip dhcp smart-relay
no ip dhcp smart-relay
```

Usage Guidelines

```
ip dhcp smart-relay
```

Examples

```
sonic-cli# configure terminal
sonic-cli(config)# ip dhcp smart-relay
sonic-cli(config)#
```

```
sonic-cli# configure terminal
sonic-cli(config)# no ip dhcp smart-relay
sonic-cli(config)#
```

Features this CLI belongs to

- DHCP smart-relay ## ip dhcp snooping ### Description

```
Enable DHCPv4 snooping globally
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip dhcp snooping
no ip dhcp snooping
```

Usage Guidelines

```
[no] ip dhcp snooping
```

Examples

```
sonic-cli# configure terminal
sonic-cli(config)# ip dhcp snooping
sonic-cli(config)#
```

```
sonic-cli# configure terminal
sonic-cli(config)# no ip dhcp snooping
sonic-cli(config)#
```

Features this CLI belongs to

- DHCP Snooping ## ip dhcp snooping trust ### Description

```
Mark Trust mode for DHCPv4 snooping on interface(s)
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip dhcp snooping trust
no ip dhcp snooping trust
```

ip dhcp snooping trust

Description

```
Mark Trust mode for DHCPv4 snooping on interface(s)
```

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [
    fallback ] [ fast_rate ]
```

Syntax

```
ip dhcp snooping trust
no ip dhcp snooping trust
```

ip dhcp snooping trust

Description

```
Mark Trust mode for DHCPv4 snooping on interface(s)
```

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
ip dhcp snooping trust  
no ip dhcp snooping trust
```

ip dhcp snooping trust

Description

```
Mark Trust mode for DHCPv4 snooping on interface(s)
```

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]  
interface range po_range_num
```

Syntax

```
ip dhcp snooping trust  
no ip dhcp snooping trust
```

ip dhcp snooping verify mac-address

Description

```
Enable DHCPv4 snooping mac verification
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip dhcp snooping verify mac-address  
no ip dhcp snooping verify mac-address
```

Usage Guidelines

```
[no] ip dhcp snooping verify mac-address
```

Examples

```
sonic-cli# configure terminal
sonic-cli(config)# ip dhcp snooping verify mac-address
sonic-cli(config)#
```

```
sonic-cli# configure terminal
sonic-cli(config)# no ip dhcp snooping verify mac-address
sonic-cli(config)#
```

Features this CLI belongs to

- DHCP Snooping ## ip dhcp snooping vlan ##### Description

```
Enable DHCPv4 snooping on vlan(s)
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip dhcp snooping vlan <vlan-range>
no ip dhcp snooping vlan <vlan-range>
```

Parameters

Name	Description	Type
vlan-range	<1..4094>	String

ip dhcp-relay

Description

```
Configures DHCPv4 relay on an interface.
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] }
```

```
no ip dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } ]
```

Parameters

Name	Description	Type
ipaddr1	A.B.C.D	String
vrfName	WORD	String
ipaddr2	A.B.C.D	String
ipaddr3	A.B.C.D	String
ipaddr4	A.B.C.D	String

Usage Guidelines

Use this command to configure a DHCPv4 relay on an interface.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay 9.0.0.9
```

ip dhcp-relay

Description

Configures DHCPv4 relay on an interface.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] }
```

```
no ip dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } ]
```

Parameters

Name	Description	Type
ipaddr1	A.B.C.D	String
vrfName	WORD	String
ipaddr2	A.B.C.D	String
ipaddr3	A.B.C.D	String
ipaddr4	A.B.C.D	String

Usage Guidelines

Use this command to configure a DHCPv4 relay on an interface.

Examples

```
sonic-clia(config)# interface Ethernet 12
sonic-clia(conf-if-Ethernet0)# ip dhcp-relay 9.0.0.9
```

ip dhcp-relay

Description

Configures DHCPv4 relay on an interface.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] }
```

```
no ip dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } ]
```

Parameters

Name	Description	Type
ipaddr1	A.B.C.D	String
vrfName	WORD	String

Name	Description	Type
ipaddr2	A.B.C.D	String
ipaddr3	A.B.C.D	String
ipaddr4	A.B.C.D	String

Usage Guidelines

Use this command to configure a DHCPv4 relay on an interface.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay 9.0.0.9
```

ip dhcp-relay

Description

Configures DHCPv4 relay on a range of interfaces.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
ip dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ]
] } ] } }
no ip dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } ]
```

Parameters

Name	Description	Type
ipaddr1	A.B.C.D	String
vrfName	WORD	String
ipaddr2	A.B.C.D	String
ipaddr3	A.B.C.D	String
ipaddr4	A.B.C.D	String

Usage Guidelines

Use this command to configure a DHCPv4 relay on a range of interfaces.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay 9.0.0.9
```

ip dhcp-relay

Description

Configures DHCPv4 relay on a range of interfaces.

Parent Commands (Modes)

```
interface range create vlan_range_num
interface range vlan_range_num
```

Syntax

```
ip dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ]
    ] } ] } }
no ip dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] }
```

Parameters

Name	Description	Type
ipaddr1	A.B.C.D	String
vrfName	WORD	String
ipaddr2	A.B.C.D	String
ipaddr3	A.B.C.D	String
ipaddr4	A.B.C.D	String

Usage Guidelines

Use this command to configure a DHCPv4 relay on a range of interfaces.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay 9.0.0.9
```

ip dhcp-relay

Description

Configures DHCPv4 relay on a range of interfaces.

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [
    fallback ] [ fast_rate ]
interface range po_range_num
```

Syntax

```
ip dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ]
    ] } ] } }
no ip dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } ]
```

Parameters

Name	Description	Type
ipaddr1	A.B.C.D	String
vrfName	WORD	String
ipaddr2	A.B.C.D	String
ipaddr3	A.B.C.D	String
ipaddr4	A.B.C.D	String

Usage Guidelines

Use this command to configure a DHCPv4 relay on a range of interfaces.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay 9.0.0.9
```

ip dhcp-relay

Description

Configures DHCPv4 relay on an interface.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ]
    ] } ] } }
no ip dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } ]
```

Parameters

Name	Description	Type
ipaddr1	A.B.C.D	String
vrfName	WORD	String
ipaddr2	A.B.C.D	String
ipaddr3	A.B.C.D	String
ipaddr4	A.B.C.D	String

Usage Guidelines

Use this command to configure a DHCPv4 relay on an interface.

Examples

```
sonic-clia(config)# interface Ethernet 12
sonic-clia(conf-if-Ethernet0)# ip dhcp-relay 9.0.0.9
```

ip dhcp-relay link-select

Description

Configure the link-selection suboption on an interface.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip dhcp-relay link-select  
no ip dhcp-relay link-select
```

Usage Guidelines

Use this command to configure the link-selection suboption on an interface.

Examples

```
sonic-cli(config)# interface Ethernet 12  
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay link-select
```

ip dhcp-relay link-select

Description

Configure the link-selection suboption on an interface.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip dhcp-relay link-select  
no ip dhcp-relay link-select
```

Usage Guidelines

Use this command to configure the link-selection suboption on an interface.

Examples

```
sonic-cli(config)# interface Ethernet 12  
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay link-select
```

```
ip dhcp-relay link-select
```

Description

Configure the link-selection suboption on an interface.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip dhcp-relay link-select  
no ip dhcp-relay link-select
```

Usage Guidelines

Use this command to configure the link-selection suboption on an interface.

Examples

```
sonic-clia(config)# interface Ethernet 12  
sonic-clia(conf-if-Ethernet0)# ip dhcp-relay link-select
```

```
ip dhcp-relay link-select
```

Description

Configure the link-selection suboption on a range of interfaces.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
ip dhcp-relay link-select  
no ip dhcp-relay link-select
```

Usage Guidelines

Use this command to configure the link-selection suboption on a range of interfaces.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay link-select
```

ip dhcp-relay link-select

Description

Configure the link-selection suboption on a range of interfaces.

Parent Commands (Modes)

```
interface range create vlan_range_num
interface range vlan_range_num
```

Syntax

```
ip dhcp-relay link-select
no ip dhcp-relay link-select
```

Usage Guidelines

Use this command to configure the link-selection suboption on a range of interfaces.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay link-select
```

ip dhcp-relay link-select

Description

Configure the link-selection suboption on a range of interfaces.

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
interface range po_range_num
```

Syntax

```
ip dhcp-relay link-select
no ip dhcp-relay link-select
```

Usage Guidelines

Use this command to configure the link-selection suboption on a range of interfaces.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay link-select
```

ip dhcp-relay link-select

Description

Configure the link-selection suboption on an interface.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip dhcp-relay link-select
no ip dhcp-relay link-select
```

Usage Guidelines

Use this command to configure the link-selection suboption on an interface.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay link-select
```

ip dhcp-relay max-hop-count

Description

Configures the maximum hop count for the DHCP packet.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
ip dhcp-relay max-hop-count <hop-count>
no ip dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCP packet.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay max-hop-count 9
```

ip dhcp-relay max-hop-count

Description

Configures the maximum hop count for the DHCP packet.

Parent Commands (Modes)

interface <vlan-if-name>

Syntax

```
ip dhcp-relay max-hop-count <hop-count>
no ip dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCP packet.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay max-hop-count 9
```

ip dhcp-relay max-hop-count

Description

Configures the maximum hop count for the DHCP packet.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip dhcp-relay max-hop-count <hop-count>
no ip dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCP packet.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay max-hop-count 9
```

ip dhcp-relay max-hop-count

Description

Configures the maximum hop count for the DHCP packet.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
ip dhcp-relay max-hop-count <hop-count>
no ip dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCP packet.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay max-hop-count 9
```

ip dhcp-relay max-hop-count

Description

Configures the maximum hop count for the DHCP packet.

Parent Commands (Modes)

```
interface range create vlan_range_num  
interface range vlan_range_num
```

Syntax

```
ip dhcp-relay max-hop-count <hop-count>  
no ip dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCP packet.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51  
%Info: Configuring only existing interfaces in range  
sonic-cli(conf-if-range-eth**)# ip dhcp-relay max-hop-count 9
```

ip dhcp-relay max-hop-count

Description

Configures the maximum hop count for the DHCP packet.

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]  
interface range po_range_num
```

Syntax

```
ip dhcp-relay max-hop-count <hop-count>
no ip dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCP packet.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay max-hop-count 9
```

ip dhcp-relay max-hop-count

Description

Configures the maximum hop count for the DHCP packet.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip dhcp-relay max-hop-count <hop-count>
no ip dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCP packet.

Examples

```
sonic-clia(config)# interface Ethernet 12
sonic-clia(conf-if-Ethernet0)# ip dhcp-relay max-hop-count 9
```

ip dhcp-relay policy-action

Description

Configure the policy for handling of DHCPv4 relay options.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip dhcp-relay policy-action <policyAction>
no ip dhcp-relay policy-action
```

Parameters

Name	Description	Type
policyAction	String(Max: 8 characters)	String

Usage Guidelines

Use this command to configure the policy for handling of DHCPv4 relay options.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay policy-action REPLACE
```

ip dhcp-relay policy-action

Description

Configure the policy for handling of DHCPv4 relay options.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip dhcp-relay policy-action <policyAction>
no ip dhcp-relay policy-action
```

Parameters

Name	Description	Type
policyAction	String(Max: 8 characters)	String

Usage Guidelines

Use this command to configure the policy for handling of DHCPv4 relay options.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay policy-action REPLACE
```

ip dhcp-relay policy-action

Description

Configure the policy for handling of DHCPv4 relay options.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip dhcp-relay policy-action <policyAction>
no ip dhcp-relay policy-action
```

Parameters

Name	Description	Type
policyAction	String(Max: 8 characters)	String

Usage Guidelines

Use this command to configure the policy for handling of DHCPv4 relay options.

Examples

```
sonic-clia(config)# interface Ethernet 12
sonic-clia(conf-if-Ethernet0)# ip dhcp-relay policy-action REPLACE
```

ip dhcp-relay policy-action

Description

Configure the policy for handling of DHCPv4 relay options.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
ip dhcp-relay policy-action <policyAction>
no ip dhcp-relay policy-action
```

Parameters

Name	Description	Type
policyAction	String(Max: 8 characters)	String

Usage Guidelines

Use this command to configure the policy for handling of DHCPv4 relay options.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay policy-action REPLACE
```

ip dhcp-relay policy-action

Description

Configure the policy for handling of DHCPv4 relay options.

Parent Commands (Modes)

```
interface range create vlan_range_num
interface range vlan_range_num
```

Syntax

```
ip dhcp-relay policy-action <policyAction>
no ip dhcp-relay policy-action
```

Parameters

Name	Description	Type
policyAction	String(Max: 8 characters)	String

Usage Guidelines

Use this command to configure the policy for handling of DHCPv4 relay options.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay policy-action REPLACE
```

ip dhcp-relay policy-action

Description

Configure the policy for handling of DHCPv4 relay options.

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [
    fallback ] [ fast_rate ]
interface range po_range_num
```

Syntax

```
ip dhcp-relay policy-action <policyAction>
no ip dhcp-relay policy-action
```

Parameters

Name	Description	Type
policyAction	String(Max: 8 characters)	String

Usage Guidelines

Use this command to configure the policy for handling of DHCPv4 relay options.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay policy-action REPLACE
```

ip dhcp-relay policy-action

Description

Configure the policy for handling of DHCPv4 relay options.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip dhcp-relay policy-action <policyAction>
no ip dhcp-relay policy-action
```

Parameters

Name	Description	Type
policyAction	String(Max: 8 characters)	String

Usage Guidelines

Use this command to configure the policy for handling of DHCPv4 relay options.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay policy-action REPLACE
```

ip dhcp-relay source-interface

Description

Configures the source IP address to be used for relaying the DHCP packets.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }  
no ip dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-cli(config)# interface Ethernet 12  
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay source-interface Ethernet36
```

ip dhcp-relay source-interface

Description

Configures the source IP address to be used for relaying the DHCP packets.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }  
no ip dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay source-interface Ethernet36
```

ip dhcp-relay source-interface

Description

Configures the source IP address to be used for relaying the DHCP packets.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }
no ip dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay source-interface Ethernet36
```

ip dhcp-relay source-interface

Description

Configures the source IP address to be used for relaying the DHCP packets.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
ip dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }
no ip dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay source-interface Ethernet36
```

ip dhcp-relay source-interface

Description

Configures the source IP address to be used for relaying the DHCP packets.

Parent Commands (Modes)

```
interface range create vlan_range_num  
interface range vlan_range_num
```

Syntax

```
ip dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }  
no ip dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51  
%Info: Configuring only existing interfaces in range  
sonic-cli(conf-if-range-eth**)# ip dhcp-relay source-interface Ethernet36
```

ip dhcp-relay source-interface

Description

Configures the source IP address to be used for relaying the DHCP packets.

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
interface range po_range_num
```

Syntax

```
ip dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }
no ip dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay source-interface Ethernet36
```

ip dhcp-relay source-interface

Description

Configures the source IP address to be used for relaying the DHCP packets.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }  
no ip dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-cli(config)# interface Ethernet 12  
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay source-interface Ethernet36
```

ip dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip dhcp-relay vrf-select  
no ip dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay vrf-select
```

ip dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip dhcp-relay vrf-select
no ip dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay vrf-select
```

ip dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [
    fallback ] [ fast_rate ]
```

Syntax

```
ip dhcp-relay vrf-select
no ip dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ip dhcp-relay vrf-select
```

ip dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
ip dhcp-relay vrf-select
no ip dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ip dhcp-relay vrf-select
```

ip dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface range create vlan_range_num
interface range vlan_range_num
```

Syntax

```
ip dhcp-relay vrf-select  
no ip dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51  
%Info: Configuring only existing interfaces in range  
sonic-cli(conf-if-range-eth**)# ip dhcp-relay vrf-select
```

ip dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]  
interface range po_range_num
```

Syntax

```
ip dhcp-relay vrf-select  
no ip dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51  
%Info: Configuring only existing interfaces in range  
sonic-cli(conf-if-range-eth**)# ip dhcp-relay vrf-select
```

ip dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip dhcp-relay vrf-select
no ip dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-clia(config)# interface Ethernet 12
sonic-clia(conf-if-Ethernet0)# ip dhcp-relay vrf-select
```

ip forward-protocol udp enable

Description

Enables IP helper relay functionality for specified UDP packets

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip forward-protocol udp enable
no ip forward-protocol udp enable
```

Usage Guidelines

```
[no] ip forward-protocol udp enable
```

Examples

```
sonic-cli# configure terminal  
sonic-cli(config)# ip forward-protocol udp enable  
sonic-cli(config)#
```

```
sonic-cli# configure terminal  
sonic-cli(config)# no ip forward-protocol udp enable  
sonic-cli(config)#
```

Features this CLI belongs to

- IP Helper #### Alternate command ##### click

```
config ip forward_protocol udp enable  
config ip forward_protocol udp disable
```

ip forward-protocol udp exclude

Description

Exclude UDP Port from list of IP packets that have to be relayed

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip forward-protocol udp exclude { <port> | <port> }
```

Parameters

Name	Description	Type
port	Well known UDP ports	Select [tftp(69) dns(53) ntp(37) netbios-name-server(137) netbios-datagram-server(138) tacacs(49)]

Usage Guidelines

```
ip forward-protocol udp exclude {  
    tftp|dns|ntp|netbios-name-server|netbios-datagram-server|tacacs|<custom-port> }
```

Examples

```
sonic-cli# configure terminal
sonic-cli(config)# ip forward-protocol udp exclude 12200
sonic-cli(config)#
```

```
sonic-cli# configure terminal
sonic-cli(config)# ip forward-protocol udp exclude tftp
sonic-cli(config)#
```

Features this CLI belongs to

- IP Helper

Alternate command

```
config ip forward_protocol udp remove
{[tftp/dns/ntp/netbios-name-server/netbios-datagram-server/tacacs] | <port> }
```

ip forward-protocol udp include

Description

Include UDP Port for which IP packets have to be relayed

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip forward-protocol udp include { <port> | <port> }
```

Parameters

Name	Description	Type
port	Well known UDP ports	Select [tftp(69) dns(53) ntp(37) netbios-name-server(137) netbios-datagram-server(138) tacacs(49)]

Usage Guidelines

```
ip forward-protocol udp include {  
    tftp|dns|ntp|netbios-name-server|netbios-datagram-server|tacacs|<custom-port> }
```

Examples

```
sonic-cli# configure terminal  
sonic-cli(config)# ip forward-protocol udp include 12200  
sonic-cli(config)#
```

```
sonic-cli# configure terminal  
sonic-cli(config)# ip forward-protocol udp include tftp  
sonic-cli(config)#
```

Features this CLI belongs to

- IP Helper

Alternate command

```
config ip forward_protocol udp add  
    {[tftp/dns/ntp/netbios-name-server/netbios-datagram-server/tacacs] | <port> }
```

ip forward-protocol udp rate-limit

Description

Rate limit CPU bound packets

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip forward-protocol udp rate-limit <rate>  
no ip forward-protocol udp rate-limit
```

Parameters

Name	Description	Type
rate	600 to 10,000 pps (default 600 pps)	Integer

Usage Guidelines

```
[no] ip forward-protocol udp rate-limit <rate>
```

Examples

```
sonic-cl# configure terminal  
sonic-cl(config)# ip forward-protocol udp rate-limit 1000  
sonic-cl(config)#
```

```
sonic-cl# configure terminal  
sonic-cl(config)# no ip forward-protocol udp rate-limit  
sonic-cl(config)#
```

Features this CLI belongs to

- IP Helper #### Alternate command ##### click

```
config ip forward_protocol udp rate_limit <value-in-pps>
```

ip host-access-list

Description

```
Create ip host-ACL
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip host-access-list <access-list-name>  
no ip host-access-list <access-list-name>
```

Parameters

Name	Description	Type
access-list-name	String(Max: 72 characters)	String

Usage Guidelines

ACL name can be of maximum 72 characters. The name must begin with A-Z, a-z or 0-9. Underscore and hypens can be used except as the first character. ACL name must be unique across all ACL types.

Examples

```
sonic(config)# ip host-access-list ipacl-example
```

ip load-share hash rotate

Description

Configure IP load-share hash rotate

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip load-share hash rotate <rotate-val>
no ip load-share hash rotate
```

Parameters

Name	Description	Type
rotate-val	IP ECMP Load share rotate value 0 to 35	Integer

Usage Guidelines

Use this command to configure IP load-share hash rotate

Examples

```
sonic-cli(config)# ip load-share hash rotate 10
```

ip name-server

Description

Configure the name server address

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip name-server <nameserver> { [ vrf { mgmt } ] }
no ip name-server <nameserver>
```

Parameters

Name	Description	Type
nameserver	A.B.C.D/A::B	String

ip name-server source-interface

Description

Configure source interface to pick the source IP, used for the DNS query

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip name-server source-interface { Ethernet | Loopback | Management | PortChannel | Vlan }  
no ip name-server source-interface
```

ip nht

Description

Configures default route based IPv4 nexthop resolution within default VRF.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip nht resolve-via-default  
no ip nht resolve-via-default
```

Usage Guidelines

Use this command to enable default route based IPv4 nexthop resolution within default VRF.

Examples

```
sonic-clis(config)# ip nht resolve-via-default
```

Features this CLI belongs to

- Nexthop Tracking ## ip ospf

Description

Configures OSPFv2 parameters within an IPv4 interface.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip ospf
```

Usage Guidelines

Use this command to configure OSPFv2 parameters under an IPv4 interface. IPv4 interface can be Ethernet interface, VLAN interface, Portchannel interface or a Loopback interface. Every OSPFv2 parameter on an interface can be associated with its specific IPv4 address by explicitly specifying the IPv4 address after the parameter. Specifying interface IPv4 address is an optional.

Examples

```
sonic-cli(config-router-ospf)# no ip ospf interface
```

Features this CLI belongs to

- OSPFv2

ip ospf

Description

Configures OSPFv2 parameters within an IPv4 interface.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf
```

Usage Guidelines

Use this command to configure OSPFv2 parameters under an IPv4 interface. IPv4 interface can be Ethernet interface, VLAN interface, Portchannel interface or a Loopback interface. Every OSPFv2 parameter on an interface can be associated with its specific IPv4 address by explicitly specifying the IPv4 address after the parameter. Specifying interface IPv4 address is an optional.

Examples

```
sonic-cli(config-router-ospf)# no ip ospf interface
```

Features this CLI belongs to

- OSPFv2

ip ospf

Description

Configures OSPFv2 parameters within an IPv4 interface.

Parent Commands (Modes)

interface <vlan-if-name>

Syntax

ip ospf

Usage Guidelines

Use this command to configure OSPFv2 parameters under an IPv4 interface. IPv4 interface can be Ethernet interface, VLAN interface, Portchannel interface or a Loopback interface. Every OSPFv2 parameter on an interface can be associated with its specific IPv4 address by explicitly specifying the IPv4 address after the parameter. Specifying interface IPv4 address is an optional.

Examples

```
sonic-clia(config-router-ospf)# no ip ospf interface
```

Features this CLI belongs to

- OSPFv2

ip ospf

Description

Configures OSPFv2 parameters within an IPv4 interface.

Parent Commands (Modes)

interface Loopback <lo-id>

Syntax

```
ip ospf
```

Usage Guidelines

Use this command to configure OSPFv2 parameters under an IPv4 interface. IPv4 interface can be Ethernet interface, VLAN interface, Portchannel interface or a Loopback interface. Every OSPFv2 parameter on an interface can be associated with its specific IPv4 address by explicitly specifying the IPv4 address after the parameter. Specifying interface IPv4 address is an optional.

Examples

```
sonic-cli(config-router-ospf)# no ip ospf interface
```

Features this CLI belongs to

- OSPFv2

ip ospf

Description

Configures OSPFv2 parameters within an IPv4 interface.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf
```

Usage Guidelines

Use this command to configure OSPFv2 parameters under an IPv4 interface. IPv4 interface can be Ethernet interface, VLAN interface, Portchannel interface or a Loopback interface. Every OSPFv2 parameter on an interface can be associated with its specific IPv4 address by explicitly specifying the IPv4 address after the parameter. Specifying interface IPv4 address is an optional.

Examples

```
sonic-cli(config-router-ospf)# no ip ospf interface
```

Features this CLI belongs to

- OSPFv2

ip ospf area

Description

```
Configures OSPFv2 interface area identifier.
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip ospf area <area-id> [ <ip-address> ]
no ip ospf area [ <ip-address> ]
```

Parameters

Name	Description	Type
area-id	A.B.C.D or 0..4294967295	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to associate an interface into an OSPFv2 area. Area identifier can be configured only when there is an already configured OSPFv2 router within the interface VRF and there are no network commands configured within that router. Area identifier configuration on an interface will get auto unconfigured while OSPFv2 router gets unconfigured from the VRF.

Examples

```
sonic-cli(config-router-ospf)# ip ospf area 19
sonic-cli(config-router-ospf)# ip ospf area 19.0.0.1
```

Features this CLI belongs to

- OSPFv2 ## ip ospf area ### Description

Configures OSPFv2 interface area identifier.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf area <area-id> [ <ip-address> ]
no ip ospf area [ <ip-address> ]
```

Parameters

Name	Description	Type
area-id	A.B.C.D or 0..4294967295	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to associate an interface into an OSPFv2 area. Area identifier can be configured only when there is an already configured OSPFv2 router within the interface VRF and there are no network commands configured within that router. Area identifier configuration on an interface will get auto unconfigured while OSPFv2 router gets unconfigured from the VRF.

Examples

```
sonic-cli(config-router-ospf)# ip ospf area 19
sonic-cli(config-router-ospf)# ip ospf area 19.0.0.1
```

Features this CLI belongs to

- OSPFv2 ## ip ospf area ### Description

Configures OSPFv2 interface area identifier.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip ospf area <area-id> [ <ip-address> ]
no ip ospf area [ <ip-address> ]
```

Parameters

Name	Description	Type
area-id	A.B.C.D or 0..4294967295	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to associate an interface into an OSPFv2 area. Area identifier can be configured only when there is an already configured OSPFv2 router within the interface VRF and there are no network commands configured within that router. Area identifier configuration on an interface will get auto unconfigured while OSPFv2 router gets unconfigured from the VRF.

Examples

```
sonic-cli(config-router-ospf)# ip ospf area 19
sonic-cli(config-router-ospf)# ip ospf area 19.0.0.1
```

Features this CLI belongs to

- OSPFv2 ## ip ospf area ### Description

Configures OSPFv2 interface area identifier.

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ip ospf area <area-id> [ <ip-address> ]
no ip ospf area [ <ip-address> ]
```

Parameters

Name	Description	Type
area-id	A.B.C.D or 0..4294967295	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to associate an interface into an OSPFv2 area. Area identifier can be configured only when there is an already configured OSPFv2 router within the interface VRF and there are no network commands configured within that router. Area identifier configuration on an interface will get auto unconfigured while OSPFv2 router gets unconfigured from the VRF.

Examples

```
sonic-cli(config-router-ospf)# ip ospf area 19
sonic-cli(config-router-ospf)# ip ospf area 19.0.0.1
```

Features this CLI belongs to

- OSPFv2 ## ip ospf area ### Description

Configures OSPFv2 interface area identifier.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf area <area-id> [ <ip-address> ]
no ip ospf area [ <ip-address> ]
```

Parameters

Name	Description	Type
area-id	A.B.C.D or 0..4294967295	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to associate an interface into an OSPFv2 area. Area identifier can be configured only when there is an already configured OSPFv2 router within the interface VRF and there are no network commands configured within that router. Area identifier configuration on an interface will get auto unconfigured while OSPFv2 router gets unconfigured from the VRF.

Examples

```
sonic-cli(config-router-ospf)# ip ospf area 19
sonic-cli(config-router-ospf)# ip ospf area 19.0.0.1
```

Features this CLI belongs to

- OSPFv2 ## ip ospf authentication ## Description

Configures OSPFv2 interface authentication type.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip ospf authentication { { [ message-digest [ <ip-address> ] ] } | { [ null [ <ip-address> ] ] } } [ <ip-address> ]
no ip ospf authentication { { [ message-digest [ <ip-address> ] ] } | { [ null [ <ip-address> ] ] } } [ <ip-address> ]
```

Parameters

Name	Description	Type
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to enable OSPFv2 authentication type for OSPFv2 messages. Authentocation types can be clear text authentication, message-digest authentication or no authentication. Interface mode authentication type would override router mode area authentication type. Based on the configured authentication type, corresponding configured authentication key would be used for OPSFV2 message authentication.

Examples

```
sonic-cl(i(config-router-ospf)# ip ospf authentication  
sonic-cl(i(config-router-ospf)# ip ospf authentication message-digest  
sonic-cl(i(config-router-ospf)# ip ospf authentication null
```

Features this CLI belongs to

- OSPFv2 ## ip ospf authentication ### Description

Configures OSPFv2 interface authentication type.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]
```

Syntax

```
ip ospf authentication { { [ message-digest [ <ip-address> ] ] } | { [ null [ <ip-address> ] ]  
    } } ] [ <ip-address> ]  
no ip ospf authentication { { [ message-digest [ <ip-address> ] ] } | { [ null [ <ip-address> ]  
    ] } } ] [ <ip-address> ]
```

Parameters

Name	Description	Type
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to enable OSPFv2 authentication type for OSPFv2 messages. Authentication types can be clear text authentication, message-digest authentication or no authentication. Interface mode authentication type would override router mode area authentication type. Based on the configured authentication type, corresponding configured authentication key would be used for OSPFV2 message authentication.

Examples

```
sonic-cl(i(config-router-ospf)# ip ospf authentication  
sonic-cl(i(config-router-ospf)# ip ospf authentication message-digest  
sonic-cl(i(config-router-ospf)# ip ospf authentication null
```

Features this CLI belongs to

- OSPFv2 ## ip ospf authentication ### Description

Configures OSPFv2 interface authentication type.

Parent Commands (Modes)

interface <vlan-if-name>

Syntax

```
ip ospf authentication { { [ message-digest [ <ip-address> ] ] } | { [ null [ <ip-address> ] ] } } [ <ip-address> ]
no ip ospf authentication { { [ message-digest [ <ip-address> ] ] } | { [ null [ <ip-address> ] ] } } [ <ip-address> ]
```

Parameters

Name	Description	Type
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to enable OSPFv2 authentication type for OSPFv2 messages. Authentication types can be clear text authentication, message-digest authentication or no authentication. Interface mode authentication type would override router mode area authentication type. Based on the configured authentication type, corresponding configured authentication key would be used for OPSFV2 message authentication.

Examples

```
sonic-cli(config-router-ospf)# ip ospf authentication
sonic-cli(config-router-ospf)# ip ospf authentication message-digest
sonic-cli(config-router-ospf)# ip ospf authentication null
```

Features this CLI belongs to

- OSPFv2 ## ip ospf authentication ### Description

Configures OSPFv2 interface authentication type.

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ip ospf authentication { { [ message-digest [ <ip-address> ] ] } | { [ null [ <ip-address> ] ] } } [ <ip-address> ]
no ip ospf authentication { { [ message-digest [ <ip-address> ] ] } | { [ null [ <ip-address> ] ] } } [ <ip-address> ]
```

Parameters

Name	Description	Type
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to enable OSPFv2 authentication type for OSPFv2 messages. Authentocation types can be clear text authentication, message-digest authentication or no authentication. Interface mode authentication type would override router mode area authentication type. Based on the configured authentication type, corresponding configured authentication key would be used for OPSFV2 message authentication.

Examples

```
sonic-cli(config-router-ospf)# ip ospf authentication
sonic-cli(config-router-ospf)# ip ospf authentication message-digest
sonic-cli(config-router-ospf)# ip ospf authentication null
```

Features this CLI belongs to

- OSPFv2 ## ip ospf authentication ### Description

Configures OSPFv2 interface authentication type.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf authentication { { [ message-digest [ <ip-address> ] ] } | { [ null [ <ip-address> ] ] } } [ <ip-address> ]
no ip ospf authentication { { [ message-digest [ <ip-address> ] ] } | { [ null [ <ip-address> ] ] } } [ <ip-address> ]
```

Parameters

Name	Description	Type
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to enable OSPFv2 authentication type for OSPFv2 messages. Authentocation types can be clear text authentication, message-digest authentication or no authentication. Interface mode authentication type would override router mode area authentication type. Based on the configured authentication type, corresponding configured authentication key would be used for OPSFV2 message authentication.

Examples

```
sonic-cl(i(config-router-ospf)# ip ospf authentication
sonic-cl(i(config-router-ospf)# ip ospf authentication message-digest
sonic-cl(i(config-router-ospf)# ip ospf authentication null
```

Features this CLI belongs to

- OSPFv2 ## ip ospf authentication-key ### Description

Configures OSPFv2 interface clear text authentication key.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip ospf authentication-key <authkey> { { [ encrypted [ <ip-address> ] ] } [ <ip-address> ] }
no ip ospf authentication-key [ <ip-address> ]
```

Parameters

Name	Description	Type
authkey	String	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 clear text authentication key. Clear text authentication key can be up to eight characters long. User provided actual password will be displayed as encrypted string in show configuration command displays and saved as encrypted string in configuration file. It is recommended to use only actual password and not to use encrypted string as password.

Examples

```
sonic-cli(config-router-ospf)# ip ospf authentication-key pa$$woRd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf authentication-key ### Description

Configures OSPFv2 interface clear text authentication key.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf authentication-key <authkey> { { [ encrypted [ <ip-address> ] ] } [ <ip-address> ] }  
no ip ospf authentication-key [ <ip-address> ]
```

Parameters

Name	Description	Type
authkey	String	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 clear text authentication key. Clear text authentication key can be up to eight characters long. User provided actual password will be displayed as encrypted string in show configuration comamnd displays and saved as encrypted string in configuration file. It is recomended to use only actual password and not to use encrypted string as password.

Examples

```
sonic-cli(config-router-ospf)# ip ospf authentication-key pa$$woRd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf authentication-key ### Description

Configures OSPFv2 interface clear text authentication key.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip ospf authentication-key <authkey> { { [ encrypted [ <ip-address> ] ] } [ <ip-address> ] }  
no ip ospf authentication-key [ <ip-address> ]
```

Parameters

Name	Description	Type
authkey	String	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 clear text authentication key. Clear text authentication key can be up to eight characters long. User provided actual password will be displayed as encrypted string in show configuration comamnd displays and saved as encrypted string in configuration file. It is recomended to use only actual password and not to use encrypted string as password.

Examples

```
sonic-cli(config-router-ospf)# ip ospf authentication-key pa$$woRd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf authentication-key ### Description

Configures OSPFv2 interface clear text authentication key.

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ip ospf authentication-key <authkey> { { [ encrypted [ <ip-address> ] ] } [ <ip-address> ] }  
no ip ospf authentication-key [ <ip-address> ]
```

Parameters

Name	Description	Type
authkey	String	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 clear text authentication key. Clear text authentication key can be up to eight characters long. User provided actual password will be displayed as encrypted string in show configuration command displays and saved as encrypted string in configuration file. It is recommended to use only actual password and not to use encrypted string as password.

Examples

```
sonic-cli(config-router-ospf)# ip ospf authentication-key pa$$woRd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf authentication-key ### Description

Configures OSPFv2 interface clear text authentication key.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf authentication-key <authkey> { { [ encrypted [ <ip-address> ] ] } [ <ip-address> ] }
no ip ospf authentication-key [ <ip-address> ]
```

Parameters

Name	Description	Type
authkey	String	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 clear text authentication key. Clear text authentication key can be up to eight characters long. User provided actual password will be displayed as encrypted string in show configuration command displays and saved as encrypted string in configuration file. It is recommended to use only actual password and not to use encrypted string as password.

Examples

```
sonic-cli(config-router-ospf)# ip ospf authentication-key pa$$woRd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf bfd ##### Description

Configures OSPFv2 interface BFD.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip ospf bfd
no ip ospf bfd
```

Usage Guidelines

Use this command to configure OSPFv2 interface BFD. Enabling BFD will establish a BFD session between the OSPF neighbors. Any failure in BFD session will bring down the OSPFv2 session.

Examples

```
sonic-cli(config-router-ospf)# ip ospf bfd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf bfd ##### Description

Configures OSPFv2 interface BFD.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf bfd  
no ip ospf bfd
```

Usage Guidelines

Use this command to configure OSPFv2 interface BFD. Enabling BFD will establish a BFD session between the OSPF neighbors. Any failure in BFD session will bring down the OSPFv2 session.

Examples

```
sonic-cli(config-router-ospf)# ip ospf bfd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf bfd ##### Description

Configures OSPFv2 interface BFD.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip ospf bfd  
no ip ospf bfd
```

Usage Guidelines

Use this command to configure OSPFv2 interface BFD. Enabling BFD will establish a BFD session between the OSPF neighbors. Any failure in BFD session will bring down the OSPFv2 session.

Examples

```
sonic-cli(config-router-ospf)# ip ospf bfd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf bfd ##### Description

Configures OSPFv2 interface BFD.

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ip ospf bfd  
no ip ospf bfd
```

Usage Guidelines

Use this command to configure OSPFv2 interface BFD. Enabling BFD will establish a BFD session between the OSPF neighbors. Any failure in BFD session will bring down the OSPFv2 session.

Examples

```
sonic-cli(config-router-ospf)# ip ospf bfd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf bfd ##### Description

Configures OSPFv2 interface BFD.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf bfd
no ip ospf bfd
```

Usage Guidelines

Use this command to configure OSPFv2 interface BFD. Enabling BFD will establish a BFD session between the OSPF neighbors. Any failure in BFD session will bring down the OSPFv2 session.

Examples

```
sonic-cli(config-router-ospf)# ip ospf bfd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf bfd profile ### Description

Configures OSPFv2 interface BFD profile.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip ospf bfd profile <profilename>
no ip ospf bfd profile
```

Parameters

Name	Description	Type
profilename	WORD	String

Usage Guidelines

Use this command to configure OSPFv2 interface BFD profile. Enabling profile will also enable BFD on interface if not enabled.

Examples

```
sonic-cli(config-router-ospf)# ip ospf bfd profile ospf
```

Features this CLI belongs to

- OSPFv2 ## ip ospf bfd profile ### Description

Configures OSPFv2 interface BFD profile.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf bfd profile <profilename>
no ip ospf bfd profile
```

Parameters

Name	Description	Type
profilename	WORD	String

Usage Guidelines

Use this command to configure OSPFv2 interface BFD profile. Enabling profile will also enable BFD on interface if not enabled.

Examples

```
sonic-cli(config-router-ospf)# ip ospf bfd profile ospf
```

Features this CLI belongs to

- OSPFv2 ## ip ospf bfd profile ### Description

Configures OSPFv2 interface BFD profile.

Parent Commands (Modes)

`interface <vlan-if-name>`

Syntax

```
ip ospf bfd profile <filename>
no ip ospf bfd profile
```

Parameters

Name	Description	Type
filename	WORD	String

Usage Guidelines

Use this command to configure OSPFv2 interface BFD profile. Enabling profile will also enable BFD on interface if not enabled.

Examples

```
sonic-clis(config-router-ospf)# ip ospf bfd profile ospf
```

Features this CLI belongs to

- OSPFv2 ## ip ospf bfd profile ### Description

Configures OSPFv2 interface BFD profile.

Parent Commands (Modes)

`interface Loopback <lo-id>`

Syntax

```
ip ospf bfd profile <profilenam>
no ip ospf bfd profile
```

Parameters

Name	Description	Type
profilenam	WORD	String

Usage Guidelines

Use this command to configure OSPFv2 interface BFD profile. Enabling profile will also enable BFD on interface if not enabled.

Examples

```
sonic-cl(i(config-router-ospf)# ip ospf bfd profile ospf
```

Features this CLI belongs to

- OSPFv2 ## ip ospf bfd profile ### Description

Configures OSPFv2 interface BFD profile.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf bfd profile <profilenam>
no ip ospf bfd profile
```

Parameters

Name	Description	Type
profilenam	WORD	String

Usage Guidelines

Use this command to configure OSPFv2 interface BFD profile. Enabling profile will also enable BFD on interface if not enabled.

Examples

```
sonic-cli(config-router-ospf)# ip ospf bfd profile ospf
```

Features this CLI belongs to

- OSPFv2 ## ip ospf cost #### Description

Configures OSPFv2 interface cost.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip ospf cost <interface-cost> [ <ip-address> ]  
no ip ospf cost [ <ip-address> ]
```

Parameters

Name	Description	Type
interface-cost		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface cost.

Examples

```
sonic-cli(config-router-ospf)# ip ospf cost 38
```

Features this CLI belongs to

- OSPFv2 ## ip ospf cost ### Description

Configures OSPFv2 interface cost.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf cost <interface-cost> [ <ip-address> ]
no ip ospf cost [ <ip-address> ]
```

Parameters

Name	Description	Type
interface-cost		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface cost.

Examples

```
sonic-cli(config-router-ospf)# ip ospf cost 38
```

Features this CLI belongs to

- OSPFv2 ## ip ospf cost ### Description

Configures OSPFv2 interface cost.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip ospf cost <interface-cost> [ <ip-address> ]  
no ip ospf cost [ <ip-address> ]
```

Parameters

Name	Description	Type
interface-cost		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface cost.

Examples

```
sonic-clia(config-router-ospf)# ip ospf cost 38
```

Features this CLI belongs to

- OSPFv2 ## ip ospf cost ### Description

Configures OSPFv2 interface cost.

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ip ospf cost <interface-cost> [ <ip-address> ]  
no ip ospf cost [ <ip-address> ]
```

Parameters

Name	Description	Type
interface-cost		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface cost.

Examples

```
sonic-cli(config-router-ospf)# ip ospf cost 38
```

Features this CLI belongs to

- OSPFv2 ## ip ospf cost ##### Description

Configures OSPFv2 interface cost.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf cost <interface-cost> [ <ip-address> ]
no ip ospf cost [ <ip-address> ]
```

Parameters

Name	Description	Type
interface-cost		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface cost.

Examples

```
sonic-cli(config-router-ospf)# ip ospf cost 38
```

Features this CLI belongs to

- OSPFv2 ## ip ospf dead-interval ##### Description

Configure OSPFv2 adjacency dead interval

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
ip ospf dead-interval { { [ <deadinterval> [ <ip-address> ] ] } | { [ minimal { [ hello-multiplier { <hellomultiplier> [ <ip-address> ] } ] } ] } }
no ip ospf dead-interval { [ <ip-address> ] | { [ minimal { [ hello-multiplier [ <ip-address> ] ] } ] } }
```

Parameters

Name	Description	Type
deadinterval		Integer
ip-address	A.B.C.D	String
hellomultiplier		Integer

ip ospf dead-interval

Description

Configure OSPFv2 adjacency dead interval

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf dead-interval { { [ <deadinterval> [ <ip-address> ] ] } | { [ minimal { [ hello-multiplier { <hellomultiplier> [ <ip-address> ] } ] } ] } }
no ip ospf dead-interval { [ <ip-address> ] | { [ minimal { [ hello-multiplier [ <ip-address> ] ] } ] } }
```

Parameters

Name	Description	Type
deadinterval		Integer
ip-address	A.B.C.D	String
hellomultiplier		Integer

ip ospf dead-interval

Description

```
Configure OSPFv2 adjacency dead interval
```

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip ospf dead-interval { { [ <deadinterval> [ <ip-address> ] ] } | { [ minimal { [ hello-multiplier { <hellomultiplier> [ <ip-address> ] } ] } ] } }  
no ip ospf dead-interval { [ <ip-address> ] | { [ minimal { [ hello-multiplier [ <ip-address> ] ] } ] } }
```

Parameters

Name	Description	Type
deadinterval		Integer
ip-address	A.B.C.D	String
hellomultiplier		Integer

ip ospf dead-interval

Description

```
Configure OSPFv2 adjacency dead interval
```

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ip ospf dead-interval { { [ <deadinterval> [ <ip-address> ] ] } | { [ minimal { [ hello-multiplier { <hellomultiplier> [ <ip-address> ] } ] } ] } }  
no ip ospf dead-interval { [ <ip-address> ] | { [ minimal { [ hello-multiplier [ <ip-address> ] ] } ] } }
```

Parameters

Name	Description	Type
deadinterval		Integer
ip-address	A.B.C.D	String
hellomultiplier		Integer

ip ospf dead-interval

Description

```
Configure OSPFv2 adjacency dead interval
```

Parent Commands (Modes)

```
interface <phy-sub-if-name>  
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf dead-interval { { [ <deadinterval> [ <ip-address> ] ] } | { [ minimal { [ hello-multiplier { <hellomultiplier> [ <ip-address> ] } ] } ] } }  
no ip ospf dead-interval { [ <ip-address> ] | { [ minimal { [ hello-multiplier [ <ip-address> ] ] } ] } }
```

Parameters

Name	Description	Type
deadinterval		Integer
ip-address	A.B.C.D	String

Name	Description	Type
hellomultiplier		Integer

ip ospf hello-interval

Description

Configures OSPFv2 interface neighbour hello interval.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
ip ospf hello-interval [ <hellointerval> [ <ip-address> ] ]
no ip ospf hello-interval [ <ip-address> ]
```

Parameters

Name	Description	Type
hellointerval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface neighbour hello interval.

Examples

```
sonic-clia(config-router-ospf)# ip ospf thello-interval 20
```

Features this CLI belongs to

- OSPFv2 ## ip ospf hello-interval ##### Description

Configures OSPFv2 interface neighbour hello interval.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf hello-interval [ <hellointerval> [ <ip-address> ] ]  
no ip ospf hello-interval [ <ip-address> ]
```

Parameters

Name	Description	Type
hellointerval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface neighbour hello interval.

Examples

```
sonic-cli(config-router-ospf)# ip ospf thello-interval 20
```

Features this CLI belongs to

- OSPFv2 ## ip ospf hello-interval ### Description

Configures OSPFv2 interface neighbour hello interval.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip ospf hello-interval [ <hellointerval> [ <ip-address> ] ]  
no ip ospf hello-interval [ <ip-address> ]
```

Parameters

Name	Description	Type
hellointerval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface neighbour hello interval.

Examples

```
sonic-clia(config-router-ospf)# ip ospf thello-interval 20
```

Features this CLI belongs to

- OSPFv2 ## ip ospf hello-interval ### Description

Configures OSPFv2 interface neighbour hello interval.

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ip ospf hello-interval [ <hellointerval> [ <ip-address> ] ]  
no ip ospf hello-interval [ <ip-address> ]
```

Parameters

Name	Description	Type
hellointerval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface neighbour hello interval.

Examples

```
sonic-cli(config-router-ospf)# ip ospf thello-interval 20
```

Features this CLI belongs to

- OSPFv2 ## ip ospf hello-interval ### Description

Configures OSPFv2 interface neighbour hello interval.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf hello-interval [ <hellointerval> [ <ip-address> ] ]
no ip ospf hello-interval [ <ip-address> ]
```

Parameters

Name	Description	Type
hellointerval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface neighbour hello interval.

Examples

```
sonic-cli(config-router-ospf)# ip ospf thello-interval 20
```

Features this CLI belongs to

- OSPFv2 ## ip ospf message-digest-key ### Description

Configures OSPFv2 interface message digest authentication key.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip ospf message-digest-key <keyid> { md5 { <md5key> { [ encrypted [ <ip-address> ] ] } [ <ip-address> ] } }  
no ip ospf message-digest-key <keyid> { md5 [ <ip-address> ] }
```

Parameters

Name	Description	Type
keyid		Integer
md5key	String	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 message digest authentication key. Message digest authentication key can be up to sixteen characters long. User provided actual password will be displayed as encrypted string in show configuration command displays and saved as encrypted string in configuration file. It is recommended to use only actual password and not to use encrypted string as password.

Examples

```
sonic-cli(config-router-ospf)# ip ospf message-digest-key 10 md5 mDpa$$woRd  
sonic-cli(config-router-ospf)# ip ospf message-digest-key 19 md5 mDpa$$woRd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf message-digest-key ### Description

Configures OSPFv2 interface message digest authentication key.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf message-digest-key <keyid> { md5 { <md5key> { [ encrypted [ <ip-address> ] ] } [ <ip-address> ] } }
no ip ospf message-digest-key <keyid> { md5 [ <ip-address> ] }
```

Parameters

Name	Description	Type
keyid		Integer
md5key	String	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 message digest authentication key. Message digest authentication key can be up to sixteen characters long. User provided actual password will be displayed as encrypted string in show configuration command displays and saved as encrypted string in configuration file. It is recommended to use only actual password and not to use encrypted string as password.

Examples

```
sonic-cli(config-router-ospf)# ip ospf message-digest-key 10 md5 mDpa$$woRd
sonic-cli(config-router-ospf)# ip ospf message-digest-key 19 md5 mDpa$$woRd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf message-digest-key ### Description

Configures OSPFv2 interface message digest authentication key.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip ospf message-digest-key <keyid> { md5 { <md5key> { [ encrypted [ <ip-address> ] ] } [ <ip-address> ] } }
no ip ospf message-digest-key <keyid> { md5 [ <ip-address> ] }
```

Parameters

Name	Description	Type
keyid		Integer
md5key	String	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 message digest authentication key. Message digest authentication key can be up to sixteen characters long. User provided actual password will be displayed as encrypted string in show configuration command displays and saved as encrypted string in configuration file. It is recommended to use only actual password and not to use encrypted string as password.

Examples

```
sonic-clt(config-router-ospf)# ip ospf message-digest-key 10 md5 mDpa$$woRd  
sonic-clt(config-router-ospf)# ip ospf message-digest-key 19 md5 mDpa$$woRd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf message-digest-key ### Description

Configures OSPFv2 interface message digest authentication key.

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ip ospf message-digest-key <keyid> { md5 { <md5key> { [ encrypted [ <ip-address> ] ] } [ <ip-address> ] } }  
no ip ospf message-digest-key <keyid> { md5 [ <ip-address> ] }
```

Parameters

Name	Description	Type
keyid		Integer
md5key	String	String

Name	Description	Type
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 message digest authentication key. Message digest authentication key can be up to sixteen characters long. User provided actual password will be displayed as encrypted string in show configuration command displays and saved as encrypted string in configuration file. It is recommended to use only actual password and not to use encrypted string as password.

Examples

```
sonic-cl(i(config-router-ospf)# ip ospf message-digest-key 10 md5 mDpa$$woRd
sonic-cl(i(config-router-ospf)# ip ospf message-digest-key 19 md5 mDpa$$woRd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf message-digest-key ### Description

Configures OSPFv2 interface message digest authentication key.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf message-digest-key <keyid> { md5 { <md5key> { [ encrypted [ <ip-address> ] ] } [ <ip-address> ] } }
no ip ospf message-digest-key <keyid> { md5 [ <ip-address> ] }
```

Parameters

Name	Description	Type
keyid		Integer
md5key	String	String
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 message digest authentication key. Message digest authentication key can be up to sixteen characters long. User provided actual password will be displayed as encrypted string in show configuration command displays and saved as encrypted string in configuration file. It is recommended to use only actual password and not to use encrypted string as password.

Examples

```
sonic-cli(config-router-ospf)# ip ospf message-digest-key 10 md5 mDpa$$woRd  
sonic-cli(config-router-ospf)# ip ospf message-digest-key 19 md5 mDpa$$woRd
```

Features this CLI belongs to

- OSPFv2 ## ip ospf mtu-ignore ##### Description

Disables OSPFv2 MTU mismatch detection.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip ospf mtu-ignore [ <ip-address> ]  
no ip ospf mtu-ignore [ <ip-address> ]
```

Parameters

Name	Description	Type
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to disable OSPFv2 MTU mismatch detection. MTU mismatch detection is enabled by default.

Examples

```
sonic-cli(config-router-ospf)# ip ospf mtu-ignore
```

Features this CLI belongs to

- OSPFv2 ## ip ospf mtu-ignore ### Description

Disables OSPFv2 MTU mismatch detection.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf mtu-ignore [ <ip-address> ]
no ip ospf mtu-ignore [ <ip-address> ]
```

Parameters

Name	Description	Type
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to disable OSPFv2 MTU mismatch detection. MTU mismatch detection is enabled by default.

Examples

```
sonic-cli(config-router-ospf)# ip ospf mtu-ignore
```

Features this CLI belongs to

- OSPFv2 ## ip ospf mtu-ignore ### Description

Disables OSPFv2 MTU mismatch detection.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip ospf mtu-ignore [ <ip-address> ]
no ip ospf mtu-ignore [ <ip-address> ]
```

Parameters

Name	Description	Type
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to disable OSPFv2 MTU mismatch detection. MTU mismatch detection is enabled by default.

Examples

```
sonic-cli(config-router-ospf)# ip ospf mtu-ignore
```

Features this CLI belongs to

- OSPFv2 ## ip ospf mtu-ignore ##### Description

Disables OSPFv2 MTU mismatch detection.

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ip ospf mtu-ignore [ <ip-address> ]
no ip ospf mtu-ignore [ <ip-address> ]
```

Parameters

Name	Description	Type
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to disable OSPFv2 MTU mismatch detection. MTU mismatch detection is enabled by default.

Examples

```
sonic-cli(config-router-ospf)# ip ospf mtu-ignore
```

Features this CLI belongs to

- OSPFv2 ## ip ospf mtu-ignore ##### Description

Disables OSPFv2 MTU mismatch detection.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf mtu-ignore [ <ip-address> ]
no ip ospf mtu-ignore [ <ip-address> ]
```

Parameters

Name	Description	Type
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to disable OSPFv2 MTU mismatch detection. MTU mismatch detection is enabled by default.

Examples

```
sonic-cli(config-router-ospf)# ip ospf mtu-ignore
```

Features this CLI belongs to

- OSPFv2 ## ip ospf network ### Description

Configures OSPFv2 interface network type.

Parent Commands (Modes)

`interface <phy-if-name>`

Syntax

```
ip ospf network { broadcast | point-to-point }
no ip ospf network
```

Usage Guidelines

Use this command to configure OSPFv2 interface network type. Broadcast and Point-to-point networks types are supported. By default Network type will be broadcast.

Examples

```
sonic-cli(config-router-ospf)# ip ospf network point-to-point
sonic-cli(config-router-ospf)# ip ospf network broadcast
```

Features this CLI belongs to

- OSPFv2 ## ip ospf network ### Description

Configures OSPFv2 interface network type.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf network { broadcast | point-to-point }
no ip ospf network
```

Usage Guidelines

Use this command to configure OSPFv2 interface network type. Broadcast and Point-to-point networks types are supported. By default Network type will be broadcast.

Examples

```
sonic-cli(config-router-ospf)# ip ospf network point-to-point  
sonic-cli(config-router-ospf)# ip ospf network broadcast
```

Features this CLI belongs to

- OSPFv2 ## ip ospf network ### Description

Configures OSPFv2 interface network type.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip ospf network { broadcast | point-to-point }  
no ip ospf network
```

Usage Guidelines

Use this command to configure OSPFv2 interface network type. Broadcast and Point-to-point networks types are supported. By default Network type will be broadcast.

Examples

```
sonic-cli(config-router-ospf)# ip ospf network point-to-point  
sonic-cli(config-router-ospf)# ip ospf network broadcast
```

Features this CLI belongs to

- OSPFv2 ## ip ospf network ### Description

Configures OSPFv2 interface network type.

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ip ospf network { broadcast | point-to-point }  
no ip ospf network
```

Usage Guidelines

Use this command to configure OSPFv2 interface network type. Broadcast and Point-to-point networks types are supported. By default Network type will be broadcast.

Examples

```
sonic-cli(config-router-ospf)# ip ospf network point-to-point  
sonic-cli(config-router-ospf)# ip ospf network broadcast
```

Features this CLI belongs to

- OSPFv2 ## ip ospf network ### Description

Configures OSPFv2 interface network type.

Parent Commands (Modes)

```
interface <phy-sub-if-name>  
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf network { broadcast | point-to-point }  
no ip ospf network
```

Usage Guidelines

Use this command to configure OSPFv2 interface network type. Broadcast and Point-to-point networks types are supported. By default Network type will be broadcast.

Examples

```
sonic-cli(config-router-ospf)# ip ospf network point-to-point  
sonic-cli(config-router-ospf)# ip ospf network broadcast
```

Features this CLI belongs to

- OSPFv2 ## ip ospf priority ### Description

Configures OSPFv2 adjacency router priority.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip ospf priority <priorityval> [ <ip-address> ]
no ip ospf priority [ <ip-address> ]
```

Parameters

Name	Description	Type
priorityval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 adjacency router priority.

Examples

```
sonic-cli(config-router-ospf)# ip ospf priority 19
```

Features this CLI belongs to

- OSPFv2 ## ip ospf priority ### Description

Configures OSPFv2 adjacency router priority.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf priority <priorityval> [ <ip-address> ]
no ip ospf priority [ <ip-address> ]
```

Parameters

Name	Description	Type
priorityval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 adjacency router priority.

Examples

```
sonic-clia(config-router-ospf)# ip ospf priority 19
```

Features this CLI belongs to

- OSPFv2 ## ip ospf priority ### Description

Configures OSPFv2 adjacency router priority.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip ospf priority <priorityval> [ <ip-address> ]  
no ip ospf priority [ <ip-address> ]
```

Parameters

Name	Description	Type
priorityval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 adjacency router priority.

Examples

```
sonic-cli(config-router-ospf)# ip ospf priority 19
```

Features this CLI belongs to

- OSPFv2 ## ip ospf priority ### Description

```
Configures OSPFv2 adjacency router priority.
```

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ip ospf priority <priorityval> [ <ip-address> ]  
no ip ospf priority [ <ip-address> ]
```

Parameters

Name	Description	Type
priorityval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

```
Use this command to configure OSPFv2 adjacency router priority.
```

Examples

```
sonic-cli(config-router-ospf)# ip ospf priority 19
```

Features this CLI belongs to

- OSPFv2 ## ip ospf priority ### Description

```
Configures OSPFv2 adjacency router priority.
```

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf priority <priorityval> [ <ip-address> ]
no ip ospf priority [ <ip-address> ]
```

Parameters

Name	Description	Type
priorityval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 adjacency router priority.

Examples

```
sonic-cli(config-router-ospf)# ip ospf priority 19
```

Features this CLI belongs to

- OSPFv2 ## ip ospf retransmit-interval ### Description

Configures OSPFv2 interface LSA retransmit interval.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip ospf retransmit-interval <retransmitinterval> [ <ip-address> ]
no ip ospf retransmit-interval [ <ip-address> ]
```

Parameters

Name	Description	Type
retransmitinterval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface LSA retransmit interval.

Examples

```
sonic-clia(config-router-ospf)# ip ospf retransmit-interval 35
```

Features this CLI belongs to

- OSPFv2 ## ip ospf retransmit-interval ### Description

Configures OSPFv2 interface LSA retransmit interval.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf retransmit-interval <retransmitinterval> [ <ip-address> ]
no ip ospf retransmit-interval [ <ip-address> ]
```

Parameters

Name	Description	Type
retransmitinterval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface LSA retransmit interval.

Examples

```
sonic-cli(config-router-ospf)# ip ospf retransmit-interval 35
```

Features this CLI belongs to

- OSPFv2 ## ip ospf retransmit-interval ### Description

Configures OSPFv2 interface LSA retransmit interval.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip ospf retransmit-interval <retransmitinterval> [ <ip-address> ]  
no ip ospf retransmit-interval [ <ip-address> ]
```

Parameters

Name	Description	Type
retransmitinterval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface LSA retransmit interval.

Examples

```
sonic-cli(config-router-ospf)# ip ospf retransmit-interval 35
```

Features this CLI belongs to

- OSPFv2 ## ip ospf retransmit-interval ### Description

Configures OSPFv2 interface LSA retransmit interval.

Parent Commands (Modes)

interface Loopback <lo-id>

Syntax

```
ip ospf retransmit-interval <retransmitinterval> [ <ip-address> ]  
no ip ospf retransmit-interval [ <ip-address> ]
```

Parameters

Name	Description	Type
retransmitinterval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface LSA retransmit interval.

Examples

```
sonic-cli(config-router-ospf)# ip ospf retransmit-interval 35
```

Features this CLI belongs to

- OSPFv2 ## ip ospf retransmit-interval ### Description

Configures OSPFv2 interface LSA retransmit interval.

Parent Commands (Modes)

interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>

Syntax

```
ip ospf retransmit-interval <retransmitinterval> [ <ip-address> ]
no ip ospf retransmit-interval [ <ip-address> ]
```

Parameters

Name	Description	Type
retransmitinterval		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface LSA retransmit interval.

Examples

```
sonic-clia(config-router-ospf)# ip ospf retransmit-interval 35
```

Features this CLI belongs to

- OSPFv2 ## ip ospf transmit-delay ### Description

Configures OSPFv2 interface LSA transmit delay interval.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip ospf transmit-delay <transmitdelay> [ <ip-address> ]
no ip ospf transmit-delay [ <ip-address> ]
```

Parameters

Name	Description	Type
transmitdelay		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface LSA transmit delay interval.

Examples

```
sonic-cli(config-router-ospf)# ip ospf transmit-delay 35
```

Features this CLI belongs to

- OSPFv2 ## ip ospf transmit-delay ### Description

Configures OSPFv2 interface LSA transmit delay interval.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip ospf transmit-delay <transmitdelay> [ <ip-address> ]
no ip ospf transmit-delay [ <ip-address> ]
```

Parameters

Name	Description	Type
transmitdelay		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface LSA transmit delay interval.

Examples

```
sonic-cli(config-router-ospf)# ip ospf transmit-delay 35
```

Features this CLI belongs to

- OSPFv2 ## ip ospf transmit-delay ### Description

Configures OSPFv2 interface LSA transmit delay interval.

Parent Commands (Modes)

interface <vlan-if-name>

Syntax

```
ip ospf transmit-delay <transmitdelay> [ <ip-address> ]  
no ip ospf transmit-delay [ <ip-address> ]
```

Parameters

Name	Description	Type
transmitdelay		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface LSA transmit delay interval.

Examples

```
sonic-cli(config-router-ospf)# ip ospf transmit-delay 35
```

Features this CLI belongs to

- OSPFv2 ## ip ospf transmit-delay ### Description

Configures OSPFv2 interface LSA transmit delay interval.

Parent Commands (Modes)

interface Loopback <lo-id>

Syntax

```
ip ospf transmit-delay <transmitdelay> [ <ip-address> ]  
no ip ospf transmit-delay [ <ip-address> ]
```

Parameters

Name	Description	Type
transmitdelay		Integer
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface LSA transmit delay interval.

Examples

```
sonic-cli(config-router-ospf)# ip ospf transmit-delay 35
```

Features this CLI belongs to

- OSPFv2 ## ip ospf transmit-delay ### Description

Configures OSPFv2 interface LSA transmit delay interval.

Parent Commands (Modes)

```
interface <phy-sub-if-name>  
interface PortChannel <lag-id-subid>
```

Syntax

```
ip ospf transmit-delay <transmitdelay> [ <ip-address> ]  
no ip ospf transmit-delay [ <ip-address> ]
```

Parameters

Name	Description	Type
transmitdelay		Integer

Name	Description	Type
ip-address	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 interface LSA transmit delay interval.

Examples

```
sonic-cli(config-router-ospf)# ip ospf transmit-delay 35
```

Features this CLI belongs to

- OSPFv2 ## ip prefix-list ### Description

Build a prefix-list

Parent Commands (Modes)

configure terminal

Syntax

```
ip prefix-list <prefix-name> { seq { <seq-no> { { permit { <ipv4-prefix> { [ ge
    <ge-min-prefix-length> ] } { [ le <le-max-prefix-length> ] } } } | { deny { <ipv4-prefix> {
        [ ge <ge-min-prefix-length> ] } { [ le <le-max-prefix-length> ] } } } } }
no ip prefix-list <prefix-name> { [ seq { <seq-no> { { permit { <ipv4-prefix> { [ ge
    <ge-min-prefix-length> ] } { [ le <le-max-prefix-length> ] } } } | { deny { <ipv4-prefix> {
        [ ge <ge-min-prefix-length> ] } { [ le <le-max-prefix-length> ] } } } } ] }
```

Parameters

Name	Description	Type
prefix-name	WORD	String
seq-no	1-4294967295	Integer
ipv4-prefix	A.B.C.D/mask	String
ge-min-prefix-length		Integer
le-max-prefix-length		Integer

ip reserve

Description

Configures value to reserve host table entries for local hosts.

Parent Commands (Modes)

configure terminal

Syntax

```
ip reserve local-neigh <value>
no ip reserve local-neigh
```

Parameters

Name	Description	Type
value	number of reserved local neighbors	Integer

Usage Guidelines

Use this command to reserve HW host table for local hosts.

Examples

```
sonic-clia(config)# ip reserve local-neigh
```

ip route

Description

Specify static route

Parent Commands (Modes)

configure terminal

Syntax

```
ip route <prefix> { { interface { <ifname> { { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> [ <pref> ] } ] } [ <pref> ] } ] } | { [ tag { <tag-val> [ <pref> ] } ] } | { blackhole { [ tag { <tag-val> [ <pref> ] } ] } [ <pref> ] } | { <next-hop-addr> { { [ interface { <ifname> { { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] { [ track { <trackid> [ <pref> ] } ] } ] } | { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } | { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] } | { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] } | { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] } | { [ track { <trackid> [ <pref> ] } ] } | { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] { [ track { <trackid> [ <pref> ] } ] } ] } }
```

```
no ip route <prefix> { { interface { <ifname> { [ nexthop-vrf <next-hop-vrf> ] } } } | blackhole | { <next-hop-addr> { { [ interface { <ifname> { [ nexthop-vrf <next-hop-vrf> ] } ] } | { [ nexthop-vrf <next-hop-vrf> ] } ] } }
```

Parameters

Name	Description	Type
prefix	A.B.C.D/mask	String
ifname	Interface Type - Ranges	
next-hop-vrf	WORD	String
tag-val	1-4294967295	Integer
pref		Integer
next-hop-addr	A.B.C.D	String
trackid		Integer

ip route vrf

Description

Configure IP Route for a VRF instance

Parent Commands (Modes)

configure terminal

Syntax

```
ip route vrf <vrfname> { <prefix> { { interface { <ifname> { { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> [ <pref> ] } ] } [ <pref> ] } ] } | { [ tag { <tag-val> [ <pref> ] } ] } | { blackhole { [ tag { <tag-val> [ <pref> ] } ] } [ <pref> ] } | { <next-hop-addr> { { [ interface { <ifname> { { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] { [ track { <trackid> [ <pref> ] } ] } ] } | { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } | { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] } | { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] } | { [ track { <trackid> [ <pref> ] } ] } | { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] { [ track { <trackid> [ <pref> ] } ] } ] } }
```

```
no ip route vrf <vrfname> { <prefix> { { interface { <ifname> { [ nexthop-vrf <next-hop-vrf> ] } } | blackhole | { <next-hop-addr> { { [ interface { <ifname> { [ nexthop-vrf <next-hop-vrf> ] } ] } | { [ nexthop-vrf <next-hop-vrf> ] } ] } } | { [ nexthop-vrf <next-hop-vrf> ] } ] } }
```

Parameters

Name	Description	Type
vrfname	VRF name (prefixed by Vrf, Max: 15 characters)	String
prefix	A.B.C.D/mask	String
ifname	Interface Type - Ranges	
next-hop-vrf	WORD	String
tag-val	1-4294967295	Integer
pref		Integer
next-hop-addr	A.B.C.D	String
trackid		Integer

ip route vrf mgmt

Description

Configure IP Route for Management VRF

Parent Commands (Modes)

configure terminal

Syntax

```
ip route vrf mgmt <prefix> { { interface { <ifname> { { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> [ <pref> ] } ] } [ <pref> ] } ] } | { [ tag { <tag-val> [ <pref> ] } ] } | [ <pref> ] } } | { blackhole { [ tag { <tag-val> [ <pref> ] } ] } [ <pref> ] } | { <next-hop-addr> { { [ interface { <ifname> { { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] { [ track { <trackid> [ <pref> ] } ] } ] } | { [ track { <trackid> [ <pref> ] } ] } | { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } | [ <pref> ] ] } ] } | { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } | [ <pref> ] ] } | { [ track { <trackid> [ <pref> ] } ] } | { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] { [ track { <trackid> [ <pref> ] } ] } ] } ] } }
```

Parameters

Name	Description	Type
prefix	A.B.C.D/mask	String
ifname	Interface Type - Ranges	
next-hop-vrf	WORD	String
tag-val	1-4294967295	Integer
pref		Integer
next-hop-addr	A.B.C.D	String
trackid		Integer

ip sla

Description

```
Configure IP SLA instance
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip sla <sla-id>
no ip sla <sla-id>
```

Parameters

Name	Description	Type
sla-id		Integer

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-10)#[
```

ip source binding

Description

Create a static DHCPv4 snooping binding entry

Parent Commands (Modes)

configure terminal

Syntax

```
ip source binding <ip-address> { <mac-address> { vlan { <phy-if-name> | PortChannel } } }  
no ip source binding <ip-address> { <mac-address> { vlan { <phy-if-name> | PortChannel } } }
```

Parameters

Name	Description	Type
ip-address	A.B.C.D	String
mac-address	nn:nn:nn:nn:nn:nn	String
phy-if-name	EthernetNUM	

ip unnumbered

Description

Configure IPv4 unnumbered interface by borrowing IPv4 address from the Donor interface.

Parent Commands (Modes)

```
interface <phy-sub-if-name>  
interface PortChannel <lag-id-subid>
```

Syntax

```
ip unnumbered <donor-interface>  
no ip unnumbered
```

Parameters

Name	Description	Type
donor-interface		

Usage Guidelines

Use this command to configure IPv4 unnumbered interface at the interface level.

Examples

```
sonic-clia(config)# interface Ethernet 0
sonic-clia(conf-if-Ethernet0)# ip unnumbered Loopback1
```

Features this CLI belongs to

- IPv4 Unnumbered Interface ## ip unnumbered ### Description

Configure IPv4 unnumbered interface by borrowing IPv4 address from the Donor interface.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip unnumbered <donor-interface>
no ip unnumbered
```

Parameters

Name	Description	Type
donor-interface		

Usage Guidelines

Use this command to configure IPv4 unnumbered interface at the interface level.

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# ip unnumbered Loopback1
```

Features this CLI belongs to

- IPv4 Unnumbered Interface ## ip unnumbered ### Description

Configure IPv4 unnumbered interface by borrowing IPv4 address from the Donor interface.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip unnumbered <donor-interface>
no ip unnumbered
```

Parameters

Name	Description	Type
donor-interface		

Usage Guidelines

Use this command to configure IPv4 unnumbered interface at the interface level.

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# ip unnumbered Loopback1
```

Features this CLI belongs to

- IPv4 Unnumbered Interface ## ip unnumbered ### Description

Configure IPv4 unnumbered interface by borrowing IPv4 address from the Donor interface.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip unnumbered <donor-interface>
no ip unnumbered
```

Parameters

Name	Description	Type
donor-interface		

Usage Guidelines

Use this command to configure IPv4 unnumbered interface at the interface level.

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# ip unnumbered Loopback1
```

Features this CLI belongs to

- IPv4 Unnumbered Interface ## ip vrf ### Description

Configure a data VRF.

Configure nexthop resolution via default route.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip vrf <vrf-name> { [ nht resolve-via-default ] }
no ip vrf <vrf-name> { [ nht resolve-via-default ] }
```

Parameters

Name	Description	Type
vrf-name	WORD	String

Usage Guidelines

```
sonic(config)# ip vrf VRF
VRF: Name of VRF (Max: 15 characters, prefixed with Vrf)
sonic(config)# ip vrf VRF nht resolve-via-default
VRF: Name of VRF (Max: 15 characters, prefixed with Vrf)
```

Examples

```
sonic# configure terminal
sonic(config)# ip vrf Vrf_red
sonic(config)# ip vrf Vrf_red nht resolve-via-default
```

ip vrf forwarding

Description

Assign an interface to a VRF. VRF is data/mgmt VRF and interface can be ethernet, loopback, portchannel or vlan interface.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ip vrf forwarding { mgmt | <vrf-name> }
no ip vrf forwarding { mgmt | <vrf-name> }
```

Parameters

Name	Description	Type
vrf-name	WORD	String

Usage Guidelines

```
sonic(conf-if-INTF)# ip vrf forwarding VRF
VRF: Name of VRF
INTF: View identifier of Ethernet, Loopback, PortChannel or Vlan interface
```

Examples

```
sonic(config)# interface Ethernet 8
sonic(conf-if-Ethernet8)# ip vrf forwarding Vrf_red
sonic(config)# interface Loopback 8
sonic(conf-if-lo8)# ip vrf forwarding Vrf_red
sonic(config)# interface PortChannel 8
sonic(conf-if-po8)# ip vrf forwarding Vrf_red
sonic(config)# interface Vlan 8
sonic(conf-if-Vlan8)# ip vrf forwarding Vrf_red
sonic(config)# interface Vlan 100
sonic(conf-if-Vlan8)# ip vrf forwarding mgmt
```

ip vrf forwarding

Description

Assign an interface to a VRF. VRF is data/mgmt VRF and interface can be ethernet, loopback, portchannel or vlan interface.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ip vrf forwarding { mgmt | <vrf-name> }
no ip vrf forwarding { mgmt | <vrf-name> }
```

Parameters

Name	Description	Type
vrf-name	WORD	String

Usage Guidelines

```
sonic(conf-if-INTF)# ip vrf forwarding VRF
VRF: Name of VRF
INTF: View identifier of Ethernet, Loopback, PortChannel or Vlan interface
```

Examples

```
sonic(config)# interface Ethernet 8
sonic(conf-if-Ethernet8)# ip vrf forwarding Vrf_red
sonic(config)# interface Loopback 8
sonic(conf-if-lo8)# ip vrf forwarding Vrf_red
sonic(config)# interface PortChannel 8
sonic(conf-if-po8)# ip vrf forwarding Vrf_red
sonic(config)# interface Vlan 8
sonic(conf-if-Vlan8)# ip vrf forwarding Vrf_red
sonic(config)# interface Vlan 100
sonic(conf-if-Vlan8)# ip vrf forwarding mgmt
```

ip vrf forwarding

Description

Configure forwarding table

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
ip vrf forwarding { mgmt | <vrf-name> }
no ip vrf forwarding { mgmt | <vrf-name> }
```

Parameters

Name	Description	Type
vrf-name	WORD	String

ip vrf forwarding

Description

Assign an interface to a VRF. VRF is data/mgmt VRF and interface can be ethernet, loopback, portchannel or vlan interface.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ip vrf forwarding { mgmt | <vrf-name> }
no ip vrf forwarding { mgmt | <vrf-name> }
```

Parameters

Name	Description	Type
vrf-name	WORD	String

Usage Guidelines

```
sonic(conf-if-INTF)# ip vrf forwarding VRF
VRF: Name of VRF
INTF: View identifier of Ethernet, Loopback, PortChannel or Vlan interface
```

Examples

```
sonic(config)# interface Ethernet 8
sonic(config-Ethernet8)# ip vrf forwarding Vrf_red
sonic(config)# interface Loopback 8
sonic(config-lo8)# ip vrf forwarding Vrf_red
sonic(config)# interface PortChannel 8
sonic(config-po8)# ip vrf forwarding Vrf_red
sonic(config)# interface Vlan 8
sonic(config-Vlan8)# ip vrf forwarding Vrf_red
sonic(config)# interface Vlan 100
sonic(config-Vlan8)# ip vrf forwarding mgmt
```

ip vrf forwarding

Description

Configure forwarding table

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
interface range po_range_num
```

Syntax

```
ip vrf forwarding { mgmt | <vrf-name> }
no ip vrf forwarding { mgmt | <vrf-name> }
```

Parameters

Name	Description	Type
vrf-name	WORD	String

ip vrf forwarding

Description

Assign an interface to a VRF. VRF is data/mgmt VRF and interface can be ethernet, loopback, portchannel or vlan interface.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ip vrf forwarding { mgmt | <vrf-name> }
no ip vrf forwarding { mgmt | <vrf-name> }
```

Parameters

Name	Description	Type
vrf-name	WORD	String

Usage Guidelines

```
sonic(conf-if-INTF)# ip vrf forwarding VRF
VRF: Name of VRF
INTF: View identifier of Ethernet, Loopback, PortChannel or Vlan interface
```

Examples

```
sonic(config)# interface Ethernet 8
sonic(conf-if-Ethernet8)# ip vrf forwarding Vrf_red
sonic(config)# interface Loopback 8
sonic(conf-if-lo8)# ip vrf forwarding Vrf_red
sonic(config)# interface PortChannel 8
sonic(conf-if-po8)# ip vrf forwarding Vrf_red
sonic(config)# interface Vlan 8
sonic(conf-if-Vlan8)# ip vrf forwarding Vrf_red
sonic(config)# interface Vlan 100
sonic(conf-if-Vlan8)# ip vrf forwarding mgmt
```

ip vrf forwarding

Description

Configure forwarding table

Parent Commands (Modes)

```
interface range create vlan_range_num
interface range vlan_range_num
```

Syntax

```
ip vrf forwarding { mgmt | <vrf-name> }
no ip vrf forwarding { mgmt | <vrf-name> }
```

Parameters

Name	Description	Type
vrf-name	WORD	String

ip vrf forwarding

Description

Assign an interface to a VRF. VRF is data/mgmt VRF and interface can be ethernet, loopback, portchannel or vlan interface.

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ip vrf forwarding { mgmt | <vrf-name> }
no ip vrf forwarding { mgmt | <vrf-name> }
```

Parameters

Name	Description	Type
vrf-name	WORD	String

Usage Guidelines

```
sonic(conf-if-INTF)# ip vrf forwarding VRF
VRF: Name of VRF
INTF: View identifier of Ethernet, Loopback, PortChannel or Vlan interface
```

Examples

```
sonic(config)# interface Ethernet 8
sonic(config-Ethernet8)# ip vrf forwarding Vrf_red
sonic(config)# interface Loopback 8
sonic(config-if-lo8)# ip vrf forwarding Vrf_red
sonic(config)# interface PortChannel 8
sonic(config-if-po8)# ip vrf forwarding Vrf_red
sonic(config)# interface Vlan 8
sonic(config-if-Vlan8)# ip vrf forwarding Vrf_red
sonic(config)# interface Vlan 100
sonic(config-if-Vlan8)# ip vrf forwarding mgmt
```

ip vrf mgmt

Description

Configure management VRF.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ip vrf mgmt
no ip vrf mgmt
```

Usage Guidelines

```
sonic(config)# ip vrf mgmt
```

Examples

```
sonic# configure terminal
sonic(config)# ip vrf mgmt
```

ipv6 access-group

Description

```
Apply IPv6 ACL to an interface.
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 access-group <access-list-name> { in | out }
no ipv6 access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	WORD	String

Usage Guidelines

ACL must be created first and must be of type IPv6 to be applied. Only 1 ACL of a given type can be applied per interface and per direction.

Examples

```
sonic(conf-if-Eth1/1)# ipv6 access-group ipv6acl-example in
```

ipv6 access-group

Description

Apply IPv6 ACL to an interface.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ipv6 access-group <access-list-name> { in | out }
no ipv6 access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	WORD	String

Usage Guidelines

ACL must be created first and must be of type IPv6 to be applied. Only 1 ACL of a given type can be applied per interface and per direction.

Examples

```
sonic(conf-if-Eth1/1)# ipv6 access-group ipv6acl-example in
```

ipv6 access-group

Description

Apply IPv6 ACL to an interface.

Parent Commands (Modes)

interface <vlan-if-name>

Syntax

```
ipv6 access-group <access-list-name> { in | out }
no ipv6 access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	WORD	String

Usage Guidelines

ACL must be created first and must be of type IPv6 to be applied. Only 1 ACL of a given type can be applied per interface and per direction.

Examples

```
sonic(conf-if-Eth1/1)# ipv6 access-group ipv6acl-example in
```

ipv6 access-group

Description

Apply IPv6 ACL to an interface.

Parent Commands (Modes)

interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>

Syntax

```
ipv6 access-group <access-list-name> { in | out }
no ipv6 access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	WORD	String

Usage Guidelines

ACL must be created first and must be of type IPv6 to be applied. Only 1 ACL of a given type can be applied per interface and per direction.

Examples

```
sonic(conf-if-Eth1/1)# ipv6 access-group ipv6acl-example in
```

ipv6 access-group

Description

Apply IPv6 ACL globally

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ipv6 access-group <access-list-name> { in | out }
no ipv6 access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	String(Max: 72 characters)	String

Usage Guidelines

ACL must be created first and must be of type IPv6 to be applied. Only 1 ACL of a given type can be applied globally per direction.

Examples

```
sonic(config)# ipv6 access-group ipv6acl-example out
```

ipv6 access-list

Description

Create IPv6 ACL

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ipv6 access-list <access-list-name>
no ipv6 access-list <access-list-name>
```

Parameters

Name	Description	Type
access-list-name	String(Max: 72 characters)	String

Usage Guidelines

ACL name can be of maximum 63 characters. The name must begin with A-Z, a-z or 0-9. Underscore and hyphens can be used except as the first character. ACL name must be unique across all ACL types.

Examples

```
sonic(config)# ipv6 access-list ipv6acl-example
```

ipv6 address

Description

IPv6 address

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
ipv6 address <addr>
no ipv6 address [ <addr> ]
```

Parameters

Name	Description	Type
addr	A::B/mask	String

ipv6 address

Description

IPv6 address

Parent Commands (Modes)

interface Management <mgmt-if-id>

Syntax

```
ipv6 address <addr> [ gwaddr <gw_addr> ]
no ipv6 address [ <addr> ]
```

Parameters

Name	Description	Type
addr	A::B/mask	String
gw_addr	A::B	String

ipv6 address

Description

IPv6 address

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ipv6 address <addr>
no ipv6 address [ <addr> ]
```

Parameters

Name	Description	Type
addr	A::B/mask	String

ipv6 address

Description

IPv6 address

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ipv6 address <addr>
no ipv6 address [ <addr> ]
```

Parameters

Name	Description	Type
addr	A::B/mask	String

ipv6 address

Description

```
IPv6 address
```

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ipv6 address <addr>
no ipv6 address [ <addr> ]
```

Parameters

Name	Description	Type
addr	A::B/mask	String

ipv6 address

Description

```
IPv6 address
```

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ipv6 address <addr>
no ipv6 address [ <addr> ]
```

Parameters

Name	Description	Type
addr	A::B/mask	String

ipv6 anycast-address

Description

Configures IPv6 Static Anycast Gateway Address for an Interface.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ipv6 anycast-address <anycast-addr>
no ipv6 anycast-address <anycast-addr>
```

Parameters

Name	Description	Type
anycast-addr	A::B/mask	String

Examples

```
sonic(conf-if-Vlan5)# ipv6 anycast-address 50::1/64
```

ipv6 anycast-address

Description

Configures IPv6 Static Anycast Gateway Address for an Ethernet sub-interface.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ipv6 anycast-address <anycast-addr>
no ipv6 anycast-address <anycast-addr>
```

Parameters

Name	Description	Type
anycast-addr	A::B/mask	String

Examples

```
sonic(conf-if-Eth5.1)# ipv6 anycast-address 50::1/64
```

ipv6 anycast-address

Description

Enable/Disable IPv6 Static Anycast Gateway functionality. By default it is enabled.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ipv6 anycast-address { enable | disable }
```

Examples

```
sonic(config)# ipv6 anycast-address enable
```

```
sonic(config)# ipv6 anycast-address disable
```

ipv6 dhcp snooping

Description

```
Enable DHCPv6 snooping globally
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ipv6 dhcp snooping  
no ipv6 dhcp snooping
```

Usage Guidelines

```
[no] ipv6 dhcp snooping
```

Examples

```
sonic-cl# configure terminal  
sonic-cl(config)# ipv6 dhcp snooping  
sonic-cl(config)#
```

```
sonic-cl# configure terminal  
sonic-cl(config)# no ipv6 dhcp snooping  
sonic-cl(config)#
```

Features this CLI belongs to

- DHCP Snooping ## ipv6 dhcp snooping trust ##### Description

```
Mark Trust mode for DHCPv6 snooping on interface(s)
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 dhcp snooping trust  
no ipv6 dhcp snooping trust
```

ipv6 dhcp snooping trust

Description

```
Mark Trust mode for DHCPv6 snooping on interface(s)
```

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]
```

Syntax

```
ipv6 dhcp snooping trust  
no ipv6 dhcp snooping trust
```

ipv6 dhcp snooping trust

Description

```
Mark Trust mode for DHCPv6 snooping on interface(s)
```

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
ipv6 dhcp snooping trust  
no ipv6 dhcp snooping trust
```

ipv6 dhcp snooping trust

Description

```
Mark Trust mode for DHCPv6 snooping on interface(s)
```

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
interface range po_range_num
```

Syntax

```
ipv6 dhcp snooping trust
no ipv6 dhcp snooping trust
```

ipv6 dhcp snooping verify mac-address

Description

```
Enable DHCPv6 snooping mac verification
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ipv6 dhcp snooping verify mac-address
no ipv6 dhcp snooping verify mac-address
```

Usage Guidelines

```
[no] ipv6 dhcp snooping verify mac-address
```

Examples

```
sonic-cl# configure terminal
sonic-cl(config)# ipv6 dhcp snooping verify mac-address
sonic-cl(config)#
```

```
sonic-cl# configure terminal
sonic-cl(config)# no ipv6 dhcp snooping verify mac-address
sonic-cl(config)#
```

Features this CLI belongs to

- DHCP Snooping ## ipv6 dhcp snooping vlan ##### Description

```
Enable DHCPv6 snooping on vlan(s)
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ipv6 dhcp snooping vlan <vlan-range>
no ipv6 dhcp snooping vlan <vlan-range>
```

Parameters

Name	Description	Type
vlan-range	<1..4094>	String

ipv6 dhcp-relay

Description

```
Configure DHCPv6 relay on an interface.
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4>
    ] ] } ] } }
no ipv6 dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } ]
```

Parameters

Name	Description	Type
ipaddr1	A::B	String
vrfName	WORD	String
ipaddr2	A::B	String
ipaddr3	A::B	String
ipaddr4	A::B	String

Usage Guidelines

Use this command to configure a DHCPv6 relay on an interface.

Examples

```
sonic-clia(config)# interface Ethernet 12  
sonic-clia(conf-if-Ethernet0)# ipv6 dhcp-relay 9000::1
```

ipv6 dhcp-relay

Description

Configure DHCPv6 relay on an interface.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]
```

Syntax

```
ipv6 dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4>  
        ] ] } ] } }  
no ipv6 dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } ]
```

Parameters

Name	Description	Type
ipaddr1	A::B	String
vrfName	WORD	String
ipaddr2	A::B	String
ipaddr3	A::B	String
ipaddr4	A::B	String

Usage Guidelines

Use this command to configure a DHCPv6 relay on an interface.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ipv6 dhcp-relay 9000::1
```

ipv6 dhcp-relay

Description

Configure DHCPv6 relay on an interface.

Parent Commands (Modes)

interface <vlan-if-name>

Syntax

```
ipv6 dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4>
    ] ] } ] } }
no ipv6 dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } ]
```

Parameters

Name	Description	Type
ipaddr1	A::B	String
vrfName	WORD	String
ipaddr2	A::B	String
ipaddr3	A::B	String
ipaddr4	A::B	String

Usage Guidelines

Use this command to configure a DHCPv6 relay on an interface.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ipv6 dhcp-relay 9000::1
```

ipv6 dhcp-relay

Description

Configure DHCPv6 relay on a range of interfaces.

Parent Commands (Modes)

interface range iface_range_num

Syntax

```
ipv6 dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4>
    ] ] } ] } }
no ipv6 dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } ]
```

Parameters

Name	Description	Type
ipaddr1	A::B	String
vrfName	WORD	String
ipaddr2	A::B	String
ipaddr3	A::B	String
ipaddr4	A::B	String

Usage Guidelines

Use this command to configure a DHCPv6 relay on a range of interfaces.

Examples

```
sonic-clisonic-clim# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-clisonic-clim# conf-if-range-eth**# ipv6 dhcp-relay 9000::1
```

ipv6 dhcp-relay

Description

Configure DHCPv6 relay on a range of interfaces.

Parent Commands (Modes)

interface range create vlan_range_num
interface range vlan_range_num

Syntax

```
ipv6 dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } }  
no ipv6 dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } ]
```

Parameters

Name	Description	Type
ipaddr1	A::B	String
vrfName	WORD	String
ipaddr2	A::B	String
ipaddr3	A::B	String
ipaddr4	A::B	String

Usage Guidelines

Use this command to configure a DHCPv6 relay on a range of interfaces.

Examples

```
sonic-clia(config)# interface range Ethernet 48-51  
%Info: Configuring only existing interfaces in range  
sonic-clia(conf-if-range-eth**)# ipv6 dhcp-relay 9000::1
```

ipv6 dhcp-relay

Description

Configure DHCPv6 relay on a range of interfaces.

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]  
interface range po_range_num
```

Syntax

```
ipv6 dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4>  
        ] ] } ] } }  
no ipv6 dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } ]
```

Parameters

Name	Description	Type
ipaddr1	A::B	String
vrfName	WORD	String
ipaddr2	A::B	String
ipaddr3	A::B	String
ipaddr4	A::B	String

Usage Guidelines

Use this command to configure a DHCPv6 relay on a range of interfaces.

Examples

```
sonic-clia(config)# interface range Ethernet 48-51  
%Info: Configuring only existing interfaces in range  
sonic-clia(conf-if-range-eth**)# ipv6 dhcp-relay 9000::1
```

ipv6 dhcp-relay

Description

Configure DHCPv6 relay on an interface.

Parent Commands (Modes)

```
interface <phy-sub-if-name>  
interface PortChannel <lag-id-subid>
```

Syntax

```
ipv6 dhcp-relay <ipaddr1> { { [ vrf-name <vrfName> ] } { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4>  
] ] } ] } }  
no ipv6 dhcp-relay [ <ipaddr1> { [ <ipaddr2> { [ <ipaddr3> [ <ipaddr4> ] ] } ] } ]
```

Parameters

Name	Description	Type
ipaddr1	A::B	String

Name	Description	Type
vrfName	WORD	String
ipaddr2	A::B	String
ipaddr3	A::B	String
ipaddr4	A::B	String

Usage Guidelines

Use this command to configure a DHCPv6 relay on an interface.

Examples

```
sonic-clia(config)# interface Ethernet 12
sonic-clia(conf-if-Ethernet0)# ipv6 dhcp-relay 9000::1
```

ipv6 dhcp-relay max-hop-count

Description

Configure the maximum hop count for the DHCPv6 packet.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 dhcp-relay max-hop-count <hop-count>
no ipv6 dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCPv6 packet.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ipv6 dhcp-relay max-hop-count 9
```

ipv6 dhcp-relay max-hop-count

Description

Configure the maximum hop count for the DHCPv6 packet.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ipv6 dhcp-relay max-hop-count <hop-count>
no ipv6 dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCPv6 packet.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ipv6 dhcp-relay max-hop-count 9
```

ipv6 dhcp-relay max-hop-count

Description

Configure the maximum hop count for the DHCPv6 packet.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ipv6 dhcp-relay max-hop-count <hop-count>
no ipv6 dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCPv6 packet.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ipv6 dhcp-relay max-hop-count 9
```

ipv6 dhcp-relay max-hop-count

Description

Configure the maximum hop count for the DHCPv6 packet.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
ipv6 dhcp-relay max-hop-count <hop-count>
no ipv6 dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCPv6 packet.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ipv6 dhcp-relay max-hop-count 9
```

ipv6 dhcp-relay max-hop-count

Description

Configure the maximum hop count for the DHCPv6 packet.

Parent Commands (Modes)

```
interface range create vlan_range_num
interface range vlan_range_num
```

Syntax

```
ipv6 dhcp-relay max-hop-count <hop-count>
no ipv6 dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCPv6 packet.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ipv6 dhcp-relay max-hop-count 9
```

ipv6 dhcp-relay max-hop-count

Description

Configure the maximum hop count for the DHCPv6 packet.

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [
    fallback ] [ fast_rate ]
interface range po_range_num
```

Syntax

```
ipv6 dhcp-relay max-hop-count <hop-count>
no ipv6 dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCPv6 packet.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ipv6 dhcp-relay max-hop-count 9
```

ipv6 dhcp-relay max-hop-count

Description

Configure the maximum hop count for the DHCPv6 packet.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ipv6 dhcp-relay max-hop-count <hop-count>
no ipv6 dhcp-relay max-hop-count
```

Parameters

Name	Description	Type
hop-count		Integer

Usage Guidelines

Use this command to configure the maximum hop count for the DHCPv6 packet.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ipv6 dhcp-relay max-hop-count 9
```

ipv6 dhcp-relay source-interface

Description

Configure the source IPv6 address to be used for relaying the DHCPv6 packets.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }  
no ipv6 dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-cli(config)# interface Ethernet 12  
sonic-cli(conf-if-Ethernet0)# ipv6 dhcp-relay source-interface Ethernet36
```

ipv6 dhcp-relay source-interface

Description

Configure the source IPv6 address to be used for relaying the DHCPv6 packets.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]
```

Syntax

```
ipv6 dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }  
no ipv6 dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-clia(config)# interface Ethernet 12
sonic-clia(conf-if-Ethernet0)# ipv6 dhcp-relay source-interface Ethernet36
```

ipv6 dhcp-relay source-interface

Description

Configure the source IPv6 address to be used for relaying the DHCPv6 packets.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ipv6 dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }
no ipv6 dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-clia(config)# interface Ethernet 12
sonic-clia(conf-if-Ethernet0)# ipv6 dhcp-relay source-interface Ethernet36
```

ipv6 dhcp-relay source-interface

Description

Configure the source IPv6 address to be used for relaying the DHCPv6 packets.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
ipv6 dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }
no ipv6 dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-clia(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-clia(conf-if-range-eth**)# ipv6 dhcp-relay source-interface Ethernet36
```

ipv6 dhcp-relay source-interface

Description

Configure the source IPv6 address to be used for relaying the DHCPv6 packets.

Parent Commands (Modes)

```
interface range create vlan_range_num  
interface range vlan_range_num
```

Syntax

```
ipv6 dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }  
no ipv6 dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51  
%Info: Configuring only existing interfaces in range  
sonic-cli(conf-if-range-eth**)# ipv6 dhcp-relay source-interface Ethernet36
```

ipv6 dhcp-relay source-interface

Description

Configure the source IPv6 address to be used for relaying the DHCPv6 packets.

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]  
interface range po_range_num
```

Syntax

```
ipv6 dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }  
no ipv6 dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51  
%Info: Configuring only existing interfaces in range  
sonic-cli(conf-if-range-eth**)# ipv6 dhcp-relay source-interface Ethernet36
```

ipv6 dhcp-relay source-interface

Description

Configure the source IPv6 address to be used for relaying the DHCPv6 packets.

Parent Commands (Modes)

```
interface <phy-sub-if-name>  
interface PortChannel <lag-id-subid>
```

Syntax

```
ipv6 dhcp-relay source-interface { <intfName> | <pchName> | <vlanName> | <loName> }  
no ipv6 dhcp-relay source-interface
```

Parameters

Name	Description	Type
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	
loName		

Usage Guidelines

Use this command to configure the source-interface.

Examples

```
sonic-cli(config)# interface Ethernet 12  
sonic-cli(conf-if-Ethernet0)# ipv6 dhcp-relay source-interface Ethernet36
```

ipv6 dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 dhcp-relay vrf-select  
no ipv6 dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ipv6 dhcp-relay vrf-select
```

ipv6 dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ipv6 dhcp-relay vrf-select
no ipv6 dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ipv6 dhcp-relay vrf-select
```

ipv6 dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ipv6 dhcp-relay vrf-select
no ipv6 dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-cli(config)# interface Ethernet 12
sonic-cli(conf-if-Ethernet0)# ipv6 dhcp-relay vrf-select
```

ipv6 dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
ipv6 dhcp-relay vrf-select
no ipv6 dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-cli(config)# interface range Ethernet 48-51
%Info: Configuring only existing interfaces in range
sonic-cli(conf-if-range-eth**)# ipv6 dhcp-relay vrf-select
```

ipv6 dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface range create vlan_range_num
interface range vlan_range_num
```

Syntax

```
ipv6 dhcp-relay vrf-select  
no ipv6 dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-clia(config)# interface range Ethernet 48-51  
%Info: Configuring only existing interfaces in range  
sonic-clia(conf-if-range-eth**)# ipv6 dhcp-relay vrf-select
```

ipv6 dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]  
interface range po_range_num
```

Syntax

```
ipv6 dhcp-relay vrf-select  
no ipv6 dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-clia(config)# interface range Ethernet 48-51  
%Info: Configuring only existing interfaces in range  
sonic-clia(conf-if-range-eth**)# ipv6 dhcp-relay vrf-select
```

ipv6 dhcp-relay vrf-select

Description

Configure the VRF selection option.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ipv6 dhcp-relay vrf-select
no ipv6 dhcp-relay vrf-select
```

Usage Guidelines

Use this command to configure the VRF selection option.

Examples

```
sonic-clis(config)# interface Ethernet 12
sonic-clis(conf-if-Ethernet0)# ipv6 dhcp-relay vrf-select
```

ipv6 enable

Description

Enable IPv6

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 enable
no ipv6 enable
```

ipv6 enable

Description

Enable IPv6

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ipv6 enable  
no ipv6 enable
```

ipv6 enable

Description

```
Enable IPv6
```

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ipv6 enable  
no ipv6 enable
```

ipv6 enable

Description

```
Enable IPv6
```

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
ipv6 enable  
no ipv6 enable
```

ipv6 enable

Description

```
Enable IPv6
```

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
ipv6 enable
no ipv6 enable
```

ipv6 enable

Description

```
Enable IPv6
```

Parent Commands (Modes)

```
interface range create vlan_range_num
interface range vlan_range_num
```

Syntax

```
ipv6 enable
no ipv6 enable
```

ipv6 enable

Description

```
Enable IPv6
```

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [
    fallback ] [ fast_rate ]
interface range po_range_num
```

Syntax

```
ipv6 enable
no ipv6 enable
```

ipv6 enable

Description

```
Enable IPv6
```

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ipv6 enable
no ipv6 enable
```

ipv6 host-access-list

Description

```
Create ipv6 host-ACL
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ipv6 host-access-list <access-list-name>
no ipv6 host-access-list <access-list-name>
```

Parameters

Name	Description	Type
access-list-name	String(Max: 72 characters)	String

Usage Guidelines

ACL name can be of maximum 72 characters. The name must begin with A-Z, a-z or 0-9. Underscore and hyphens can be used except as the first character. ACL name must be unique across all ACL types.

Examples

```
sonic(config)# ipv6 host-access-list ipv6acl-example
```

ipv6 nd

Description

```
,
```

Dont send router advertisement messages. The no form of this command enables sending RA messages.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 nd { [ suppress-ra ] | [ managed-config-flag ] | [ other-config-flag ] | [ home-agent-config-flag ] | [ ra-fast-retrans ] | [ adv-interval-option ] | { [ dnssl { <SUFFIX> { [ <lifetime_interval> ] | [ infinite ] ] } } ] } | { [ home-agent-lifetime <home-agent-lifetime-interval> ] } | { [ home-agent-preference <home-agent-preference-interval> ] } | { [ mtu <mtu-value> ] } | { [ prefix { <ra-prefix> { [ <lifetime_interval> { <preferred_lifetime_interval> | infinite } ] } | { infinite { <preferred_lifetime_interval> | infinite1 } ] } } { [ router-address ] | { [ off-link [ no-autoconfig ] ] } | { [ no-autoconfig [ off-link ] ] } ] } } | { [ ra-hop-limit <hop-limit-value> ] } | { [ ra-interval { <ra-interval-sec> } ] } | { [ ra-lifetime <ra-lifetime-value> ] } | { [ ra-retrans-interval <ra-retrans-value> ] } | { [ rdnss { <rdnss-prefix> { [ <lifetime_interval> ] | [ infinite ] ] } } ] } | { [ reachable-time <reachable-time-value> ] } | { [ router-preference { high | low | medium } ] } }
```

```
no ipv6 nd { [ suppress-ra ] | [ managed-config-flag ] | [ other-config-flag ] | [ home-agent-config-flag ] | [ ra-fast-retrans ] | [ adv-interval-option ] | { [ dnssl <SUFFIX> ] } | [ home-agent-lifetime ] | [ home-agent-preference ] | [ mtu ] | { [ prefix <ra-prefix> ] } | [ ra-hop-limit ] | { [ ra-interval { [ msec ] ] } ] } | [ ra-lifetime ] | [ ra-retrans-interval ] | { [ rdnss <rdnss-prefix> ] } | [ reachable-time ] | [ router-preference ] }
```

Parameters

Name	Description	Type
SUFFIX	SUFFIX	String
lifetime_interval	(0-4294967295)	Integer
home-agent-lifetime-interval	(0-65520)	Integer
home-agent-preference-interval	(0-65535)	Integer
mtu-value	(1-65535)	Integer
ra-prefix	X:X::X:X/M	String
preferred_lifetime_interval	(0-4294967295)	Integer
hop-limit-value	(0-255)	Integer
ra-interval-sec	(1-1800)	Integer
ra-lifetime-value	(0-9000)	Integer
ra-retrans-value	(0-4294967295)	Integer

Name	Description	Type
rdnss-prefix	X:X::X:X	String
reachable-time-value	1-3600000	Integer

Usage Guidelines

Use this command to suppress sending router advertisement messages. By default, sending router advertisement messages are suppressed.

Examples

```
sonic(conf-if-Ethernet46)# ipv6 nd suppress-ra
```

Features this CLI belongs to

- ND ## ipv6 nd ### Description

```
,
```

Dont send router advertisement messages. The no form of this command enables sending RA messages.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ipv6 nd { [ suppress-ra ] | [ managed-config-flag ] | [ other-config-flag ] | [ home-agent-config-flag ] | [ ra-fast-retrans ] | [ adv-interval-option ] | { [ dnssl { <SUFFIX> { [ <lifetime_interval> ] | [ infinite ] ] } } ] } | { [ home-agent-lifetime <home-agent-lifetime-interval> ] } | { [ home-agent-preference <home-agent-preference-interval> ] } | { [ mtu <mtu-value> ] } | { [ prefix { <ra-prefix> { [ <lifetime_interval> { <preferred_lifetime_interval> | infinite } ] } } | { [ infinite { <preferred_lifetime_interval> | infinite1 } ] } ] } | { [ router-address ] | { [ off-link [ no-autoconfig ] ] } | { [ no-autoconfig [ off-link ] ] } ] } ] } | { [ ra-hop-limit <hop-limit-value> ] } | { [ ra-interval { <ra-interval-sec> } ] } | { [ ra-lifetime <ra-lifetime-value> ] } | { [ ra-retrans-interval <ra-retrans-value> ] } | { [ rdnss { <rdnss-prefix> { [ <lifetime_interval> ] | [ infinite ] ] } } ] } | { [ reachable-time <reachable-time-value> ] } | { [ router-preference { high | low | medium } ] } }
```

```
no ipv6 nd { [ suppress-ra ] | [ managed-config-flag ] | [ other-config-flag ] | [ home-agent-config-flag ] | [ ra-fast-retrans ] | [ adv-interval-option ] | { [ dnssl <SUFFIX> ] } | { [ home-agent-lifetime ] } | { [ home-agent-preference ] } | { [ mtu ] } | { [ prefix <ra-prefix> ] } | { [ ra-hop-limit ] } | { [ ra-interval { [ msec ] ] } ] } | { [ ra-lifetime ] } | { [ ra-retrans-interval ] } | { [ rdnss <rdnss-prefix> ] } | { [ reachable-time ] } | { [ router-preference ] } }
```

Parameters

Name	Description	Type
SUFFIX	SUFFIX	String
lifetime_interval	(0-4294967295)	Integer
home-agent-lifetime-interval	(0-65520)	Integer
home-agent-preference-interval	(0-65535)	Integer
mtu-value	(1-65535)	Integer
ra-prefix	X:X::X:X/M	String
preferred_lifetime_interval	(0-4294967295)	Integer
hop-limit-value	(0-255)	Integer
ra-interval-sec	(1-1800)	Integer
ra-lifetime-value	(0-9000)	Integer
ra-retrans-value	(0-4294967295)	Integer
rdnss-prefix	X:X::X:X	String
reachable-time-value	1-3600000	Integer

Usage Guidelines

Use this command to suppress sending router advertisement messages. By default, sending router advertisement messages are suppressed.

Examples

```
sonic(conf-if-Ethernet46)# ipv6 nd suppress-ra
```

Features this CLI belongs to

- ND ## ipv6 nd ### Description

,

Dont send router advertisement messages. The no form of this command enables sending RA messages.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ipv6 nd { [ suppress-ra ] | [ managed-config-flag ] | [ other-config-flag ] | [ home-agent-config-flag ] | [ ra-fast-retrans ] | [ adv-interval-option ] | { [ dnssl { <SUFFIX> { [ <lifetime_interval> ] | [ infinite ] } } ] } | { [ home-agent-lifetime <home-agent-lifetime-interval> ] } | { [ home-agent-preference <home-agent-preference-interval> ] } | { [ mtu <mtu-value> ] } | { [ prefix { <ra-prefix> { [ <lifetime_interval> { <preferred_lifetime_interval> | infinite } ] } } | { [ infinite { <preferred_lifetime_interval> | infinite1 } ] } ] } | { [ router-address ] | { [ off-link [
```

```

no-autoconfig ] ] } | { [ no-autoconfig [ off-link ] ] } ] } ] } | { [ ra-hop-limit
<hop-limit-value> ] } | { [ ra-interval { <ra-interval-sec> } ] } | { [ ra-lifetime
<ra-lifetime-value> ] } | { [ ra-retrans-interval <ra-retrans-value> ] } | { [ rdnss {
<rdnss-prefix> { [ <lifetime_interval> ] | [ infinite ] ] } ] } | { [ reachable-time
<reachable-time-value> ] } | { [ router-preference { high | low | medium } ] } }
no ipv6 nd { [ suppress-ra ] | [ managed-config-flag ] | [ other-config-flag ] | [
home-agent-config-flag ] | [ ra-fast-retrans ] | [ adv-interval-option ] | { [ dnssl
<SUFFIX> ] | [ home-agent-lifetime ] | [ home-agent-preference ] | [ mtu ] | { [ prefix
<ra-prefix> ] } | [ ra-hop-limit ] | { [ ra-interval { [ msec ] ] } ] } | [ ra-lifetime ] | [
ra-retrans-interval ] | { [ rdnss <rdnss-prefix> ] } | [ reachable-time ] | [
router-preference ] }

```

Parameters

Name	Description	Type
SUFFIX	SUFFIX	String
lifetime_interval	(0-4294967295)	Integer
home-agent-lifetime-interval	(0-65520)	Integer
home-agent-preference-interval	(0-65535)	Integer
mtu-value	(1-65535)	Integer
ra-prefix	X:X::X:X/M	String
preferred_lifetime_interval	(0-4294967295)	Integer
hop-limit-value	(0-255)	Integer
ra-interval-sec	(1-1800)	Integer
ra-lifetime-value	(0-9000)	Integer
ra-retrans-value	(0-4294967295)	Integer
rdnss-prefix	X:X::X:X	String
reachable-time-value	1-3600000	Integer

Usage Guidelines

Use this command to suppress sending router advertisement messages. By default, sending router advertisement messages are suppressed.

Examples

```
sonic(conf-if-Ethernet46)# ipv6 nd suppress-ra
```

Features this CLI belongs to

- ND ## ipv6 nd ## Description

,

Dont send router advertisement messages. The no form of this command enables sending RA messages.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ipv6 nd { [ suppress-ra ] | [ managed-config-flag ] | [ other-config-flag ] | [
    home-agent-config-flag ] | [ ra-fast-retrans ] | [ adv-interval-option ] | { [ dnssl {
        <SUFFIX> { [ <lifetime_interval> ] | [ infinite ] ] } ] } | { [ home-agent-lifetime
        <home-agent-lifetime-interval> ] } | { [ home-agent-preference
        <home-agent-preference-interval> ] } | { [ mtu <mtu-value> ] } | { [ prefix { <ra-prefix> {
            { [ <lifetime_interval> { <preferred_lifetime_interval> | infinite } ] } | { [ infinite {
                <preferred_lifetime_interval> | infinite1 } ] } ] } { [ router-address ] | { [ off-link [
                    no-autoconfig ] ] } | { [ no-autoconfig [ off-link ] ] } ] } ] } | { [ ra-hop-limit
        <hop-limit-value> ] } | { [ ra-interval { <ra-interval-sec> } ] } | { [ ra-lifetime
        <ra-lifetime-value> ] } | { [ ra-retrans-interval <ra-retrans-value> ] } | { [ rdnss {
            <rdnss-prefix> { [ <lifetime_interval> ] | [ infinite ] ] } ] } | { [ reachable-time
            <reachable-time-value> ] } | { [ router-preference { high | low | medium } ] } }
no ipv6 nd { [ suppress-ra ] | [ managed-config-flag ] | [ other-config-flag ] | [
    home-agent-config-flag ] | [ ra-fast-retrans ] | [ adv-interval-option ] | { [ dnssl
        <SUFFIX> ] } | [ home-agent-lifetime ] | [ home-agent-preference ] | [ mtu ] | { [ prefix
        <ra-prefix> ] } | [ ra-hop-limit ] | { [ ra-interval { [ msec ] ] } ] } | [ ra-lifetime ] |
        [ ra-retrans-interval ] | { [ rdnss <rdnss-prefix> ] } | [ reachable-time ] | [
            router-preference ] }
```

Parameters

Name	Description	Type
SUFFIX	SUFFIX	String
lifetime_interval	(0-4294967295)	Integer
home-agent-lifetime-interval	(0-65520)	Integer
home-agent-preference-interval	(0-65535)	Integer
mtu-value	(1-65535)	Integer
ra-prefix	X:X::X:X/M	String
preferred_lifetime_interval	(0-4294967295)	Integer
hop-limit-value	(0-255)	Integer
ra-interval-sec	(1-1800)	Integer
ra-lifetime-value	(0-9000)	Integer
ra-retrans-value	(0-4294967295)	Integer
rdnss-prefix	X:X::X:X	String
reachable-time-value	1-3600000	Integer

Usage Guidelines

Use this command to suppress sending router advertisement messages. By default, sending router advertisement messages are suppressed.

Examples

```
sonic(conf-if-Ethernet46)# ipv6 nd suppress-ra
```

Features this CLI belongs to

- ND ## ipv6 neighbor ### Description

```
Configure static ND.
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 neighbor <static-ip> <neigh>
no ipv6 neighbor <static-ip> <neigh>
```

Parameters

Name	Description	Type
static-ip	A::B	String
neigh	nn:nn:nn:nn:nn:nn	String

Usage Guidelines

```
Use this command to configure static ND.
```

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# ip neighbor ip_addr mac
```

ipv6 neighbor

Description

```
Configure static ND.
```

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
ipv6 neighbor <static-ip> <neigh>  
no ipv6 neighbor <static-ip> <neigh>
```

Parameters

Name	Description	Type
static-ip	A.B.C.D or A:B:C:D:E:F:G:H	String
neigh	nn:nn:nn:nn:nn:nn	String

Usage Guidelines

Use this command to configure static ND.

Examples

```
sonic-clia(config)# interface Ethernet 0  
sonic-clia(conf-if-Ethernet0)# ip neighbor ip_addr mac
```

ipv6 neighbor

Description

Configure static ND.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
ipv6 neighbor <static-ip> <neigh>  
no ipv6 neighbor <static-ip> <neigh>
```

Parameters

Name	Description	Type
static-ip	A::B	String
neigh	nn:nn:nn:nn:nn:nn	String

Usage Guidelines

Use this command to configure static ND.

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# ip neighbor ip_addr mac
```

ipv6 neighbor

Description

Configure static ND.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
ipv6 neighbor <static-ip> <neigh>
no ipv6 neighbor <static-ip> <neigh>
```

Parameters

Name	Description	Type
static-ip	A.B.C.D or A:B:C:D:E:F:G:H	String
neigh	nn:nn:nn:nn:nn:nn	String

Usage Guidelines

Use this command to configure static ND.

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# ipv6 neighbor ip_addr mac
```

ipv6 nht

Description

Configures default route based IPv6 nexthop resolution within default VRF.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ipv6 nht resolve-via-default
no ipv6 nht resolve-via-default
```

Usage Guidelines

Use this command to enable default route based IPv6 nexthop resolution within default VRF.

Examples

```
sonic-cli(config)# ipv6 nht resolve-via-default
```

Features this CLI belongs to

- Nexthop Tracking ## ipv6 ospf6

Description

Configures OSPFv3 parameters within an IPv6 interface.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 ospf6
```

Usage Guidelines

Use this command to configure OSPFv3 parameters under an IPv6 interface. IPv6 interface can be Ethernet interface, VLAN interface, Portchannel interface or a Loopback interface. Every OSPFv3 parameter on an interface can be associated with its specific IPv6 address by explicitly specifying the IPv6 address after the parameter. Specifying interface IPv6 address is an optional.

Examples

```
sonic-clt(config-router-ospf6)# no ipv6 ospf6 interface
```

Features this CLI belongs to

- OSPFv3

ipv6 ospf6 area

Description

Configures OSPFv3 interface area identifier.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 ospf6 area <area-id>
no ipv6 ospf6 area
```

Parameters

Name	Description	Type
area-id	A.B.C.D or 0..4294967295	String

Usage Guidelines

Use this command to associate an interface into an OSPFv3 area. Area identifier can be configured only when there is an already configured OSPFv3 router within the interface VRF and there are no network commands configured within that router. Area identifier configuration on an interface will get auto unconfigured while OSPFv3 router gets unconfigured from the VRF.

Examples

```
sonic-cli(config-router-ospf6)# ipv6 ospf6 area 19
sonic-cli(config-router-ospf6)# ipv6 ospf6 area 19.0.0.1
```

Features this CLI belongs to

- OSPFv3 ## ipv6 ospf6 bfd ##### Description

Configures OSPFv3 interface BFD.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 ospf6 bfd
no ipv6 ospf6 bfd
```

Usage Guidelines

Use this command to configure OSPFv3 interface BFD. Enabling BFD will establish a BFD session between the OSPF neighbors. Any failure in BFD session will bring down the OSPFv3 session.

Examples

```
sonic-cli(config-router-ospf6)# ipv6 ospf6 bfd
```

Features this CLI belongs to

- OSPFv3 ## ipv6 ospf6 bfd profile ### Description

Configures OSPFv3 interface BFD profile.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
ipv6 ospf6 bfd profile <profilename>
no ipv6 ospf6 bfd profile
```

Parameters

Name	Description	Type
profilename	WORD	String

Usage Guidelines

Use this command to configure OSPFv3 interface BFD profile. Enabling profile will also enable BFD on interface if not enabled.

Examples

```
sonic-cli(config-router-ospf6)# ipv6 ospf6 bfd profile ospf
```

Features this CLI belongs to

- OSPFv3 ## ipv6 ospf6 cost ### Description

Configures OSPFv3 interface cost.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
ipv6 ospf6 cost <interface-cost>
no ipv6 ospf6 cost
```

Parameters

Name	Description	Type
interface-cost		Integer

Usage Guidelines

Use this command to configure OSPFv3 interface cost.

Examples

```
sonic-cli(config-router-ospf6)# ipv6 ospf6 cost 38
```

Features this CLI belongs to

- OSPFv3 ## ipv6 ospf6 dead-interval ##### Description

Configure OSPFv3 adjacency dead interval

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 ospf6 dead-interval [ <deadinterval> ]
no ipv6 ospf6 dead-interval
```

Parameters

Name	Description	Type
deadinterval		Integer

ipv6 ospf6 hello-interval

Description

Configures OSPFv3 interface neighbour hello interval.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
 ipv6 ospf6 hello-interval [ <hellointerval> ]
 no ipv6 ospf6 hello-interval
```

Parameters

Name	Description	Type
hellointerval		Integer

Usage Guidelines

Use this command to configure OSPFv3 interface neighbour hello interval.

Examples

```
sonic-cli(config-router-ospf6)# ipv6 ospf6 thello-interval 20
```

Features this CLI belongs to

- OSPFv3 ## ipv6 ospf6 mtu-ignore ### Description

Disables OSPFv3 MTU mismatch detection.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
ipv6 ospf6 mtu-ignore
no ipv6 ospf6 mtu-ignore
```

Usage Guidelines

Use this command to disable OSPFv3 MTU mismatch detection. MTU mismatch detection is enabled by default.

Examples

```
sonic-cli(config-router-ospf6)# ipv6 ospf6 mtu-ignore
```

Features this CLI belongs to

- OSPFv3 ## ipv6 ospf6 network ### Description

Configures OSPFv3 interface network type.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 ospf6 network { broadcast | point-to-point }
no ipv6 ospf6 network
```

Usage Guidelines

Use this command to configure OSPFv3 interface network type. Broadcast and Point-to-point networks types are supported. By default Network type will be broadcast.

Examples

```
sonic-cli(config-router-ospf6)# ipv6 ospf6 network point-to-point
sonic-cli(config-router-ospf6)# ipv6 ospf6 network broadcast
```

Features this CLI belongs to

- OSPFv3 ## ipv6 ospf6 priority ### Description

Configures OSPFv3 adjacency router priority.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 ospf6 priority <priorityval>
no ipv6 ospf6 priority
```

Parameters

Name	Description	Type
priorityval		Integer

Usage Guidelines

Use this command to configure OSPFv3 adjacency router priority.

Examples

```
sonic-cli(config-router-ospf6)# ipv6 ospf6 priority 19
```

Features this CLI belongs to

- OSPFv3 ## ipv6 ospf6 retransmit-interval ### Description

Configures OSPFv3 interface LSA retransmit interval.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 ospf6 retransmit-interval <retransmitinterval>
no ipv6 ospf6 retransmit-interval
```

Parameters

Name	Description	Type
retransmitinterval		Integer

Usage Guidelines

Use this command to configure OSPFv3 interface LSA retransmit interval.

Examples

```
sonic-clia(config-router-ospf6)# ipv6 ospf6 retransmit-interval 35
```

Features this CLI belongs to

- OSPFv3 ## ipv6 ospf6 transmit-delay ### Description

Configures OSPFv3 interface LSA transmit delay interval.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
ipv6 ospf6 transmit-delay <transmitdelay>
no ipv6 ospf6 transmit-delay
```

Parameters

Name	Description	Type
transmitdelay		Integer

Usage Guidelines

Use this command to configure OSPFv3 interface LSA transmit delay interval.

Examples

```
sonic-cli(config-router-ospf6)# ipv6 ospf6 transmit-delay 35
```

Features this CLI belongs to

- OSPFv3 ## ipv6 prefix-list ### Description

```
Build a prefix-list
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ipv6 prefix-list <prefix-name> { seq { <seq-no> { { permit { <ipv6-prefix> { [ ge <ge-min-prefix-length> ] } { [ le <le-max-prefix-length> ] } } } | { deny { <ipv6-prefix> { [ ge <ge-min-prefix-length> ] } { [ le <le-max-prefix-length> ] } } } } } } no ipv6 prefix-list <prefix-name> { [ seq { <seq-no> { { permit { <ipv6-prefix> { [ ge <ge-min-prefix-length> ] } { [ le <le-max-prefix-length> ] } } } | { deny { <ipv6-prefix> { [ ge <ge-min-prefix-length> ] } { [ le <le-max-prefix-length> ] } } } } } ] }
```

Parameters

Name	Description	Type
prefix-name	WORD	String
seq-no	1-4294967295	Integer
ipv6-prefix	A::B/mask	String
ge-min-prefix-length		Integer
le-max-prefix-length		Integer

ipv6 route

Description

```
Specify static route
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ipv6 route <prefix> { { interface { <ifname> { { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> [ <pref> ] } ] } [ <pref> ] } ] } | { [ tag { <tag-val> [ <pref> ] } ] } | { blackhole { [ tag { <tag-val> [ <pref> ] } ] } [ <pref> ] } | { <next-hop-addr> { { interface { <ifname> { { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] { [ track { <trackid> [ <pref> ] } ] } ] } | { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } | { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] } | { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] } | { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] } | { [ track { <trackid> [ <pref> ] } ] } | { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] { [ track { <trackid> [ <pref> ] } ] } ] } | { [ <pref> ] } | { [ track { <trackid> [ <pref> ] } ] } | { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] } { [ track { <trackid> [ <pref> ] } ] } ] } | { [ <pref> ] } | { [ track { <trackid> [ <pref> ] } ] } | { [ nexthop-vrf { <next-hop-vrf> } ] } } }
```

Parameters

Name	Description	Type
prefix	A::B/mask	String
ifname	Interface Type - Ranges	
next-hop-vrf	WORD	String
tag-val	1-4294967295	Integer
pref		Integer
next-hop-addr	A::B	String
trackid		Integer

ipv6 route vrf

Description

Configure IP Route for a VRF instance

Parent Commands (Modes)

configure terminal

Syntax

```
ipv6 route vrf <vrfname> { <prefix> { { interface { <ifname> { { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> [ <pref> ] } ] } [ <pref> ] } ] } | { [ tag { <tag-val> [ <pref> ] } ] } | { blackhole { [ tag { <tag-val> [ <pref> ] } ] } [ <pref> ] } | { <next-hop-addr> { { interface { <ifname> { { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] { [ track { <trackid> [ <pref> ] } ] } ] } | { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } | { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] } | { [ <pref> ] ] } } | { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] } | { [ <pref> ] ] } } | { [ track { <trackid> [ <pref> ] } ] } | { [ nexthop-vrf { <next-hop-vrf> { [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] { [ track { <trackid> [ <pref> ] } ] } ] } | { [ <pref> ] ] } } }
```

```
no ipv6 route vrf <vrfname> { { prefix> { { interface { <ifname> { [ nexthop-vrf <next-hop-vrf>
] } } } | blackhole | { <next-hop-addr> { { [ interface { <ifname> { [ nexthop-vrf
<next-hop-vrf> ] } } ] } | { [ nexthop-vrf <next-hop-vrf> ] } ] } } }
```

Parameters

Name	Description	Type
vrfname	VRF name (prefixed by Vrf, Max: 15 characters)	String
prefix	A::B/mask	String
ifname	Interface Type - Ranges	
next-hop-vrf	WORD	String
tag-val	1-4294967295	Integer
pref		Integer
next-hop-addr	A::B	String
trackid		Integer

ipv6 route vrf mgmt

Description

Configure IP Route for Management VRF

Parent Commands (Modes)

configure terminal

Syntax

```
ipv6 route vrf mgmt <prefix> { { interface { <ifname> { { [ nexthop-vrf { <next-hop-vrf> { [
tag { <tag-val> [ <pref> ] } ] } [ <pref> ] } ] } | { [ tag { <tag-val> [ <pref> ] } ] } | [
<pref> ] ] } } | { blackhole { [ tag { <tag-val> [ <pref> ] } ] } [ <pref> ] } | {
<next-hop-addr> { { [ interface { <ifname> { { [ nexthop-vrf { <next-hop-vrf> { [ tag {
<tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] { [ track {
<trackid> [ <pref> ] } ] } ] } | { [ track { <trackid> [ <pref> ] } ] } | { [ tag {
<tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } | [ <pref> ] ] } ] } |
{ [ tag { <tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } | [ <pref> ] |
{ [ track { <trackid> [ <pref> ] } ] } | { [ nexthop-vrf { <next-hop-vrf> { [ tag {
<tag-val> { [ track { <trackid> [ <pref> ] } ] } [ <pref> ] } ] } [ <pref> ] { [ track {
<trackid> [ <pref> ] } ] } ] } ] }
no ipv6 route vrf mgmt <prefix> { { interface { <ifname> { [ nexthop-vrf <next-hop-vrf> ] } } } |
blackhole | { <next-hop-addr> { { [ interface { <ifname> { [ nexthop-vrf <next-hop-vrf> ]
} } ] } | { [ nexthop-vrf <next-hop-vrf> ] } ] } }
```

Parameters

Name	Description	Type
prefix	A::B/mask	String
ifname	Interface Type - Ranges	
next-hop-vrf	WORD	String
tag-val	1-4294967295	Integer
pref		Integer
next-hop-addr	A::B	String
trackid		Integer

ipv6 source binding

Description

Create a static DHCPv6 snooping binding entry

Parent Commands (Modes)

configure terminal

Syntax

```
ipv6 source binding <ip-address> { <mac-address> { vlan { <phy-if-name> | PortChannel } } }
no ipv6 source binding <ip-address> { <mac-address> { vlan { <phy-if-name> | PortChannel } } }
```

Parameters

Name	Description	Type
ip-address	A::B	String
mac-address	nn:nn:nn:nn:nn:nn	String
phy-if-name	EthernetNUM	

ipv6 vrf

Description

Configure IPv6 nexthop resolution via default route.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ipv6 vrf <vrf-name> { nht resolve-via-default }
no ipv6 vrf <vrf-name> { nht resolve-via-default }
```

Parameters

Name	Description	Type
vrf-name	WORD	String

Usage Guidelines

```
sonic(config)# ipv6 vrf VRF nht resolve-via-default
VRF: Name of VRF (Max: 15 characters, prefixed with Vrf)
```

Examples

```
sonic# configure terminal
sonic(config)# ipv6 vrf Vrf_red nht resolve-via-default
```

kdump enable

Description

Enable or disable KDUMP operation. These commands require a reboot to complete.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
kdump enable
```

Usage Guidelines

Use the command "kdump enable" to enable the kdump operation.

Use the command "no kdump" to disable the kdump operation.

Examples

```
sonic# configure terminal
sonic(config)# kdump enable
Kdump configuration changes will be applied after the system reboots.
Save SONiC configuration using 'write memory' before issuing the reboot command.
sonic(config)# no kdump
Kdump configuration changes will be applied after the system reboots.
Save SONiC configuration using 'write memory' before issuing the reboot command.
```

Features this CLI belongs to

- KDUMP

Alternate command

```
config kdump enable
```

kdump memory

Description

Set or reset to default the amount of memory reserved for kdump.

These commands require a reboot to complete.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
kdump memory <kdump_memory>
no kdump memory
```

Parameters

Name	Description	Type
kdump_memory		String

Usage Guidelines

Use the commands "kdump memory <X>" or "no kdump memory" to set or reset to default the amount of memory reserved for kdump.

Examples

```
sonic# configure terminal
sonic(config)# kdump memory 512M
Kdump updated memory will be only operational after the system reboots.
Save SONiC configuration using 'write memory' before issuing the reboot command.
sonic(config)# no kdump memory
Kdump updated memory will be only operational after the system reboots.
Save SONiC configuration using 'write memory' before issuing the reboot command.
```

Features this CLI belongs to

- KDUMP #### Alternate command ##### click

```
config kdump memory
```

kdump num-dumps

Description

Set or reset to default value the maximum number of kernel core files stored locally.
These commands typically require a reboot to complete.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
kdump num-dumps <kdump_num.dumps>
no kdump num-dumps
```

Parameters

Name	Description	Type
kdump_num.dumps	Maximum number of kernel core files stored locally	Integer

Usage Guidelines

Use the commands "kdump num-dumps <X>" or "no kdump num-dumps" to set or reset to default the maximum number of kernel core files stored locally.

Examples

```
sonic# configure terminal
sonic(config)# kdump num-dumps 5
sonic(config)# no kdump num-dumps
```

Features this CLI belongs to

- KDUMP #### Alternate command ##### click

```
config kdump num_dumps
```

keepalive-interval

Description

Configures MCLAG keepalive interval in seconds

Parent Commands (Modes)

```
mclag domain <mclag-domain-id>
```

Syntax

```
keepalive-interval <KA>
```

Parameters

Name	Description	Type
KA	Keepalive Interval	Integer

Usage Guidelines

Use this command to change the default MCLAG keepalive interval

Examples

```
sonic-cli(config-mclag-domain-100)#keepalive-interval 10
```

key config-key password-encrypt

Description

This command is used to configure a user selected passphrase that is used to derive the primary encryption key. In the absence of user configured passphrase, the system derives a unique default key.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
key config-key password-encrypt  
no key config-key password-encrypt [ override ]
```

Usage Guidelines

User configures the primary encryption key passphrase from which the primary encryption key will be derived. It is important to note that config cannot be copied/migrated from one box to another unless the same primary encryption key passphrase is configured on both the devices.

Examples

```
sonic#configure  
sonic(conf)#key config-key password-encrypt  
New key:  
Old key:  
Primary encryption key updated successfully.
```

link debounce link-up time

Description

Enables the debounce link-up timer for the amount of time (1 to 5000 ms) specified. If you specify 0 milliseconds, the debounce link-up timer is disabled.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
link debounce link-up time <milliseconds>
no link debounce link-up time
```

Parameters

Name	Description	Type
milliseconds		Integer

Examples

```
sonic(conf-if-Ethernet4)# link debounce link-up 1000
```

link debounce time

Description

Enables the debounce timer for the specified time (1 to 5000 milliseconds). If you specify 0 milliseconds, the debounce timer is disabled.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
link debounce time <milliseconds>
no link debounce time
```

Parameters

Name	Description	Type
milliseconds		Integer

Examples

```
sonic(conf-if-Ethernet4)# link debounce time 1000
```

link state track

Description

Create a link state tracking group.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
link state track <grp-name>
no link state track <grp-name>
```

Parameters

Name	Description	Type
grp-name	name	String

Usage Guidelines

Link state tracking group name can be of maximum 63 characters. The name must begin with A-Z, a-z or 0-9. Underscore and hyphens can be used except as the first character.

Examples

```
sonic(config)# link state track FooBar
```

Alternate command

```
admin@sonic:~$ sudo config linktrack add <name>
```

link state track

Description

Configure upstream or downstream interfaces.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
link state track <grp-name> { upstream | downstream }
no link state track { <grp-name> | upstream | downstream } { upstream | downstream }
```

Parameters

Name	Description	Type
grp-name	name	String

Examples

```
sonic(conf-if-Vlan100)# link state track FooBar upstream
sonic(conf-if-Ethernet4)# link state track FooBar downstream
```

Alternate command

```
admin@sonic:~$ sudo config linktrack update <name> --upstream <interfaces> --downstream
<interfaces>
```

link state track

Description

Configure upstream or downstream interfaces.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [
    fallback ] [ fast_rate ]
```

Syntax

```
link state track <grp-name> { upstream | downstream }
no link state track { <grp-name> | upstream | downstream } { upstream | downstream }
```

Parameters

Name	Description	Type
grp-name	name	String

Examples

```
sonic(conf-if-Vlan100)# link state track FooBar upstream
sonic(conf-if-Ethernet4)# link state track FooBar downstream
```

Alternate command

```
admin@sonic:~$ sudo config linktrack update <name> --upstream <interfaces> --downstream
<interfaces>
```

link state track

Description

Configure upstream or downstream interfaces.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
link state track <grp-name> { upstream | downstream }
no link state track { <grp-name> | upstream | downstream } { upstream | downstream }
```

Parameters

Name	Description	Type
grp-name	name	String

Examples

```
sonic(conf-if-Vlan100)# link state track FooBar upstream
sonic(conf-if-Ethernet4)# link state track FooBar downstream
```

Alternate command

```
admin@sonic:~$ sudo config linktrack update <name> --upstream <interfaces> --downstream
<interfaces>
```

link state track

Description

Configure upstream or downstream interfaces.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
link state track <grp-name> { upstream | downstream }
no link state track { <grp-name> | upstream | downstream } { upstream | downstream }
```

Parameters

Name	Description	Type
grp-name	name	String

Examples

```
sonic(conf-if-Vlan100)# link state track FooBar upstream
sonic(conf-if-Ethernet4)# link state track FooBar downstream
```

Alternate command

```
admin@sonic:~$ sudo config linktrack update <name> --upstream <interfaces> --downstream
<interfaces>
```

link-error-disable

Description

```
Configure link error disable
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
link-error-disable [ flap-threshold { <flaps> { sampling-interval { <window> {
    recovery-interval <interval> } } } } ]
no link-error-disable
```

Parameters

Name	Description	Type
flaps		Integer
window		Integer
interval		Integer

link-error-disable

Description

```
Configure link error disable
```

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
link-error-disable [ flap-threshold { <flaps> { sampling-interval { <window> {  
    recovery-interval <interval> } } } ]  
no link-error-disable
```

Parameters

Name	Description	Type
flaps		Integer
window		Integer
interval		Integer

listen limit

Description

This command sets listen limit for BGP for dynamic BGP neighbors.

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
listen limit <lmt-val>
```

Parameters

Name	Description	Type
lmt-val		Integer

Usage Guidelines

Use this command to configure Maximum number of BGP Dynamic Neighbors that can be created.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# listen limit 123
```

listen range

Description

This command creates a listen range for BGP for dynamic BGP neighbors. BGP will accept connections from any peers in the specified prefix. Configuration from the specified peer-group is used to configure these peers.

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
listen range <addr> { peer-group <pgname> }
```

Parameters

Name	Description	Type
addr	A.B.C.D/mask or A::B/mask	String
pgname	WORD	String

Usage Guidelines

Use this command to accept peering connection from neighbors and create dynamic neighbors

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# listen range 192.168.0.0/16 peer-group PG_Ext
```

lldp

Description

Configure LLDP frame Reception and Transmission mode.

Parent Commands (Modes)

configure terminal

Syntax

```
lldp <mode>
no lldp <mode>
```

Parameters

Name	Description	Type
mode	Mode type	Select [receive transmit]

Usage Guidelines

Use this command to configure LLDP frame Reception and Transmission mode.

Examples

```
sonic-cli(config)# lldp receive
      or
sonic-cli(config)# lldp transmit
```

Features this CLI belongs to

- LLDP ## lldp ### Description

Configure LLDP frame Reception and Transmission mode at interface level.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
lldp <mode>
no lldp <mode>
```

Parameters

Name	Description	Type
mode	Mode type	Select [receive transmit]

Usage Guidelines

Use this command to configure LLDP frame Reception and Transmission mode at interface level.

Examples

```
sonic-clia(config)# interface Ethernet 0  
sonic-clia(conf-if-Ethernet0)# lldp receive
```

Features this CLI belongs to

- LLDP ## lldp enable ### Description

Enable LLDP at global level

Parent Commands (Modes)

configure terminal

Syntax

```
lldp enable  
no lldp enable
```

Usage Guidelines

Use this command to enable LLDP globally

Examples

```
sonic-clia(config)# lldp enable
```

Features this CLI belongs to

- LLDP ## lldp enable ### Description

Enable LLDP at interface level

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
lldp enable  
no lldp enable
```

Usage Guidelines

Use this command to enable LLDP at interface level

Examples

```
sonic-clia(config)# interface Ethernet 0  
sonic-clia(conf-if-Ethernet0)# lldp enable
```

Features this CLI belongs to

- LLDP ## lldp med-tlv-select ### Description

Enables LLDP-MED TLV advertisement.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
lldp med-tlv-select <tlv>  
no lldp med-tlv-select <tlv>
```

Parameters

Name	Description	Type
tlv	MED TLV type	Select [network-policy power-management]

Usage Guidelines

Use this command to select which LLDP-MED TLVs to be advertised.

Examples

```
sonic-cl(i(config)# interface Ethernet 0
sonic-cl(i(conf-if-Ethernet0)# lldp med-tlv-select network-policy
```

Features this CLI belongs to

- LLDP-MED ## lldp multiplier ##### Description

Configure LLDP multiplier value that is used to determine the timeout interval (i.e. hello-time x multiplier value) after which LLDP neighbor entry is deleted

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
lldp multiplier <multiplier>
no lldp multiplier
```

Parameters

Name	Description	Type
multiplier		Integer

Usage Guidelines

Use this command to set LLDP multiplier value. Default value is 4.

Examples

```
sonic-cli(config)# lldp multiplier 6
```

Features this CLI belongs to

- LLDP ## lldp system-description ### Description

```
Configure LLDP System description.
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
lldp system-description <system_description>
no lldp system-description
```

Parameters

Name	Description	Type
system_description	Line	String

Usage Guidelines

```
Use this command to configure LLDP system description.
```

Examples

```
sonic-cli(config)# lldp system-description "Broadcom Sonic"
```

Features this CLI belongs to

- LLDP ## lldp system-name ### Description

```
Configure LLDP System name.
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
lldp system-name <system_name>
no lldp system-name
```

Parameters

Name	Description	Type
system_name	String	String

Usage Guidelines

Use this command to configure LLDP system name.

Examples

```
sonic-clia(config)# lldp system-name "BroadcomSonic"
```

Features this CLI belongs to

- LLDP ## lldp timer ### Description

Configure LLDP hello time. It is the time interval at which periodic hellos are exchanged.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
lldp timer <hello-time>
no lldp timer
```

Parameters

Name	Description	Type
hello-time		Integer

Usage Guidelines

Use this command to set LLDP hello time. Default hello time is 30 seconds.

Examples

```
sonic-cli(config)# lldp timer 10
```

Features this CLI belongs to

- LLDP ## lldp tlv-select ##### Description

Enable sending of TLVs in LLDP frames.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
lldp tlv-select <tlv>
no lldp tlv-select <tlv>
```

Parameters

Name	Description	Type
tlv	TLV type	Select [management-address system-capabilities]

Usage Guidelines

Use this command to configure LLDP TLVs to be advertised

Examples

```
sonic-cli(config)# lldp tlv-select system-capabilities
sonic-cli(config)# lldp tlv-select management-address
```

Features this CLI belongs to

- LLDP ## lldp tlv-select ### Description

Enables LLDP 802.3 TLV advertisement.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
lldp tlv-select <tlv>
no lldp tlv-select <tlv>
```

Parameters

Name	Description	Type
tlv	802.3 TLV type	Select [power-management]

Usage Guidelines

Use this command to select which LLDP 802.3 TLVs to be advertised.

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# lldp tlv-select power-management
```

Features this CLI belongs to

- LLDP-MED ## lldp tlv-set ### Description

Configure an IPv4 or IPv6 management address that will be used to advertise by LLDP on an interface.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
lldp tlv-set management-address { { ipv4 <address> } | { ipv6 <address> } }
no lldp tlv-set management-address { ipv4 | ipv6 }
```

Parameters

Name	Description	Type
address	A.B.C.D	String

Usage Guidelines

Use this command to configure an IPv4 or IPv6 management address that will be used to advertise by LLDP on an interface.

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# lldp tlv-set management-address ipv4 10.1.1.1
```

Features this CLI belongs to

- LLDP ## local-as ### Description

This command specifies an alternate AS for this BGP process when interacting with the specified peer. With no modifiers, the specified local-as is prepended to the received AS_PATH when receiving routing updates from the peer, and prepended to the outgoing AS_PATH (after the process local AS) when transmitting local routes to the peer.

If the no-prepend CLI option is specified, then the supplied local-as is not prepended to the received AS_PATH.

If the replace-as CLI option is specified, then only the supplied local-as is prepended to the AS_PATH when transmitting local-route updates to this peer.

Note that replace-as can only be specified if no-prepend is.

This command is only allowed for eBGP peers.

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
local-as <asnum> { [ no-prepend [ replace-as ] ] }
no local-as
```

Parameters

Name	Description	Type
asnum	1-4294967295	Integer

Usage Guidelines

Use this command to configure local AS number for a BGP neighbor and control how the local AS number is prepended to the AS_PATH of incoming and outgoing routes.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# local-as 65200 no-prepend
```

local-as

Description

This command specifies an alternate AS for this BGP process when interacting with the specified peers in a peer-group. With no modifiers, the specified local-as is prepended to the received AS_PATH when receiving routing updates from the peer, and prepended to the outgoing AS_PATH (after the process local AS) when transmitting local routes to the peer.

If the no-prepend CLI option is specified, then the supplied local-as is not prepended to the received AS_PATH.

If the replace-as CLI option is specified, then only the supplied local-as is prepended to the AS_PATH when transmitting local-route updates to this peer.

Note that replace-as can only be specified if no-prepend is.

This command is only allowed for eBGP peers.

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
local-as <asnum> { [ no-prepend [ replace-as ] ] }
no local-as
```

Parameters

Name	Description	Type
asnum	1-4294967295	Integer

Usage Guidelines

Use this command to configure local AS number for BGP neighbors in a peer-group and control how the local AS number is prepended to the AS_PATH of incoming and outgoing routes.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# local-as 65200 non-prepend
```

locator-led chassis off

Description

Disable Locator Chassis LED

Syntax

```
locator-led chassis off
```

locator-led chassis on

Description

Enable Locator Chassis LED

Syntax

```
locator-led chassis on
```

log-adjacency-changes

Description

Enables OSPFv2 adjacency state logs.

Parent Commands (Modes)

router ospf [vrf <vrf-name>]

Syntax

```
log-adjacency-changes [ detail ]
no log-adjacency-changes [ detail ]
```

Usage Guidelines

Use this command to enable OSPFv2 adjacency state logs.

Examples

```
sonic-clia(config-router-ospf)# log-adjacency-changes
sonic-clia(config-router-ospf)# log-adjacency-changes detail
```

Features this CLI belongs to

- OSPFv2 ## log-adjacency-changes ### Description

Enables OSPFv3 adjacency state logs.

Parent Commands (Modes)

router ospf6 [vrf <vrf-name>]

Syntax

```
log-adjacency-changes [ detail ]
no log-adjacency-changes [ detail ]
```

Usage Guidelines

Use this command to enable OSPFv3 adjacency state logs.

Examples

```
sonic-cli(config-router-ospf6)# log-adjacency-changes
sonic-cli(config-router-ospf6)# log-adjacency-changes detail
```

Features this CLI belongs to

- OSPFv3 ## log-neighbor-changes ##### Description

```
This command enables logging of neighbor's state transition events
```

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
log-neighbor-changes
no log-neighbor-changes
```

Usage Guidelines

```
Use this command to enable logging of neighbor UP/Down events along with
reason code for down event.
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# log-neighbor-changes
```

logger

Description

```
Enter messages into the system log
```

Syntax

```
logger <message>
```

Parameters

Name	Description	Type
message	String	String

logging console

Description

Configures the device to log messages to the console session.

Parent Commands (Modes)

configure terminal

Syntax

```
logging console <log_severity>
no logging console
```

Parameters

Name	Description	Type
log_severity		Integer

logging level

Description

Configure log level of the messages

Parent Commands (Modes)

configure terminal

Syntax

```
logging level <log_facility> <log_severity>
no logging level [ <log_facility> ]
```

Parameters

Name	Description	Type
log_facility	SYSLOG Log level facility	Select [kernel user-level mail system security4 syslog lpd nntp uucp time security10 ftpd ntpd logaudit logalert clock local0 local1 local2 local3 local4 local5 local6 local7]
log_severity		Integer

logging monitor

Description

Enables the device to log messages to the monitor. The configuration applies to Telnet and SSH sessions.

Parent Commands (Modes)

configure terminal

Syntax

```
logging monitor <log_severity>
no logging monitor
```

Parameters

Name	Description	Type
log_severity		Integer

logging profile

Description

Configure logging profile to apply filtering

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
logging profile <log_profile>
```

Parameters

Name	Description	Type
log_profile	String	String

Examples

```
sonic(config)# logging profile default
sonic(config)#
```

logging sai-level

Description

```
SAI component logging severity level setting
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
logging sai-level <component_name> <loglevel_value>
```

Parameters

Name	Description	Type
component_name	String	String
loglevel_value	String	String

logging server

Description

Configure remote syslog server to forward syslog messages

Parent Commands (Modes)

configure terminal

Syntax

```
logging server <host> [ message-type <msgtype> ] [ remote-port <vrport> ] [ source-interface {  
    Ethernet | Loopback | Management | PortChannel | Vlan | SubInterface } ] [ vrf { mgmt |  
    <vrf-name> } ] [ facility <facility_val> ] [ severity <severity_val> ] [ template {  
    <template_name> <template_pattern> } ]  
no logging server <host> { [ remote-port ] | [ source-interface ] | [ vrf ] }
```

Parameters

Name	Description	Type
host	WORD	String
msgtype	message type	Select [log(log) event(event)]
vrport	port	Integer
vrf-name	VRF name (prefixed by Vrf, Max: 15 characters)	String
facility_val	SYSLOG Log level facility	Select [kernel user-level mail system security4 syslog lpd nntp uucp time security10 ftpd ntpd logaudit logalert clock local0 local1 local2 local3 local4 local5 local6 local7] Select [emergency alert critical error warning notice info debug]
severity_val		
template_name	String	String
template_pattern	String(DOUBLE-QUOTED)	String

Examples

```
sonic(config)# logging server 20.1.1.1 source-interface Ethernet 2 vrf Vrf1  
sonic(config)#
```

logging sonic-level

Description

SONiC component logging severity level setting

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
logging sonic-level <component_name> <loglevel_value>
```

Parameters

Name	Description	Type
component_name	String	String
loglevel_value	String	String

logging template

Description

```
Local logging template configuration command
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
logging template <template_name> <template_pattern>
no logging template
```

Parameters

Name	Description	Type
template_name	String	String
template_pattern	String(DOUBLE-QUOTED)	String

mab

Description

Enable Mac-based authentication Bypass(MAB)on a specified port.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
mab [ auth-type { chap | eap-md5 | pap } ]  
no mab
```

mab

Description

Enable Mac-based authentication Bypass(MAB)on a specified port.

Parent Commands (Modes)

interface range iface_range_num

Syntax

```
mab [ auth-type { chap | eap-md5 | pap } ]  
no mab
```

mab request format attribute 1 groupsize

Description

This command sets configuration parameters that are used to format attribute1 for MAB requests to the RADIUS server.

RADIUS attribute 1 is the username, which is often the client MAC address.

Parent Commands (Modes)

configure terminal

Syntax

```
mab request format attribute 1 groupsize <groupsize> { separator { <separator_type> { [ lowercase ] | [ uppercase ] ] } } }
```

Parameters

Name	Description	Type
groupsize	Select [1(1) 2(2) 4(4) 12(12)]	
separator_type	Select [-(-) :(:) .(.)]	

Usage Guidelines

By using this command user can set the configuration parameters that are used to format attribute1 for MAB requests to the RADIUS server.

Examples

```
sonic(config)# mab request format attribute 1 groupsize {1 | 2 | 4 | 12} separator {- | : | .}
[lowercase | uppercase]
```

mab timeout server-timeout

Description

Time period after which specify the number of seconds the access device sends a RADIUS Access-Request packet to the authentication server. If no response is received when this timer expires, the MAB authentication fails.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
mab timeout server-timeout <time-period>
no mab timeout server-timeout
```

Parameters

Name	Description	Type
time-period		Integer

Usage Guidelines

This command allows configuring server timeout.

Examples

```
sonic-cli(conf-if-Eth1/1/1)# mab timeout server-timeout
```

mab timeout server-timeout

Description

Time period after which specify the number of seconds the access device sends a RADIUS Access-Request packet to the authentication server. If no response is received when this timer expires, the MAB authentication fails.

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
mab timeout server-timeout <time-period>
no mab timeout server-timeout
```

Parameters

Name	Description	Type
time-period		Integer

Usage Guidelines

This command allows configuring server timeout.

Examples

```
sonic-cli(conf-if-Eth1/1/1)# mab timeout server-timeout
```

mac access-group

Description

Apply MAC ACL to an interface.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
mac access-group <access-list-name> { in | out }
no mac access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	WORD	String

Usage Guidelines

ACL must be created first and must be of type MAC to be applied. Only 1 ACL of a given type can be applied per interface and per direction.

Examples

```
sonic(conf-if-Vlan100)# mac access-group macacl-example in
```

mac access-group

Description

Apply MAC ACL to an interface.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
mac access-group <access-list-name> { in | out }
no mac access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	WORD	String

Usage Guidelines

ACL must be created first and must be of type MAC to be applied. Only 1 ACL of a given type can be applied per interface and per direction.

Examples

```
sonic(conf-if-Vlan100)# mac access-group macacl-example in
```

mac access-group

Description

Apply MAC ACL to an interface.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
mac access-group <access-list-name> { in | out }
no mac access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	WORD	String

Usage Guidelines

ACL must be created first and must be of type MAC to be applied. Only 1 ACL of a given type can be applied per interface and per direction.

Examples

```
sonic(conf-if-Vlan100)# mac access-group macacl-example in
```

mac access-group

Description

Apply MAC ACL to an interface.

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
mac access-group <access-list-name> { in | out }
no mac access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	WORD	String

Usage Guidelines

ACL must be created first and must be of type MAC to be applied. Only 1 ACL of a given type can be applied per interface and per direction.

Examples

```
sonic(conf-if-Vlan100)# mac access-group macacl-example in
```

mac access-group

Description

Apply MAC ACL globally.

Parent Commands (Modes)

configure terminal

Syntax

```
mac access-group <access-list-name> { in | out }
no mac access-group <access-list-name> { in | out }
```

Parameters

Name	Description	Type
access-list-name	String(Max: 72 characters)	String

Usage Guidelines

ACL must be created first and must be of type MAC to be applied. Only 1 ACL of a given type can be applied globally per direction.

Examples

```
sonic(config)# mac access-group macacl-example in
```

mac access-list

Description

Create MAC ACL

Parent Commands (Modes)

configure terminal

Syntax

```
mac access-list <access-list-name>
no mac access-list <access-list-name>
```

Parameters

Name	Description	Type
access-list-name	String(Max: 72 characters)	String

Usage Guidelines

ACL name can be of maximum 63 characters. The name must begin with A-Z, a-z or 0-9. Underscore and hyphens can be used except as the first character. ACL name must be unique across all ACL types.

Examples

```
sonic(config)# mac access-list macacl-example
```

mac address-table

Description

```
MAC address-table configure commands
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
mac address-table <mac-address> { vlan { <phy-if-name> | PortChannel } }
no mac address-table <mac-address> vlan
```

Parameters

Name	Description	Type
mac-address	nn:nn:nn:nn:nn:nn	String
phy-if-name	EthernetNUM	

mac address-table aging-time

Description

MAC aging time

Parent Commands (Modes)

configure terminal

Syntax

```
mac address-table aging-time <mac-time>
no mac address-table aging-time
```

Parameters

Name	Description	Type
mac-time		Integer

mac address-table dampening-interval

Description

MAC move dampening threshold interval

Parent Commands (Modes)

configure terminal

Syntax

```
mac address-table dampening-interval <interval-value>
no mac address-table dampening-interval
```

Parameters

Name	Description	Type
interval-value		Integer

mac address-table dampening-threshold

Description

```
MAC move dampening threshold
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
mac address-table dampening-threshold <threshold-value>
no mac address-table dampening-threshold
```

Parameters

Name	Description	Type
threshold-value		Integer

mac move-policy

Description

```
Configure mac move policy
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
mac move-policy [ mac-move-threshold { <mac-move-threshold-val> { [ mac-detect-intvl
    <mac-detect-intvl-val> ] } } ] [ port-move-threshold { <port-move-threshold-val> { [
        port-detect-intvl <port-detect-intvl-val> ] } } ] [ hold-intvl <hold-intvl-val> ] [ action
    { port-shutdown | port-learn-disable | vlan-member-remove | log } ]
no mac move-policy { [ mac-move-threshold ] | [ port-move-threshold ] | [ action ] }
```

Parameters

Name	Description	Type
mac-move-threshold-val		Integer
mac-detect-intvl-val		Integer
port-move-threshold-val		Integer
port-detect-intvl-val		Integer
hold-intvl-val		Integer

mac-address

Description

```
Set SVI mac address
```

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
mac-address <mac-addr>
no mac-address
```

Parameters

Name	Description	Type
mac-addr	nn:nn:nn:nn:nn:nn	String

map

Description

Command to configure VNI-VLAN mappings and VNI-VRF mappings

Parent Commands (Modes)

interface vxlan <vxlan-if-name>

Syntax

```
map vni { <vnid> { { vlan { <vid> { [ count [ <numvid> ] ] } } } | { vrf <vrf-name> } } }  
no map vni { <vnid> { { vlan <vid> { [ count [ <numvid> ] ] } } | { vrf <vrf-name> } } }
```

Parameters

Name	Description	Type
vnid		Integer
vid		Integer
numvid		Integer
vrf-name	WORD	String

Usage Guidelines

```
(conf-if-vxlan-vtep)# map vni VNID vlan VLANID count COUNT  
(conf-if-vxlan-vtep)# map vni VNID vrf VRFNAME  
VNID - VNI value between 1 to 16777215  
VLANID - VLAN value between 1 to 4094  
COUNT - number of mappings (optional parameter)  
VRFNAME - string
```

Examples

```
sonic(config)# interface vxlan vtep1  
sonic(conf-if-vxlan-vtep1)# map vni 100 vlan 100 count 2  
sonic(conf-if-vxlan-vtep1)# map vni 100 vrf vrf1
```

match access-group

Description

Configures class-map match ACL

Parent Commands (Modes)

```
class-map <fbcs-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
match access-group { mac | ip | ipv6 } <access-list-name>
no match access-group
```

Parameters

Name	Description	Type
access-list-name	WORD	String

Examples

```
sonic(config)# class-map class_ip_acl match-type acl
sonic(config-class-map)# match access-group ip ip_acl1
```

match as-path

Description

```
Set routing policy match criteria as-path
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match as-path <as-path-name>
no match as-path
```

Parameters

Name	Description	Type
as-path-name	WORD	String

match community

Description

```
Set routing policy match criteria to BGP community
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match community <community-name>
no match community
```

Parameters

Name	Description	Type
community-name	WORD	String

match dei

Description

```
Configures class-map match packet criteria DEI
```

Parent Commands (Modes)

```
class-map <fbs-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
match dei <dei-val>
no match dei
```

Parameters

Name	Description	Type
dei-val	0-1	Integer

Examples

```
sonic(config)# class-map class1_fields match-type fields match-all
sonic(config-class-map)# match dei 0
```

match destination-address

Description

Configures class-map match packet destination-address

Parent Commands (Modes)

```
class-map <fbm-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
match destination-address { { mac { { <destination-mac-addr> <destination-mac-mask> } | { destination-mac-host <destination-mac-addr> } } } | { ip { <destination-ip-prefix> | { destination-ip-host <destination-ip> } } } | { ipv6 { <destination-ip-prefix> | { destination-ip-host <destination-ip> } } } }
no match destination-address { mac | ip | ipv6 }
```

Parameters

Name	Description	Type
destination-mac-addr	MACADDRESS	String
destination-mac-mask	MACADDRESS	String
destination-ip-prefix	A.B.C.D/mask	String
destination-ip	A.B.C.D	String

Examples

```
sonic(config)# class-map class1_fields match-type fields match-all
sonic(config-class-map)# match destination-address ip 2.1.1.1/32
```

match dscp

Description

Configures class-map match dscp

Parent Commands (Modes)

```
class-map <fb-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
match dscp { default | cs1 | cs2 | cs3 | cs4 | cs5 | cs6 | cs7 | af11 | af12 | af13 | af21 |
    af22 | af23 | af31 | af32 | af33 | af41 | af42 | af43 | ef | voice-admit | <dscp-val> }
no match dscp
```

Parameters

Name	Description	Type
dscp-val		Integer

Examples

```
sonic(config)# class-map class1_fields match-type fields match-all
sonic(config-class-map)# match dscp cs1
```

match ethertype

Description

Configures class-map match packet criteria ethertype

Parent Commands (Modes)

```
class-map <fb-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
match ethertype { ethertype-ip | ethertype-ipv6 | ethertype-arp | <ETHERTYPE> }
no match ethertype
```

Parameters

Name	Description	Type
ETHERTYPE	0x600-0xffff	String

Examples

```
sonic(config)# class-map class1_fields match-type fields match-all
sonic(config-class-map)# match ethertype ip
```

match evpn

Description

```
Set routing policy match criteria to BGP Ethernet Virtual Private Network
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match evpn { default-route | { route-type { macip | multicast | prefix } } | { vni <vni-number> } }
no match evpn { default-route | { route-type { macip | multicast | prefix } } | { vni <vni-number> } }
```

Parameters

Name	Description	Type
vni-number		Integer

match ext-community

Description

```
Set routing policy match criteria to BGP extended community
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match ext-community <community-name>
no match ext-community
```

Parameters

Name	Description	Type
community-name	WORD	String

match interface

Description

```
Set routing policy match criteria to interface
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match interface { <phy-if-name> | <Loopback> | PortChannel | { Vlan <vlan-id> } }
no match interface
```

Parameters

Name	Description	Type
phy-if-name	EthernetX.Y	
Loopback		
vlan-id		Integer

match ip address prefix-list

Description

```
Set routing policy match criteria to IPv4 prefix-list
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match ip address prefix-list <prefix-list-name>
no match ip address prefix-list
```

Parameters

Name	Description	Type
prefix-list-name	WORD	String

match ip next-hop prefix-list

Description

```
Set routing policy match criteria to next-hop prefix-list
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match ip next-hop prefix-list <match-hop>
no match ip next-hop prefix-list
```

Parameters

Name	Description	Type
match-hop	WORD	String

match ip protocol

Description

Updates class-map match ip attributes

Parent Commands (Modes)

```
class-map <fb-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
match ip protocol { <ip-protocol-val> | icmp | icmpv6 | tcp | udp }
no match ip protocol
```

Parameters

Name	Description	Type
ip-protocol-val		Integer

Examples

```
sonic(config)# class-map class1_fields match-type fields match-all
sonic(config-class-map)# match ip protocol tcp
```

match ipv6 address prefix-list

Description

Set routing policy match criteria to IPv6 prefix-list

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match ipv6 address prefix-list <prefix-list-name>
no match ipv6 address prefix-list
```

Parameters

Name	Description	Type
prefix-list-name	WORD	String

match l4-port

Description

```
Configures class-map match 14 source port
```

Parent Commands (Modes)

```
class-map <fbcs-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
match 14-port { source | destination } { { eq <eq-port-val> } | { range <begin-port-val> <end-port-val> } }  
no match 14-port { source | destination }
```

Parameters

Name	Description	Type
eq-port-val		Integer
begin-port-val		Integer
end-port-val		Integer

Usage Guidelines

```
Match on source port is allowed only when IP protocol is set to TCP or UDP
```

Examples

```
sonic(config)# class-map class1_fields match-type fields match-all  
sonic(config-class-map)# match 14-port source eq 10
```

match local-preference

Description

```
Set routing policy match criteria to local-preference
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match local-preference <match-loc>
no match local-preference
```

Parameters

Name	Description	Type
match-loc	(0-4294967295)	Integer

match metric

Description

```
Set routing policy match criteria to metric
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match metric <match-met>
no match metric
```

Parameters

Name	Description	Type
match-met	(0-4294967295)	Integer

match origin

Description

Specify BGP origin

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match origin { egp | igp | incomplete }
no match origin
```

match pcp

Description

Configures class-map match packet criteria pcp

Parent Commands (Modes)

```
class-map <fbs-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
match pcp { pcp-be | pcp-bk | pcp-ee | pcp-ca | pcp-vi | pcp-vo | pcp-ic | pcp-nc | { <pcp-val>
    { [ pcp-mask <pcp-val-mask> ] } } }
no match pcp
```

Parameters

Name	Description	Type
pcp-val	0-7	Integer
pcp-val-mask	0-7	Integer

Examples

```
sonic(config)# class-map class1_fields match-type fields match-all
sonic(config-class-map)# match pcp vi
```

match peer

Description

```
Set routing policy match criteria to peer IP
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match peer { <match-peer> | <phy-if-name> | PortChannel | { Vlan <vlan-id> } }
no match peer
```

Parameters

Name	Description	Type
match-peer	A.B.C.D/A::B	String
phy-if-name	EthernetX.Y	
vlan-id		Integer

match protocol

Description

```
Matches protocol trap to class-map
```

Parent Commands (Modes)

```
class-map <fbs-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
match protocol <trap-id>
no match protocol <trap-id>
```

Parameters

Name	Description	Type
trap-id	WORD	String

match source-address

Description

```
Configures class-map match packet source-address
```

Parent Commands (Modes)

```
class-map <fb-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
match source-address { { mac { { <source-mac-addr> <source-mac-mask> } | { source-mac-host
    <source-mac-addr> } } } | { ip { <source-ip-prefix> | { source-ip-host <source-ip> } } } |
    { ipv6 { <source-ip-prefix> | { source-ip-host <source-ip> } } } }
no match source-address { mac | ip | ipv6 }
```

Parameters

Name	Description	Type
source-mac-addr	MACADDRESS	String
source-mac-mask	MACADDRESS	String
source-ip-prefix	A.B.C.D/mask	String
source-ip	A.B.C.D	String

Examples

```
sonic(config)# class-map class1_fields match-type fields match-all
sonic(config-class-map)# match source-address ip 1.1.1.1/32
```

match source-protocol

Description

Specify source protocol

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match source-protocol { bgp | ospf | static | connected }
no match source-protocol
```

match source-vrf

Description

Source VRF

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match source-vrf <src-vrf>
no match source-vrf
```

Parameters

Name	Description	Type
src-vrf	WORD	String

match tag

Description

Redistributes routes in the routing table that match the specified tags.

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
match tag <match-tag>
no match tag
```

Parameters

Name	Description	Type
match-tag	1-4294967295	Integer

match tcp-flags

Description

```
Configures class-map match TCP Flags
```

Parent Commands (Modes)

```
class-map <fb-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
match tcp-flags { [ fin ] | [ not-fin ] } ] { [ syn ] | [ not-syn ] } ] { [ rst ] | [ not-rst ]
} ] { [ psh ] | [ not-psh ] } ] { [ ack ] | [ not-ack ] } ] { [ urg ] | [ not-urg ] } ]
no match tcp-flags { [ fin ] | [ not-fin ] } ] { [ syn ] | [ not-syn ] } ] { [ rst ] | [
not-rst ] } ] { [ psh ] | [ not-psh ] } ] { [ ack ] | [ not-ack ] } ] { [ urg ] | [ not-urg
] } ]
```

Usage Guidelines

Match on TCP flags is allowed only when IP protocol is set to TCP. not-xxx keyword can be used to match the corresponding flag set to 0

Examples

```
sonic(config)# class-map class1_fields match-type fields match-all
sonic(config-class-map)# match tcp-flags urg
```

match vlan

Description

Configures class-map match packet criteria vlan ID

Parent Commands (Modes)

```
class-map <fb-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
match vlan <vlan-val>
no match vlan
```

Parameters

Name	Description	Type
vlan-val	<1..4094>	String

Examples

```
sonic(config)# class-map class1_fields match-type fields match-all
sonic(config-class-map)# match vlan 200
```

max-med

Description

This command instructs BGP to advertise routes with max MED value under a given condition

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
max-med { { on-startup { <stime> [ <maxmedval> ] } } | { administrative [ <maxmedval> ] } }
no max-med { { on-startup <stime> } | administrative } [ <maxmedval> ]
```

Parameters

Name	Description	Type
stime		Integer
maxmedval	(0-4294967295)	Integer

Usage Guidelines

Use this command to instruct BGP to advertise routes with max MED value. The command allows user to set the condition under which routes with max MED value will be sent. One is during the startup for a prespecified number of seconds. The other is permanently. User can also specify the value for max MED.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# max-med on-startup 300 2000
```

max-metric

Description

Enables infinite metric advertising in OSPFv2 LSAs.

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
max-metric { [ router-lsa { [ administrative ] | { [ on-startup <onstartuptimervalue> ] } | { [ all [ <maxmetricvalue> ] ] } | { [ include-stub [ <maxmetricvalue> ] ] } | { [ external-lsa { { [ all [ <maxmetricvalue> ] ] } | { [ connected [ <maxmetricvalue> ] ] } ] } { [ all [ <maxmetricvalue> ] ] } { [ include-stub [ <maxmetricvalue> ] ] } ] } ] }
no max-metric { router-lsa { [ administrative ] | [ on-startup ] | [ all ] | [ include-stub ] | { external-lsa { [ all ] | [ connected ] } ] } }
```

Parameters

Name	Description	Type
onstartuptimervalue		Integer
maxmetricvalue		Integer

Usage Guidelines

Use this command to enable infinite metric advertising in OSPFv2 LSAs. Max-metric can be enabled administratively or during OSPFv2 startup time. When enabled administratively, max-metric advertizing will be effective till it is unconfigures explicitly. When enabled for restart time, max-metric will be advertised for a period of time specified in the config.

Examples

```
sonic-cli(config-router-ospf)# max-metric router-lsa administrative  
sonic-cli(config-router-ospf)# max-metric router-lsa on-startup 90
```

Features this CLI belongs to

- OSPFv2 ## maximum-paths ### Description

Sets the maximum number of equal cost multipath for eBGP.

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
maximum-paths <paths>  
no maximum-paths
```

Parameters

Name	Description	Type
paths		Integer

Usage Guidelines

Use this command to configure BGP to control the maximum number of equal cost multipath routes to eBGP destinations. This command is per address-family

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# maximum-paths 32
```

maximum-paths

Description

Forward packets over multiple paths

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
maximum-paths <paths>
no maximum-paths
```

Parameters

Name	Description	Type
paths		Integer

maximum-paths

Description

Max number of multiple paths for ECMP support

Parent Commands (Modes)

```
router ospf6 [ vrf <vrf-name> ]
```

Syntax

```
maximum-paths <maximum-paths-num>
no maximum-paths
```

Parameters

Name	Description	Type
maximum-paths-num		Integer

Examples

```
sonic-cli(config-router-ospf6)# maximum-paths
```

Features this CLI belongs to

- OSPFv3 ## maximum-paths ibgp ##### Description

```
Sets the maximum number of equal cost multipath for iBGP.
```

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
maximum-paths ibgp <ipaths> [ equal-cluster-length ]
no maximum-paths ibgp
```

Parameters

Name	Description	Type
ipaths		Integer

Usage Guidelines

Use this command to configure BGP to control the maximum number of equal cost multipath routes to iBGP destinations. This command is per address-family

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# maximum-paths ibgp 32
```

maximum-paths ibgp

Description

IBGP-multipath

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
maximum-paths ibgp <ipaths> [ equal-cluster-length ]
no maximum-paths ibgp
```

Parameters

Name	Description	Type
ipaths		Integer

maximum-prefix

Description

This command configures the maximum number of prefix to accept from this BGP neighbor.

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
maximum-prefix <max-prefix-val> { [ <threshold-val> ] { [ warning-only ] | { [ restart <interval> ] } ] } }  
no maximum-prefix [ <max-prefix-val> { [ threshold-val ] | [ warning-only ] | { [ restart <val> ] } ] } ]
```

Parameters

Name	Description	Type
max-prefix-val	1-4294967295	Integer
threshold-val		Integer
interval		Integer

Usage Guidelines

Use this command to set the upper limit on number of BGP prefixes to accept from this neighbor. This command has optional parameters for warning user when a threshold is reached and restarting BGP neighborship when maximum prefix limit has exceeded.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 20.20.20.2  
sonic(config-router-bgp-neighbor)# remote-as 300  
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast  
sonic(config-router-bgp-neighbor-af)# maximum-prefix 2000 80 warning-only
```

maximum-prefix

Description

This command configures the maximum number of prefix to accept from BGP neighbors in a peer-group.

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
maximum-prefix <max-prefix-val> { [ <threshold-val> ] { [ warning-only ] | { [ restart
    <interval> ] } ] } }
no maximum-prefix [ <max-prefix-val> { [ threshold-val ] | [ warning-only ] | { [ restart <val>
    ] } ] } ]
```

Parameters

Name	Description	Type
max-prefix-val	1-4294967295	Integer
threshold-val		Integer
interval		Integer

Usage Guidelines

Use this command to set the upper limit on number of BGP prefixes to accept from neighbors in a peer-group. This command has optional parameters for warning user when a threshold is reached and restarting BGP neighborship when maximum prefix limit has exceeded.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Int
sonic(config-router-bgp Pg)# address-family ipv4 unicast
sonic(config-router-bgp Pg Af)# maximum-prefix 2000 80 warning-only
```

maximum-prefix

Description

Maximum number of prefixes to accept from this peer

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
maximum-prefix <max-prefix-val> { [ <threshold-val> ] { [ warning-only ] | { [ restart
    <interval> ] } ] } }
no maximum-prefix [ <max-prefix-val> { [ threshold-val ] | [ warning-only ] | { [ restart <val>
    ] } ] } ]
```

Parameters

Name	Description	Type
max-prefix-val	1-4294967295	Integer
threshold-val		Integer
interval		Integer

maximum-prefix

Description

Maximum number of prefixes to accept from this peer

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
maximum-prefix <max-prefix-val> { [ <threshold-val> ] { [ warning-only ] | { [ restart
    <interval> ] } ] } }
no maximum-prefix [ <max-prefix-val> { [ threshold-val ] | [ warning-only ] | { [ restart <val>
    ] } ] } ]
```

Parameters

Name	Description	Type
max-prefix-val	1-4294967295	Integer
threshold-val		Integer
interval		Integer

mclag

Description

Configure MLAG interface

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
mclag <domain_id>
no mclag <domain_id>
```

Parameters

Name	Description	Type
domain_id		Integer

mclag domain

Description

Enter MLAG domain configuration mode

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
mclag domain <mclag-domain-id>
no mclag domain <mclag-domain-id>
```

Parameters

Name	Description	Type
mclag-domain-id		Integer

mclag gateway-mac

Description

Delete Gateway Mac for MCLAG.

Parent Commands (Modes)

configure terminal

Syntax

```
mclag gateway-mac <gw_mac>
no mclag gateway-mac <gw_mac>
```

Parameters

Name	Description	Type
gw_mac	nn:nn:nn:nn:nn:nn	String

mclag-peer-gateway

Description

Configure MCLAG Peer Gateway on Vlan interface

Parent Commands (Modes)

interface <vlan-if-name>

Syntax

```
mclag-peer-gateway
no mclag-peer-gateway
```

Usage Guidelines

```
sonic-clt(conf-if-Vlan10)# mclag-peer-gateway
```

Examples

```
sonic-cli(config)# interface Vlan 10
sonic-cli(conf-if-Vlan10)# mclag-peer-gateway
```

mclag-peer-gateway

Description

```
Configure MLAG Peer Gateway on Vlan interface
```

Parent Commands (Modes)

```
interface range create vlan_range_num
interface range vlan_range_num
```

Syntax

```
mclag-peer-gateway
no mclag-peer-gateway
```

mclag-separate-ip

Description

```
Configure separate IP on Vlan interface for L3 protocol support over MCLAG
```

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
mclag-separate-ip
no mclag-separate-ip
```

Usage Guidelines

```
sonic-cli(conf-if-Vlan10)# mclag-separate-ip
```

Examples

```
sonic-cli(config)# interface Vlan 10
sonic-cli(conf-if-Vlan10)# mclag-separate-ip
```

mclag-separate-ip

Description

Configure separate IP on Vlan interface for L3 protocol support over MCLAG

Parent Commands (Modes)

```
interface range create vlan_range_num  
interface range vlan_range_num
```

Syntax

```
mclag-separate-ip  
no mclag-separate-ip
```

mclag-system-mac

Description

Configures MCLAG system MAC address

Parent Commands (Modes)

```
mclag domain <mclag-domain-id>
```

Syntax

```
mclag-system-mac <MSM>  
no mclag-system-mac
```

Parameters

Name	Description	Type
MSM	nn:nn:nn:nn:nn:nn	String

Usage Guidelines

Use this command to set MCLAG system mac which will be used for MCLAG interface mac

Examples

```
sonic-cli(config-mclag-domain-100)#mclag-system-mac 00:bb:bb:bb:cc:cc
```

min-links

Description

```
Configure min-links for the portchannel
```

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
min-links <min_links>
no min-links
```

Parameters

Name	Description	Type
min_links		Integer

minimum-ttl

Description

```
Configure minimum TTL value expected in incoming BFD control packet for an multihop peer.
```

Parent Commands (Modes)

```
peer <peer_ipv4>
peer <peer_ipv6>
peer [ interface ] <interfacename>
peer [ local-address ] <local_ipv4>
peer [ local-address ] <local_ipv6>
peer [ multihop ]
peer [ vrf ] <vrffname>
```

Syntax

```
minimum-ttl <ttl-val>
no minimum-ttl
```

Parameters

Name	Description	Type
ttl-val		Integer

Usage Guidelines

Default value is 254.

Examples

```
device()#configure terminal
device(config)#bfd
device(config-bfd)# peer 192.168.0.5 interface Ethernet0
device(config-bfd-peer)# minimum-ttl 250
```

minimum-ttl

Description

Configure minimum TTL value expected in incoming BFD control packet for an multihop peer.

Parent Commands (Modes)

```
profile <profilename>
```

Syntax

```
minimum-ttl <ttl-val>
no minimum-ttl
```

Parameters

Name	Description	Type
ttl-val		Integer

Usage Guidelines

Default value is 254.

Examples

```
device()#configure terminal
device(config)#bfd
device(conf-bfd)# profile fast
device(conf-bfd-profile)# minimum-ttl 250
```

mirror

Description

Add mirror to dropcounter

Parent Commands (Modes)

dropcounters <counter-name>

Syntax

```
mirror <mirror-str>
no mirror
```

Parameters

Name	Description	Type
mirror-str	String	String

mirror-session

Description

Mirror session configuration

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
mirror-session <session-name>
no mirror-session <session-name>
```

Parameters

Name	Description	Type
session-name	String(Max: 24 characters)	String

mtu

Description

```
Configure MTU
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
mtu <mtu>
no mtu
```

Parameters

Name	Description	Type
mtu		Integer

mtu

Description

Configure MTU

Parent Commands (Modes)

interface <vlan-if-name>

Syntax

```
mtu <mtu>
no mtu
```

Parameters

Name	Description	Type
mtu		Integer

mtu

Description

Configure MTU

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
mtu <mtu>
no mtu
```

Parameters

Name	Description	Type
mtu		Integer

mtu

Description

Configure MTU

Parent Commands (Modes)

interface Management <mgmt-if-id>

Syntax

```
mtu <mtu>
no mtu
```

Parameters

Name	Description	Type
mtu		Integer

mtu

Description

Configure interface link MTU

Parent Commands (Modes)

interface range iface_range_num

Syntax

```
mtu <mtu>
no mtu
```

Parameters

Name	Description	Type
mtu		Integer

mtu

Description

Configure interface link MTU

Parent Commands (Modes)

```
interface range create vlan_range_num
interface range vlan_range_num
```

Syntax

```
mtu <mtu>
no mtu
```

Parameters

Name	Description	Type
mtu		Integer

mtu

Description

Configure interface link MTU

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
interface range po_range_num
```

Syntax

```
mtu <mtu>
no mtu
```

Parameters

Name	Description	Type
mtu		Integer

mtu

Description

```
Configures mtu for routed subinterface
```

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
mtu <mtu>
no mtu
```

Parameters

Name	Description	Type
mtu		Integer

Usage Guidelines

```
[no] mtu <mtu-value>
```

Examples

```
sonic-cl# configure terminal
sonic-cl(config)# interface Ethernet 0.10
sonic-cl(conf-subif-Ethernet0.10)# mtu 2000
sonic-cl(config)#+
```

```
sonic-cli# configure terminal
sonic-cli(config)# interface Ethernet 0.10
sonic-cli(conf-subif-Ethernet0.10)# no mtu
sonic-cli(config)#
```

Features this CLI belongs to

- Subinterface ## nat

Description

Enter NAT configuration

Parent Commands (Modes)

configure terminal

Syntax

nat

nat-zone

Description

NAT Zone

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
nat-zone <zone>
no nat-zone
```

Parameters

Name	Description	Type
zone		Integer

nat-zone

Description

```
NAT Zone
```

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
nat-zone <zone>
no nat-zone
```

Parameters

Name	Description	Type
zone		Integer

nat-zone

Description

```
NAT Zone
```

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
nat-zone <zone>
no nat-zone
```

Parameters

Name	Description	Type
zone		Integer

nat-zone

Description

NAT Zone

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
nat-zone <zone>
no nat-zone
```

Parameters

Name	Description	Type
zone		Integer

neigh-suppress

Description

Enable ARP and ND Suppression

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
neigh-suppress
no neigh-suppress
```

neighbor

Description

This command creates a BGP neighbor

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }  
no neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Parameters

Name	Description	Type
ip	A.B.C.D/A::B	String

Usage Guidelines

Use this command to create a BGP neighbor. User can input neighbor's IPv4/IPv6 address directly or can input an interface name for unnumbered BGP neighbor.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300  
sonic(config-router-bgp)# neighbor 30.30.30.3  
sonic(config-router-bgp-neighbor)#[
```

neighbor-route

Description

Configure neighbor route creation on Vlan interface.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
neighbor-route  
no neighbor-route
```

Usage Guidelines

```
sonic-cli(conf-if-Vlan10)# neighbor-route
```

Examples

```
sonic-cli(config)# interface Vlan 10  
sonic-cli(config-Vlan10)# neighbor-route
```

network

Description

```
This command enables user to add a network to announce via BGP
```

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
network <prefix> { [ backdoor ] { [ route-map <route-map-name> ] } }  
no network <prefix> { [ backdoor ] { [ route-map <route-map-name> ] } }
```

Parameters

Name	Description	Type
prefix	A.B.C.D/mask	String
route-map-name	WORD	String

Usage Guidelines

This command can be used by users and network administrators to statically inject routes into BGP. User can use route-map optional parameter to modify/set the various attributes of the route.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# network 10.10.0.0/16
```

network

Description

Enable routing on an IP network

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
network <prefix> { [ route-map <route-map-name> ] }
no network <prefix> { [ route-map <route-map-name> ] }
```

Parameters

Name	Description	Type
prefix	A::B/mask	String
route-map-name	WORD	String

network

Description

Configures networks in an area.

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
network <ipaddrmask> { area <areaid> }
no network <ipaddrmask> { area <areaid> }
```

Parameters

Name	Description	Type
ipaddrmask	A.B.C.D/mask	String
areaid	A.B.C.D or 0..4294967295	String

Usage Guidelines

Use this command to configure or associate network addresses into specific areas. Interfaces belonging to these network addresses will be considered as part of the area specified. This config command is mutually exclusive with interface mode area command within a VRF. When network command is used in a VRF, area command cannot be used at interface mode configuration.

Examples

```
sonic-cli(config-router-ospf)# network 10.1.1.0/24 area 0
sonic-cli(config-router-ospf)# network 19.1.1.0/16 area 19
```

Features this CLI belongs to

- OSPFv2 ## network import-check ### Description

This command instructs BGP to check if BGP network route exists in local route table before advertising the network

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
network import-check
no network import-check
```

Usage Guidelines

By default, BGP networks are advertised to neighbors irrespective of whether the same route exists in local route table or not. This behavior may lead to data traffic blackholing. User can use this command to put a restriction on BGP networks to get advertised only if a corresponding route from IGP exists in local route table.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# network import-check
```

network-policy

Description

Configure network policy profile for an interface.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
network-policy <number>
no network-policy
```

Parameters

Name	Description	Type
number		Integer

Usage Guidelines

Use this command to configure the network policy profile for an interface.

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# network-policy 1
```

Features this CLI belongs to

- LLDP-MED ## network-policy ### Description

Configure a network policy profile.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
network-policy profile <np_num>
no network-policy profile <np_num>
```

Parameters

Name	Description	Type
np_num		Integer

Usage Guidelines

This command allows to configure a network policy profile.

Examples

```
sonic-cli(config)# network-policy profile 1
```

Features this CLI belongs to

- LLDP-MED ## next-hop-self ### Description

This command sets next-hop attribute as it's own address in the outbound route updates to this BGP neighbor

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
next-hop-self [ force ]
no next-hop-self [ force ]
```

Usage Guidelines

Use this command to disable BGP next-hop attribute computation and override the next-hop by sender's own address.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# next-hop-self
```

next-hop-self

Description

This command sets next-hop attribute as it's own address in the outbound route updates to BGP neighbors in a peer-group

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
next-hop-self [ force ]
no next-hop-self [ force ]
```

Usage Guidelines

Use this command to disable BGP next-hop attribute computation and override the next-hop by sender's own address.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp-pg)# address-family ipv4 unicast
sonic(config-router-bgp-af)# next-hop-self
```

next-hop-self

Description

This command sets next-hop attribute as it's own address in the outbound route updates to this BGP neighbor

Parent Commands (Modes)

address-family l2vpn evpn

Syntax

```
next-hop-self [ force ]
no next-hop-self [ force ]
```

Usage Guidelines

Use this command to disable BGP next-hop attribute computation and override the next-hop by sender's own address.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)# next-hop-self
```

next-hop-self

Description

This command sets next-hop attribute as it's own address in the outbound route updates to BGP neighbors in a peer-group

Parent Commands (Modes)

address-family l2vpn evpn

Syntax

```
next-hop-self [ force ]
no next-hop-self [ force ]
```

Usage Guidelines

Use this command to disable BGP next-hop attribute computation and override the next-hop by sender's own address.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# address-family l2vpn evpn
sonic(config-router-bgp Pg-af)# next-hop-self
```

next-hop-self

Description

Disable the next hop calculation for this neighbor

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
next-hop-self [ force ]
no next-hop-self [ force ]
```

next-hop-self

Description

Disable the next hop calculation for this neighbor

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
next-hop-self [ force ]
no next-hop-self [ force ]
```

no

Description

This command to unconfigure WRED minimum, maximum thresholds and drop probability for color green or yellow.

Parent Commands (Modes)

```
qos wred-policy <name>
```

Syntax

```
no { green | yellow }
```

Usage Guidelines

Use this command to unconfigure WRED minimum, maximum and drop probability for green or yellow color packets.

Examples

```
sonic(conf-wred-wred-green)# no color green  
sonic(conf-wred-wred-yellow)# no color yellow
```

ntp master

Description

Initialize a NTP server in configuration mode.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ntp master [ stratum <stratum-number> ]  
no ntp master
```

Parameters

Name	Description	Type
stratum-number		Integer

Usage Guidelines

```
sonic(config)# ntp master NTP-SERVER-STRATUM
NTP-SERVER-STRATUM:          0-15                                Stratum number
```

Examples

```
sonic# configure
sonic(config)# ntp master 7
```

ntp server

Description

Configure a NTP server in configuration mode. NTP sever can be identified by IPv4, IPv6 address or host name.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ntp server { <ip> | <server-name> } [ minpoll { <min-poll> { maxpoll <max-poll> } } ] [ prefer
    <isprefer> ]
no ntp server { <ip> | <server-name> }
```

Parameters

Name	Description	Type
ip	A.B.C.D/A::B	String
server-name	WORD	String
min-poll	seconds	Integer
max-poll	seconds	Integer
isprefer	Boolean choice	Select [true(1) false(0)]

Usage Guidelines

```
sonic(config)# ntp server NTP-SERVER-IP
sonic(config)# ntp server NTP-SERVER-HOST-NAME
NTP-SERVER-IP:          A.B.C.D/A::B
NTP-SERVER-HOST-NAME: WORD (Max: 253 characters)
```

IPv4/IPv6 address of NTP server
Hostname of NTP server

Examples

```
sonic# configure
sonic(config)# ntp server 10.11.0.1 key 128
sonic(config)# ntp server pool.ntp.org
```

ntp source-interface

Description

Configure NTP source interface in configuration mode. NTP source interface can be an ethernet, loopback, management, port channel or vlan interface.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ntp source-interface { Ethernet | Loopback | Management | PortChannel | Vlan }
no ntp source-interface { Ethernet | Loopback | Management | PortChannel | Vlan }
```

Usage Guidelines

```
sonic(config)# ntp source-interface INTERFACE
INTERFACE: Ethernet      Ethernet interface or sub-interface
           Loopback       Loopback interface
           Management     Management Interface
           PortChannel    PortChannel interface or sub-interface
           Vlan           Vlan interface
```

Examples

```
sonic# configure
sonic(config)# ntp source-interface Ethernet 8
sonic(config)# ntp source-interface Ethernet 8.100
sonic(config)# ntp source-interface Management 0
sonic(config)# ntp source-interface Loopback 88
sonic(config)# ntp source-interface PortChannel 18
sonic(config)# ntp source-interface PortChannel 18.200
sonic(config)# ntp source-interface Vlan 888
```

ntp vrf

Description

Enable NTP on VRF in configuration mode. NTP can be enabled on management VRF or default VRF.

Parent Commands (Modes)

configure terminal

Syntax

```
ntp vrf { mgmt | default }
no ntp vrf
```

Usage Guidelines

```
sonic(config)# ntp vrf VRF
VRF: mgmt      Enable NTP on management VRF
      default   Enable NTP on default VRF
```

Examples

```
sonic# configure
sonic(config)# ntp vrf mgmt
sonic(config)# ntp vrf default
```

ospf

Description

Configures OSPFv2 router parameters.

Parent Commands (Modes)

router ospf [vrf <vrf-name>]

Syntax

```
ospf
no ospf
```

Usage Guidelines

Use this command to configure OSPFv2 router parameters like router-id and ABR type.

Examples

Features this CLI belongs to

- OSPFv2 ## ospf abr-type ### Description

Configures OSPFv2 router ABR type.

Parent Commands (Modes)

router ospf [vrf <vrf-name>]

Syntax

```
ospf abr-type { cisco | ibm | shortcut | standard }
no ospf abr-type
```

Usage Guidelines

Use this command to configure OSPFv2 router ABR type.

Examples

```
sonic-cli(config-router-ospf)# ospf abr-type shortcut
sonic-cli(config-router-ospf)# ospf abr-type cisco
```

Features this CLI belongs to

- OSPFv2 ## ospf router-id ### Description

Configures OSPFv2 router ID.

Parent Commands (Modes)

router ospf [vrf <vrf-name>]

Syntax

```
ospf router-id <routerid>
no ospf router-id
```

Parameters

Name	Description	Type
routerid	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 router ID. At the time of OSPFv2 router startup or new configuration, if the router-id configuration is not present then the highest IPv4 address among the available IPv4 interfaces will be considered as router-id. Upon configuring a specific router-id, newly configured router-id will be used. Upon unconfiguring a router-id, same router-id will continue to be used till next router-id config or system restart or OSPFv2 router unconfigure.

Examples

```
sonic-cli(config-router-ospf)# ospf router-id 19.1.1.1
```

Features this CLI belongs to

- OSPFv2 ## ospf6 ### Description

Configures OSPFv3 router parameters.

Parent Commands (Modes)

```
router ospf6 [ vrf <vrf-name> ]
```

Syntax

```
ospf6  
no ospf6
```

Usage Guidelines

Use this command to configure OSPFv3 router parameters like router-id.

Examples

Features this CLI belongs to

- OSPFv3 ## ospf6 router-id ### Description

Configures OSPFv3 router ID.

Parent Commands (Modes)

```
router ospf6 [ vrf <vrf-name> ]
```

Syntax

```
ospf6 router-id <routerid>
no ospf6 router-id
```

Parameters

Name	Description	Type
routerid	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv3 router ID. Upon configuring a specific router-id, newly configured router-id will be used. Upon unconfiguring a router-id, same router-id will continue to be used till next router-id config or system restart or OSPFv3 router unconfigure.

Examples

```
sonic-cli(config-router-ospf6)# ospf6 router-id 10.11.1.1
```

Features this CLI belongs to

- OSPFv3 ## override-capability ### Description

This command instructs BGP to override the result of Capability Negotiation with local configuration. Ignore remote peer's capability value.

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
override-capability  
no override-capability
```

Usage Guidelines

Use this command to ignored the negotiated capability parameters with neighbor and instead use the locally configured parameters.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 30.30.30.3  
sonic(config-router-bgp-neighbor)# override-capability
```

override-capability

Description

This command instructs BGP to override the result of Capability Negotiation with local configuration. Ignore remote peer's capability value.

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
override-capability  
no override-capability
```

Usage Guidelines

Use this command to ignored the negotiated capability parameters with neighbors in a peer-group and instead use the locally configured parameters.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Ext  
sonic(config-router-bgp-Pg)# override-capability
```

passive

Description

This command makes BGP neighbor passive. That is, this BGP neighbor will not initiate a session. However, it will listen to any incoming BGP session.

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
passive  
no passive
```

Usage Guidelines

Use this command to set a BGP neighbor as passive.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 30.30.30.3  
sonic(config-router-bgp-neighbor)# passive
```

passive

Description

This command makes BGP neighbors in a peer-group passive. That is, BGP neighbors will not initiate a session. However, it will listen to any incoming BGP session.

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
passive  
no passive
```

Usage Guidelines

Use this command to set BGP neighbors in a peer-group as passive.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# passive
```

passive-interface

Description

Configures OSPFv2 passive interfaces.

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
passive-interface { default | { <interfacename> { { [ <ipaddr> [ non-passive ] ] } | [ non-passive ] } } }
no passive-interface { default | { <interfacename> { [ <ipaddr> ] } } }
```

Parameters

Name	Description	Type
interfacename	Interface Type - Ranges	
ipaddr	A.B.C.D	String

Usage Guidelines

Use this command to configure OSPFv2 passive interfaces. Default mode makes all interfaces as passive interface. When default mode is enabled, user will be able to activate chosen interfaces by enabling non-passive mode of passive interface configuration.

Examples

```
sonic-cli(config-router-ospf)# passive-interface Ethernet0
    or
sonic-cli(config-router-ospf)# passive-interface default
sonic-cli(config-router-ospf)# passive-interface Ethernet0 non-passive
```

Features this CLI belongs to

- OSPFv2 ## passive-mode ### Description

Configure Bidirectional Forwarding detection(BFD) peer as passive, session creation will not be initiated by this peer.

Parent Commands (Modes)

```
peer <peer_ipv4>
peer <peer_ipv6>
peer [ interface ] <interfacename>
peer [ local-address ] <local_ipv4>
peer [ local-address ] <local_ipv6>
peer [ multihop ]
peer [ vrf ] <vrfname>
```

Syntax

```
passive-mode
no passive-mode
```

Usage Guidelines

By default passive mode is disabled for BFD peer.

Examples

```
device()#configure terminal
device(config)#bfd
device(conf-bfd)# peer 192.168.0.5 interface Ethernet0
device(conf-bfd-peer)# passive-mode
```

passive-mode

Description

Configure Bidirectional Forwarding detection(BFD) peer as passive, session creation will not be initiated by this peer.

Parent Commands (Modes)

```
profile <filename>
```

Syntax

```
passive-mode  
no passive-mode
```

Usage Guidelines

By default passive mode is disabled for BFD peer.

Examples

```
device()#configure terminal  
device(config)#bfd  
device(conf-bfd)# profile fast  
device(conf-bfd-profile)# passive-mode
```

password

Description

This command sets a MD5 password to be used with the tcp socket that is being used to connect to the remote peer.

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
password <String> [ encrypted ]  
no password
```

Parameters

Name	Description	Type
String	String	String

Usage Guidelines

This command is for security purposes. When Password is configured for a BGP neighbor, sender will include a 16-bytes MD5 digest in TCP header of BGP message and the receiver should be able to validate the digest then only accept the BGP message.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# password jackandjillwentupthehill
```

password

Description

This command sets a MD5 password to be used with the tcp socket that is being used to connect to the remote peers in a peer-group.

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
password <String> [ encrypted ]
no password
```

Parameters

Name	Description	Type
String	String	String

Usage Guidelines

This command is for security purposes. When Password is configured for BGP neighbors in a peer-group, sender will include a 16-bytes MD5 digest in TCP header of BGP message and the receiver should be able to validate the digest then only accept the BGP message.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp-pg)# password ilovebeansbecausetheyaremean
```

path

Description

This command is used to configure the sensor path of an already existing sensor group.

Parent Commands (Modes)

```
sensor-group <sg-id>
```

Syntax

```
path <path_val> depth <depth_val>
```

Parameters

Name	Description	Type
path_val	Enter the string URL in one of the following formats:	String
depth_val		Integer

Usage Guidelines

Use this command is used to configure the sensor path of an already existing sensor group (depth parameter present for compatibility, not used).

Examples

```
sonic(config-telemetry)# sensor-group sg1
sonic(conf-tm-sensor-sg1)# path platform/cpu depth 0
sonic(conf-tm-sensor-sg1)#{
```

path-target

Description

This command is used to set the data source for the current subscription.

Parent Commands (Modes)

subscription <sub-id>

Syntax

path-target <path_target_val>

Parameters

Name	Description	Type
path_target_val	WORD	String

Usage Guidelines

Use this command to set the data source for the current subscription.

Examples

```
sonic(config-telemetry)# subscription s1
sonic(conf-tm-sub-s1)# path-target OC_YANG
```

pbf next-hop-group

Description

Creates a policy based forwarding next-hop group

Parent Commands (Modes)

configure terminal

Syntax

```
pbf next-hop-group <fbn-nhgrp-name> [ type { ip | ipv6 } ]
no pbf next-hop-group <fbn-nhgrp-name>
```

Parameters

Name	Description	Type
fbs-nhgrp-name	WORD	String

Usage Guidelines

A next-hop-group name can be of maximum 63 characters. The name must begin with A-Z, a-z or 0-9. Underscore and hypens can be used except as the first character. A next-hop-group can be of type ip or ipv6. The group type can not be updated after its created. The group type is mandatory at the time of creating the group.

Examples

```
sonic(config)# pbf next-hop-group ipv4-group-1 type ip  
sonic-clia(config-pbf-ip-nh-group)#[
```

pbf replication-group

Description

Creates a policy based forwarding replication group

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
pbf replication-group <fbs-replgrp-name> [ type { ip | ipv6 } ]  
no pbf replication-group <fbs-replgrp-name>
```

Parameters

Name	Description	Type
fbs-replgrp-name	WORD	String

Usage Guidelines

A replication-group name can be of maximum 63 characters. The name must begin with A-Z, a-z or 0-9. Underscore and hypens can be used except as the first character. A replication-group can be of type ip or ipv6. The group type can not be updated after its created. The group type is mandatory at the time of creating the group.

Examples

```
sonic(config)# pbf replication-group ipv4-group-1 type ip  
sonic-cli(config-pbf-ip-repl-group)#
```

pbs

Description

Peak Burst Size in bytes

Syntax

```
pbs <pbs>  
no pbs
```

Parameters

Name	Description	Type
pbs		Integer

peer

Description

Configure single-hop and multi-hop Bidirectional Forwarding detection(BFD) peer.

Parent Commands (Modes)

```
bfd
```

Syntax

```
peer { <peer_ipv4> | <peer_ipv6> } [ vrf <vriname> ] [ multihop ] [ local-address {  
    <local_ipv4> | <local_ipv6> } ] [ interface <interfacename> ]  
no peer { <peer_ipv4> | <peer_ipv6> } [ vrf <vriname> ] [ multihop ] [ local-address {  
    <local_ipv4> | <local_ipv6> } ] [ interface <interfacename> ]
```

Parameters

Name	Description	Type
peer_ipv4	A.B.C.D	String
peer_ipv6	A::B	String
vrfname	String	String
local_ipv4	A.B.C.D	String
local_ipv6	A::B	String
interfacename	Interface Type - Ranges	

Usage Guidelines

For single-hop BFD peer interface must be configured and for multi-hop BFD peer local address must be configured.

For link-local BFD Peer address, it is mandatory to configure link-local local address.

Examples

```
device()#configure terminal  
device(config)#bfd  
device(conf-bfd)# peer 192.168.0.5 interface Ethernet0
```

```
device()#configure terminal  
device(config)#bfd  
device(conf-bfd)# peer 192.168.0.5 interface Ethernet0 vrf Vrf1
```

```
device()#configure terminal  
device(config)#bfd  
device(conf-bfd)# peer 192.168.0.2 multihop local-address 192.168.0.3
```

```
device()#configure terminal  
device(config)#bfd  
device(conf-bfd)# peer 192.168.0.2 multihop local-address 192.168.0.3 vrf Vrf1
```

peer-group

Description

This command creates a BGP peer-group

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
peer-group <template-str>
no peer-group <template>
```

Parameters

Name	Description	Type
template-str	WORD	String

Usage Guidelines

Use this command to create a BGP peer-group.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp-pg)#[
```

peer-group

Description

This command assigns a BGP neighbor to a peer-group

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
peer-group <template-name>
no peer-group [ <template-name> ]
```

Parameters

Name	Description	Type
template-name	WORD	String

Usage Guidelines

Assigning a BGP neighbor to a peer-group will inherit parameters from peer-group.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# peer-group PG_Ext
```

peer-ip

Description

Configures MLAG session's peer IP address

Parent Commands (Modes)

```
mclag domain <mclag-domain-id>
```

Syntax

```
peer-ip <PIP>
no peer-ip
```

Parameters

Name	Description	Type
PIP	A.B.C.D	String

Usage Guidelines

Use this command to configure/change MLAG session's peer ip address

Examples

```
sonic-cli(config-mclag-domain-100)#peer-ip 10.1.1.2
```

peer-link

Description

Configures MLAG peer interface

Parent Commands (Modes)

```
mclag domain <mclag-domain-id>
```

Syntax

```
peer-link { <eth-if> | <po-if> }
no peer-link
```

Parameters

Name	Description	Type
eth-if	EthernetNUM	
po-if	PortChannelNUM	

Usage Guidelines

Use this command to configure/change MLAG session's peer link

Examples

```
sonic-cli(config-mclag-domain-100)#peer-link Eth 1/12/1
```

pfc-priority

Description

This command to add PFC Priority to queue entry in map.

Parent Commands (Modes)

```
qos map pfc-priority-queue <name>
```

Syntax

```
pfc-priority <dot1p_list> { queue <qid> }  
no pfc-priority <dot1p_list>
```

Parameters

Name	Description	Type
dot1p_list		String
qid		Integer

Usage Guidelines

Use this command to add entry to map PFC Priority to queue.

Examples

```
sonic# configure terminal  
sonic(config)# dot1p 1 queue 0  
sonic(config)# dot1p 2 queue 0  
sonic(config)# dot1p 3 queue 1
```

ping

Description

Send ICMP ECHO_REQUEST to network hosts

Syntax

```
ping [ source <source_dev_ip> ]
```

Parameters

Name	Description	Type
source_dev_ip	WORD	String

ping vrf

Description

```
Select VRF instance
```

Syntax

```
ping vrf <vrf-name> [ source <source_dev_ip> ]
```

Parameters

Name	Description	Type
vrf-name	VRF name (prefixed by Vrf, Max: 15 characters)	String
source_dev_ip	WORD	String

ping vrf mgmt

Description

```
Ping using management VRF
```

Syntax

```
ping vrf mgmt [ source <source_dev_ip> ]
```

Parameters

Name	Description	Type
source_dev_ip	WORD	String

ping6

Description

Send ICMPv6 ECHO_REQUEST to network hosts

Syntax

```
ping6 [ source <source_dev_ip> ]
```

Parameters

Name	Description	Type
source_dev_ip	WORD	String

ping6 vrf

Description

Select VRF instance

Syntax

```
ping6 vrf <vrf-name> [ source <source_dev_ip> ]
```

Parameters

Name	Description	Type
vrf-name	VRF name (prefixed by Vrf, Max: 15 characters)	String
source_dev_ip	WORD	String

ping6 vrf mgmt

Description

Ping6 using management VRF

Syntax

```
ping6 vrf mgmt [ source <source_dev_ip> ]
```

Parameters

Name	Description	Type
source_dev_ip	WORD	String

pir

Description

Peak Information Rate in Kbps

Syntax

```
pir <pir>
no pir
```

Parameters

Name	Description	Type
pir		Integer

police

Description

Configures policer action for qos Flow

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
police { { cir <cir-value> { [ cbs <cbs-value> ] } { [ pir <pir-value> ] } { [ pbs <pbs-value> ] } } | { cbs <cbs-value> { [ pir <pir-value> ] } { [ pbs <pbs-value> ] } } | { pir <pir-value> { [ pbs <pbs-value> ] } } | { pbs <pbs-value> } }  
no police [ cir ] [ cbs ] [ pir ] [ pbs ]
```

Parameters

Name	Description	Type
cir-value		String
cbs-value		String
pir-value		String
pbs-value		String

Usage Guidelines

CIR: Committed information rate in bits per second. CIR is mandatory. The value can be optionally suffixed with kbps(1000), mbps(1000000), gbps (1000000000) or tbps (1000000000000).cir 300000000 cbs 300000000 pir 300000000 pbs 300000000.

CBS: Committed burst size in bytes. The value can be suffixed with KB(1000), MB(1000000), GB(1000000000) or TB(1000000000000). The default value is 20% of the CIR in bytes. If configured by the user, it must be greater than or equal to CIR in bytes.

PIR: Peak information rate in bits per second. The value can be optionally suffixed with kbps(1000), mbps(1000000), gbps (1000000000) or tbps (1000000000000). If configured by the user, it must be greater than CIR

PBS: Peak burst size. The value can be suffixed with KB(1000), MB(1000000), GB(1000000000) or TB(1000000000000). The default value is 20% of the PIR value in bytes. If configured by the user, it must be greater than PIR value in bytes and also CBS value

If only CIR is configured, then its 1 rate, 2 color policer. Any traffic exceeding CIR value will be marked as red and will be dropped.

If both CIR and PIR is configured, then is 2 rate 3 color policer. Any traffic that exceeds CIR but less than PIR will be marked as yellow. Any traffic that is more than PIR will be marked as red and will be dropped

Examples

```
sonic(config)# policy-map policy_qos type qos  
sonic(config-policy-map)# class class_permit_ip priority 10  
sonic(config-policy-map-flow)#cir 300000000 cbs 300000000 pir 300000000 pbs 300000000
```

police

Description

Configures policer action for acl-copp Flow

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
police { { cir <cir-value> { [ cbs <cbs-value> ] } { [ pir <pir-value> ] } { [ pbs <pbs-value> ] } } | { cbs <cbs-value> { [ pir <pir-value> ] } { [ pbs <pbs-value> ] } } | { pir <pir-value> { [ pbs <pbs-value> ] } } | { pbs <pbs-value> } }  
no police [ cir ] [ cbs ] [ pir ] [ pbs ]
```

Parameters

Name	Description	Type
cir-value		String
cbs-value		String
pir-value		String
pbs-value		String

Usage Guidelines

CIR: Committed information rate in bits per second. CIR is mandatory. The value can be optionally suffixed with kbps(1000), mbps(1000000), gbps (1000000000) or tbps (1000000000000).cir 300000000 cbs 300000000 pir 300000000 pbs 300000000.

CBS: Committed burst size in bytes. The value can be suffixed with KB(1000), MB(1000000), GB(1000000000) or TB(1000000000000). The default value is 20% of the CIR in bytes. If configured by the user, it must be greater than or equal to CIR in bytes.

PIR: Peak information rate in bits per second. The value can be optionally suffixed with kbps(1000), mbps(1000000), gbps (1000000000) or tbps (1000000000000). If configured by the user, it must be greater than CIR.

PBS: Peak burst size. The value can be suffixed with KB(1000), MB(1000000), GB(1000000000) or TB(1000000000000). The default value is 20% of the PIR value in bytes. If configured by the user, it must be greater than PIR value in bytes and also CBS value.

If only CIR is configured, then its 1 rate, 2 color policer. Any traffic exceeding CIR value will be marked as red and will be dropped.

If both CIR and PIR is configured, then is 2 rate 3 color policer. Any traffic that exceeds CIR but less than PIR will be marked as yellow. Any traffic that is more than PIR will be marked as red and will be dropped.

Examples

```
sonic(config)# policy-map policy_acl-copp type acl-copp
sonic(config-policy-map)# class class_permit_ip priority 10
sonic(config-policy-map-flow)#cir 300000000 cbs 300000000 pir 300000000 pbs 300000000
```

police

Description

Set rate limiting parameters

Parent Commands (Modes)

```
copp-action <copp-action-name>
```

Syntax

```
police { { cir <cir-value> { [ cbs <cbs-value> ] } } | { cbs <cbs-value> } }
no police [ cir ] [ cbs ]
```

Parameters

Name	Description	Type
cir-value		String
cbs-value		String

Examples

```
sonic(config-action)# police cir 6000 cbs 6000
sonic(config-action)#

```

```
sonic(config-action)# police mode sr_tcm red drop
sonic(config-action)#

```

policy-map

Description

Configures policy-map

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
policy-map <fbs-policy-name> type { qos | monitoring | forwarding | copp | acl-copp }  
no policy-map <fbs-policy-name>
```

Parameters

Name	Description	Type
fbs-policy-name	WORD	String

Usage Guidelines

```
Policy-map name can be of maximum 63 characters. The name must begin with A-Z, a-z or 0-9.  
Underscore and hypens can be used except as the first character
```

Examples

```
sonic(config)# policy-map policy_vrf type forwarding
```

pool

Description

```
Create NAT pool
```

Parent Commands (Modes)

```
nat
```

Syntax

```
pool <pool-name> <global-ip-range> [ <global-port-range> ]  
no pool <pool-name>
```

Parameters

Name	Description	Type
pool-name	String	String
global-ip-range	String	String
global-port-range	[1-65535]-[1-65535]	String

port

Description

This command specifies the UDP port on which a device listens for RADIUS requests from configured RADIUS clients.

Parent Commands (Modes)

aaa server radius dynamic-author

Syntax

```
port <portNumber>
no port
```

Parameters

Name	Description	Type
portNumber		Integer

Usage Guidelines

This command configures the UDP port on which a device listens for RADIUS requests from configured RADIUS clients.

The supported range for the port-number is 1025 to 65535.

Examples

```
sonic-cl(i(config)# aaa server radius dynamic-author
sonic-cl(i(config-radius-da)# port 8989
```

port

Description

This command sets TCP port for a BGP neighbor.

Parent Commands (Modes)

neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }

Syntax

```
port <tcpport>
no port
```

Parameters

Name	Description	Type
tcpport		Integer

Usage Guidelines

Use this command to set a specific port for BGP neighbor. This command is optional.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# port 61356
```

port

Description

This command sets TCP port for BGP neighbors in a peer-group.

Parent Commands (Modes)

peer-group <template-str>

Syntax

```
port <tcpport>
no port
```

Parameters

Name	Description	Type
tcpport		Integer

Usage Guidelines

Use this command to set a specific port for BGP neighbors in a peer-group. This command is optional.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# port 65001
```

port-group

Description

This command configure port speed for the member ports of a port-group. The port-group is not supported in all the hardware platforms.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
port-group <pg> { speed <port_speed> }
no port-group <pg> speed
```

Parameters

Name	Description	Type
pg port_speed	Port-group (id) Port speed	Select [10(10MBPS) 100(100MBPS) 1000(1GIGE) 2500(2.5GIGE) 5000(5GIGE) 10000(10GIGE) 20000(20GIGE) 25000(25GIGE) 40000(40GIGE) 50000(50GIGE) 100000(100GIGE) 400000(400GIGE)]

Usage Guidelines

Use this command to change the speed of the ports for the platform which supports port-group.
`port-group id speed speed-in-Mbps`

Examples

```
sonic# configure terminal
sonic(config)# port-group 1 speed 10000
```

port-number-base

Description

Configure port number base for the portchannel

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
port-number-base <numberbase>
no port-number-base
```

Parameters

Name	Description	Type
numberbase		Integer

port-security enable

Description

Enable PMS at interface level.

Parent Commands (Modes)

`interface <phy-if-name>`

Syntax

```
port-security enable  
no port-security enable
```

Usage Guidelines

Use this command to enable PMS at interface level

Examples

```
sonic-clia(config)# interface Ethernet 0  
sonic-clia(conf-if-Ethernet0)# port-security enable
```

port-security enable

Description

Enable PMS at interface level.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]
```

Syntax

```
port-security enable  
no port-security enable
```

Usage Guidelines

Use this command to enable PMS at interface level

Examples

```
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# port-security enable
```

port-security maximum

Description

Configures Maximum no of secure MAC's allowed on this interface

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
port-security maximum <maximum>
no port-security maximum
```

Parameters

Name	Description	Type
maximum		Integer

Usage Guidelines

Configures Maximum no of secure MAC's allowed on this interface

Examples

```
sonic-cli(config)# port-security maximum 3
```

port-security maximum

Description

Configures Maximum no of secure MAC's allowed on this interface

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
port-security maximum <maximum>  
no port-security maximum
```

Parameters

Name	Description	Type
maximum		Integer

Usage Guidelines

```
Configures Maximum no of secure MAC's allowed on this interface
```

Examples

```
sonic-cli(config)# port-security maximum 3
```

port-security violation

Description

```
Configures the action to be taken in the event of security violation
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
port-security violation <violation>  
no port-security violation
```

Parameters

Name	Description	Type
violation	Action to be taken in the event of violation	Select [protect]

Usage Guidelines

Configures the action to be taken in the event of security violation

Examples

```
sonic-clia(config)# port-security violation protect
```

port-security violation

Description

Configures the action to be taken in the event of security violation

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
port-security violation <violation>
no port-security violation
```

Parameters

Name	Description	Type
violation	Action to be taken in the event of violation	Select [protect]

Usage Guidelines

Configures the action to be taken in the event of security violation

Examples

```
sonic-cli(config)# port-security violation protect
```

portchannel graceful-shutdown

Description

Enable portchannel graceful shutdown

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
portchannel graceful-shutdown  
no portchannel graceful-shutdown
```

Usage Guidelines

Use this command to enable portchannel graceful shutdown

Examples

```
sonic-cli(config)# portchannel graceful-shutdown
```

preempt

Description

Configure preempt for IPv4 VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
preempt  
no preempt
```

Examples

```
sonic(config)# interface Ethernet4  
sonic(conf-if-Ethernet4)#  
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4  
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#preempt
```

preempt

Description

Configure preempt for IPv6 VRRP instance

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
preempt  
no preempt
```

Examples

```
sonic(config)# interface Ethernet4  
sonic(conf-if-Ethernet4)#  
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6  
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#preempt
```

preempt

Description

Configure preempt for IPv4 VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
preempt  
no preempt
```

Examples

```
sonic(config)# interface Ethernet4  
sonic(conf-if-Ethernet4)#  
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4  
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#preempt
```

preempt

Description

Configure preempt for IPv6 VRRP instance

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
preempt  
no preempt
```

Examples

```
sonic(config)# interface Ethernet4  
sonic(conf-if-Ethernet4)#  
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6  
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#preempt
```

preempt

Description

Configure preempt for IPv4 VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
preempt  
no preempt
```

Examples

```
sonic(config)# interface Ethernet4  
sonic(conf-if-Ethernet4)#  
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4  
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#preempt
```

preempt

Description

Configure preempt for IPv6 VRRP instance

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
preempt  
no preempt
```

Examples

```
sonic(config)# interface Ethernet4  
sonic(conf-if-Ethernet4)#  
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6  
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#preempt
```

preempt

Description

Configure preempt for IPv4 VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
preempt  
no preempt
```

Examples

```
sonic(config)# interface Ethernet4  
sonic(conf-if-Ethernet4)#  
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4  
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#preempt
```

preempt

Description

```
Configure preempt for IPv6 VRRP instance
```

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
preempt  
no preempt
```

Examples

```
sonic(config)# interface Ethernet4  
sonic(conf-if-Ethernet4)#  
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6  
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#preempt
```

prefix-list

Description

```
This command configures prefix list for a BGP neighbor
```

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
prefix-list <pname> { in | out }
no prefix-list <pname> { in | out }
```

Parameters

Name	Description	Type
pname	WORD	String

Usage Guidelines

Use this command to define policy (route filtering) for a BGP neighbor in outbound or/and inbound direction.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# prefix-list pl_allow_remote in
```

prefix-list

Description

This command configures prefix list for a BGP peer-group

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
prefix-list <pname> { in | out }
no prefix-list <pname> { in | out }
```

Parameters

Name	Description	Type
pname	WORD	String

Usage Guidelines

Use this command to define policy (route filtering) for a BGP peer-group in outbound or/and inbound direction.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# address-family ipv4 unicast
sonic(config-router-bgp Pg Af)# prefix-list pl_allow_remote in
```

prefix-list

Description

Filter updates to/from this neighbor

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
prefix-list <pname> { in | out }
no prefix-list <pname> { in | out }
```

Parameters

Name	Description	Type
pname	WORD	String

prefix-list

Description

Filter updates to/from this neighbor

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
prefix-list <pname> { in | out }
no prefix-list <pname> { in | out }
```

Parameters

Name	Description	Type
pname	WORD	String

primary-ip

Description

Command to set the primary IPv4 address

Parent Commands (Modes)

interface vxlan <vxlan-if-name>

Syntax

```
primary-ip <PIP>
no primary-ip
```

Parameters

Name	Description	Type
PIP	A.B.C.D	String

Usage Guidelines

```
(conf-if-vxlan-vtep)# primary-ip SOURCEIP
SOURCEIP - primary IPv4 address
```

Examples

```
sonic(config)# interface vxlan vtep1
sonic(conf-if-vxlan-vtep1)# primary-ip 1.1.1.2
```

priority

Description

Configure priority for IPv4 VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
priority <priority-value>
no priority
```

Parameters

Name	Description	Type
priority-value		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#priority 120
```

priority

Description

```
Configure priority for IPv6 VRRP instance
```

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
priority <priority-value>
no priority
```

Parameters

Name	Description	Type
priority-value		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#priority 120
```

priority

Description

```
Configure priority for IPv4 VRRP instance
```

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
priority <priority-value>
no priority
```

Parameters

Name	Description	Type
priority-value		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#priority 120
```

priority

Description

Configure priority for IPv6 VRRP instance

Parent Commands (Modes)

vrrp ipv6

Syntax

```
priority <priority-value>
no priority
```

Parameters

Name	Description	Type
priority-value		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#priority 120
```

priority

Description

```
Configure priority for IPv4 VRRP instance
```

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
priority <priority-value>
no priority
```

Parameters

Name	Description	Type
priority-value		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#priority 120
```

priority

Description

```
Configure priority for IPv6 VRRP instance
```

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
priority <priority-value>
no priority
```

Parameters

Name	Description	Type
priority-value		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#priority 120
```

priority

Description

Configure priority for IPv4 VRRP instance

Parent Commands (Modes)

vrrp ipv4

Syntax

```
priority <priority-value>
no priority
```

Parameters

Name	Description	Type
priority-value		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#priority 120
```

priority

Description

```
Configure priority for IPv6 VRRP instance
```

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
priority <priority-value>
no priority
```

Parameters

Name	Description	Type
priority-value		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#priority 120
```

priority-flow-control

Description

```
PFC Configuration
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
priority-flow-control { { priority <dot1p> } | asymmetric }
no priority-flow-control { { priority [ <dot1p> ] } | asymmetric }
```

Parameters

Name	Description	Type
dot1p		Integer

priority-flow-control watchdog action

Description

```
PFC watchdog storm action
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
priority-flow-control watchdog action { alert | drop | forward }  
no priority-flow-control watchdog action
```

priority-flow-control watchdog counter-poll

Description

```
Enable PFC watchdog FLEX counters
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
priority-flow-control watchdog counter-poll  
no priority-flow-control watchdog counter-poll
```

priority-flow-control watchdog detect-time

Description

```
PFC watchdog detection time
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
priority-flow-control watchdog detect-time <detection-time>
no priority-flow-control watchdog detect-time
```

Parameters

Name	Description	Type
detection-time		Integer

priority-flow-control watchdog off

Description

```
PFC watchdog disable
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
priority-flow-control watchdog off
```

priority-flow-control watchdog on

Description

```
PFC watchdog enable
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
priority-flow-control watchdog on
```

priority-flow-control watchdog polling-interval

Description

```
Configure watchdog
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
priority-flow-control watchdog polling-interval <interval>
no priority-flow-control watchdog polling-interval
```

Parameters

Name	Description	Type
interval		Integer

priority-flow-control watchdog restore-time

Description

```
PFC watchdog restoration time
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
priority-flow-control watchdog restore-time <restore-time>
no priority-flow-control watchdog restore-time
```

Parameters

Name	Description	Type
restore-time		Integer

profile

Description

Configure Bidirectional Forwarding detection(BFD) profile.

Parent Commands (Modes)

bfd

Syntax

```
profile <profilename>
no profile <profilename>
```

Parameters

Name	Description	Type
profilename	WORD	String

Usage Guidelines

BFD profile can be associated with BFD peer, all the configuration in BFD profile are applied to the BFD peer.

BFD profile can also be associated with protocol(BGP) config and the dynamic session created derives all the config from the corresponding BFD profile.

Examples

```
device()#configure terminal
device(config)#bfd
device(conf-bfd)# profile fast
```

profile

Description

Configure profile to be associated with the peer, all the configuration in the profile will be applied to the peer.

Parent Commands (Modes)

```
peer <peer_ipv4>
peer <peer_ipv6>
peer [ interface ] <interfacename>
peer [ local-address ] <local_ipv4>
peer [ local-address ] <local_ipv6>
peer [ multihop ]
peer [ vrf ] <vriname>
```

Syntax

```
profile <profilename>
no profile
```

Parameters

Name	Description	Type
profilename	WORD	String

Examples

```
device()#configure terminal
device(config)#bfd
device(conf-bfd)# peer 192.168.0.5 interface Ethernet0
device(conf-bfd-peer)# profile fast
```

proxy-arp

Description

Configure Proxy ARP on Vlan interface.

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
proxy-arp  
no proxy-arp
```

Usage Guidelines

```
sonic-cli(conf-if-Vlan10)# proxy-arp
```

Examples

```
sonic-cli(config)# interface Vlan 10  
sonic-cli(config)# proxy-arp
```

ptp announce-timeout

Description

```
Configure PTP announce receipt timeout value. The default value is 3.
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ptp announce-timeout <ptp_announce_timeout>
```

Parameters

Name	Description	Type
ptp_announce_timeout		Integer

Examples

```
sonic(config)# ptp announce-timeout 2  
Success
```

ptp domain

Description

```
Configure PTP domain
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ptp domain <ptp_domain>
```

Parameters

Name	Description	Type
ptp_domain		Integer

Examples

```
sonic(config)# ptp domain 1
Success
```

ptp domain-profile

Description

```
Configure PTP domain profile
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ptp domain-profile <ptp_domain_profile>
```

Parameters

Name	Description	Type
ptp_domain_profile	Select [default(ieee1588) g8275.1(G.8275.1) g8275.2(G.8275.2)]	

Examples

```
sonic(config)# ptp domain-profile default
Success
```

ptp ipv6-scope

Description

Configure PTP IPv6 multicast address scope

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ptp ipv6-scope <ptp_ipv6_scope>
```

Parameters

Name	Description	Type
ptp_ipv6_scope	Hexadecimal range (0x0 to 0xf)	String

Examples

```
sonic(config)# ptp ipv6-scope 0xe
Success
```

ptp log-announce-interval

Description

Configure PTP log announce interval value. The interval should be the same in the whole domain.
It's specified as a power of two in seconds. The default is 1 (2 seconds).

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ptp log-announce-interval <ptp_announce_interval>
```

Parameters

Name	Description	Type
ptp_announce_interval		Integer

Examples

```
sonic(config)# ptp log-announce-interval 0
Success
```

ptp log-min-delay-req-interval

Description

Configure PTP log min delay req interval value. It is specified as a power of two in seconds.
The default is 0 (1 second).

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ptp log-min-delay-req-interval <ptp_delay_request_interval>
```

Parameters

Name	Description	Type
ptp_delay_request_interval		Integer

Examples

```
sonic(config)# ptp log-min-delay-req-interval 0  
Success
```

ptp log-sync-interval

Description

Configure PTP log sync interval value. It is specified as a power of two in seconds. The default is 0 (1 second).

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ptp log-sync-interval <ptp_sync_interval>
```

Parameters

Name	Description	Type
ptp_sync_interval		Integer

Examples

```
sonic(config)# ptp log-sync-interval 0  
Success
```

ptp mode

Description

```
Configure PTP clock type
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ptp mode <mode_type>
```

Parameters

Name	Description	Type
mode_type	PTP mode	Select [boundary-clock(BC) peer-to-peer-transparent-clock(P2P_TC) end-to-end-transparent-clock(E2E_TC) disable(disable)]

Examples

```
sonic(config)# ptp mode boundary-clock
Success
```

ptp network-transport

Description

Configure PTP network-transport

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ptp network-transport <ptp_network_transport_type> <ptp_master_slave>
```

Parameters

Name	Description	Type
ptp_network_transport_type		Select [l2(L2) ipv4(UDPv4) ipv6(UDPV6)]
ptp_master_slave	unicast/multicast	Select [unicast(unicast) multicast(multicast)]

Examples

```
sonic(config)# ptp network-transport ipv4 unicast  
Success
```

ptp port add

Description

Add a PTP port

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ptp port add <Ethernet>
```

Parameters

Name	Description	Type
Ethernet	EthernetNUM	

Examples

```
sonic(config)# ptp port add Ethernet 64  
Success
```

ptp port del

Description

Delete a PTP port

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ptp port del <Ethernet>
```

Parameters

Name	Description	Type
Ethernet	EthernetNUM	

Examples

```
sonic(config)# ptp port del Ethernet 64
Success
```

ptp port master-table

Description

Add/Delete a master IP/MAC from the master table for the designated slave port

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ptp port master-table { <Ethernet> { { add { <l3_ip> | <mac> } } | { del { <l3_ip> | <mac> } } }
```

Parameters

Name	Description	Type
Ethernet	EthernetNUM	
l3_ip	A.B.C.D/A::B	String
mac	nn:nn:nn:nn:nn:nn	String

Examples

```
sonic(config)# ptp port master-table Ethernet 64 add 10.1.1.1
Success
sonic(config)# ptp port master-table Ethernet 64 del 10.1.1.1
Success
```

ptp priority1

Description

Configure PTP priority1 value

Parent Commands (Modes)

configure terminal

Syntax

```
ptp priority1 <ptp_priority1>
```

Parameters

Name	Description	Type
ptp_priority1		Integer

Examples

```
sonic(config)# ptp priority1 128
Success
```

ptp priority2

Description

Configure PTP priority2 value

Parent Commands (Modes)

configure terminal

Syntax

```
ptp priority2 <ptp_priority2>
```

Parameters

Name	Description	Type
ptp_priority2		Integer

Examples

```
sonic(config)# ptp priority2 128
Success
```

ptp two-step

Description

Configure PTP two-step mode

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ptp two-step <ptp_two_step>
```

Parameters

Name	Description	Type
ptp_two_step	enable or disable	Select [enable(enable) disable(disable)]

Examples

```
sonic(config)# ptp two-step enable
Success
```

qos map dot1p-tc

Description

This command creates map to associates set of DOT1P to Traffic classes.
This map used to assign a traffic class to data packets on the basis of the received packets DOT1P field.

Parent Commands (Modes)

configure terminal

Syntax

```
qos map dot1p-tc <name>
no qos map dot1p-tc <name>
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

Use this command to create DOT1P to Traffic class entires map.

Examples

```
sonic# configure terminal
sonic(config)#qos map dot1p-tc dot1p-map
```

qos map dscp-tc

Description

This command creates map to associates set of DSCP(Differentiated Services Code Point) to Traffic classes. This map used to assign a traffic class to data packets on the basis of the received packets DSCP field.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
qos map dscp-tc <name>
no qos map dscp-tc <name>
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

Use this command to create DSCP to Traffic class entires map.

Examples

```
sonic# configure terminal
sonic(config)# qos map dscp-tc dscp-map
```

qos map pfc-priority-queue

Description

This command creates map to associates set of PFC Priorities to Queues. This map is used to classify a queue for data packets on the basis of the received packets DOT1P field.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
qos map pfc-priority-queue <name>
no qos map pfc-priority-queue <name>
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

Use this command to create PFC Priority to Queue entires map.

Examples

```
sonic# configure terminal  
sonic(config)#qos map pfc-priority-queue pfc-priority-queue-map
```

qos map tc-dot1p

Description

This command creates map to associates set of TC(Traffic Class) to DOT1P(Vlan PCP value).

This map is used to Remark the DOT1P field in egress packets on the basis of the internal Traffic Class.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
qos map tc-dot1p <name>  
no qos map tc-dot1p <name>
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

Use this command to create Traffic class to DOT1P entries map.

Examples

```
sonic# configure terminal  
sonic(config)# qos map tc-dot1p tc_dot1p
```

qos map tc-dscp

Description

This command creates map to associates set of TC(Traffic Class) to DSCP(Differentiated Services Code Point).
This map is used to Remark the DSCP field in egress packets on the basis of the internal Traffic Class.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
qos map tc-dscp <name>  
no qos map tc-dscp <name>
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

Use this command to create Traffic class to DSCP entries map.

Examples

```
sonic# configure terminal  
sonic(config)# qos map tc-dscp tc_dscp
```

```
qos map tc-pg
```

Description

This command creates map to associates set of TC(Traffic Class) to queue. This map is used to assign a priority group to data packets on the basis of the traffic class.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
qos map tc-pg <name>  
no qos map tc-pg <name>
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

Use this command to create Traffic class to Priority-Group entires map.

Examples

```
sonic# configure terminal  
sonic(config)#qos map tc-pg tc-pg-map
```

```
qos map tc-queue
```

Description

This command creates map to associates set of TC(Traffic Class) to queue. This map is used to assign an egress queue to data packets on the basis of the traffic class.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
qos map tc-queue <name>
no qos map tc-queue <name>
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

Use this command to create Traffic class to Queue entires map.

Examples

```
sonic# configure terminal
sonic(config)# qos map tc-queue tc-queue-map
```

qos scheduler-policy

Description

Scheduler Policy Configuration

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
qos scheduler-policy <name>
no qos scheduler-policy <name>
```

Parameters

Name	Description	Type
name	String	String

qos wred-policy

Description

This command creates WRED policy

Parent Commands (Modes)

configure terminal

Syntax

```
qos wred-policy <name>
no qos wred-policy <name>
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

Use this command to create WRED policy.

Examples

```
sonic# configure terminal
sonic(config)# qos wred-policy wred-green
```

qos-map dot1p-tc

Description

DOT1P to TC map configuration

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
qos-map dot1p-tc <dot1p_tc_map_name>
no qos-map dot1p-tc
```

Parameters

Name	Description	Type
dot1p_tc_map_name	WORD	String

qos-map dot1p-tc

Description

DOT1P to TC map configuration

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [
  fallback ] [ fast_rate ]
```

Syntax

```
qos-map dot1p-tc <dot1p_tc_map_name>
no qos-map dot1p-tc
```

Parameters

Name	Description	Type
dot1p_tc_map_name	WORD	String

qos-map dscp-tc

Description

DSCP to TC map configuration

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
qos-map dscp-tc <dscp_tc_map_name>
no qos-map dscp-tc
```

Parameters

Name	Description	Type
dscp_tc_map_name	WORD	String

qos-map dscp-tc

Description

DSCP to TC map configuration

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
qos-map dscp-tc <dscp_tc_map_name>
no qos-map dscp-tc
```

Parameters

Name	Description	Type
dscp_tc_map_name	WORD	String

qos-map dscp-tc

Description

DSCP to TC map configuration

Parent Commands (Modes)

interface <vlan-if-name>

Syntax

```
qos-map dscp-tc <dscp_tc_map_name>
no qos-map dscp-tc
```

Parameters

Name	Description	Type
dscp_tc_map_name	WORD	String

qos-map dscp-tc

Description

DSCP to TC map configuration

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
qos-map dscp-tc <dscp_tc_map_name>
no qos-map dscp-tc
```

Parameters

Name	Description	Type
dscp_tc_map_name	WORD	String

qos-map pfc-priority-queue

Description

```
PFC Priority to Queue map configuration
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
qos-map pfc-priority-queue <pfc_priority_queue_map_name>
no qos-map pfc-priority-queue
```

Parameters

Name	Description	Type
pfc_priority_queue_map_name	WORD	String

qos-map tc-dot1p

Description

```
TC to DOT1P map configuration
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
qos-map tc-dot1p <tc_dot1p_map_name>
no qos-map tc-dot1p
```

Parameters

Name	Description	Type
tc_dot1p_map_name	WORD	String

qos-map tc-dot1p

Description

TC to DOT1P map configuration

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
qos-map tc-dot1p <tc_dot1p_map_name>
no qos-map tc-dot1p
```

Parameters

Name	Description	Type
tc_dot1p_map_name	WORD	String

qos-map tc-dscp

Description

TC to DSCP map configuration

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
qos-map tc-dscp <tc_dscp_map_name>
no qos-map tc-dscp
```

Parameters

Name	Description	Type
tc_dscp_map_name	WORD	String

qos-map tc-dscp

Description

```
TC to DSCP map configuration
```

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [
    fallback ] [ fast_rate ]
```

Syntax

```
qos-map tc-dscp <tc_dscp_map_name>
no qos-map tc-dscp
```

Parameters

Name	Description	Type
tc_dscp_map_name	WORD	String

qos-map tc-dscp

Description

TC to DSCP map configuration

Parent Commands (Modes)

interface <vlan-if-name>

Syntax

```
qos-map tc-dscp <tc_dscp_map_name>
no qos-map tc-dscp
```

Parameters

Name	Description	Type
tc_dscp_map_name	WORD	String

qos-map tc-dscp

Description

TC to DSCP map configuration

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
qos-map tc-dscp <tc_dscp_map_name>
no qos-map tc-dscp
```

Parameters

Name	Description	Type
tc_dscp_map_name	WORD	String

qos-map tc-pg

Description

TC to priority group map configuration

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
qos-map tc-pg <tc_pg_map_name>
no qos-map tc-pg
```

Parameters

Name	Description	Type
tc_pg_map_name	WORD	String

qos-map tc-queue

Description

TC to Queue map configuration

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
qos-map tc-queue <tc_queue_map_name>
no qos-map tc-queue
```

Parameters

Name	Description	Type
tc_queue_map_name	WORD	String

qos-mode

Description

Configure QoS Mode

Parent Commands (Modes)

```
interface vxlan <vxlan-if-name>
```

Syntax

```
qos-mode { uniform | { pipe { dscp <dscp-value> } } }
```

Parameters

Name	Description	Type
dscp-value		Integer

queue

Description

Queue configuration

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
queue <qid> { { wred-policy <wred_prof_name> } | { scheduler-policy <sp_name> } }
no queue <qid> { { wred-policy [ <wred_prof_name> ] } | { scheduler-policy [ <sp_name> ] } }
```

Parameters

Name	Description	Type
qid		Integer
wred_prof_name	WORD	String
sp_name	String	String

radius-server auth-type

Description

```
Configures global auth-type for RADIUS.
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
radius-server auth-type <auth_type>
no radius-server auth-type
```

Parameters

Name	Description	Type
auth_type	authentication type	Select [pap(pap) chap(chap) mschapv2(mschapv2)]

Examples

```
sonic(config)# radius-server auth-type chap
```

radius-server host

Description

```
Configures a server for RADIUS.
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
radius-server host <host> [ auth-port <vauth_port> ] [ auth-type <vauth_type> ] [ key { <vkey> [ encrypted ] } ] [ priority <vpriority> ] [ retransmit <vretransmit> ] [ source-interface { Ethernet | Loopback | Management | PortChannel | Vlan | SubInterface } ] [ timeout <vttimeout> ] [ vrf { mgmt | <vrf-name> } ]  
no radius-server host <host> { [ auth-port ] | [ auth-type ] | [ key ] | [ priority ] | [ retransmit ] | [ source-interface ] | [ timeout ] | [ vrf ] }
```

Parameters

Name	Description	Type
host	WORD	String
vauth_port	port	Integer
vauth_type	authentication type	Select [pap(pap) chap(chap) mschapv2(mschapv2)]
vkey	WORD	String
vpriority	(1..64)	Integer
vretransmit	attempts	Integer
vttimeout	seconds	Integer
vrf-name	VRF name (prefixed by Vrf, Max: 15 characters)	String

Examples

```
sonic(config)# radius-server host 10.59.100.2 key testing123
```

radius-server key

Description

```
Configures global shared secret for RADIUS.
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
radius-server key <key> [ encrypted ]  
no radius-server key
```

Parameters

Name	Description	Type
key	WORD	String

Examples

```
sonic(config)# radius-server key testing123
```

radius-server nas-ip

Description

Configures global NAS-IP|IPV6-Address (Type 4|95) attribute for RADIUS PDU.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
radius-server nas-ip <nas_ip>
no radius-server nas-ip
```

Parameters

Name	Description	Type
nas_ip	A.B.C.D/A::B	String

Examples

```
sonic(config)# radius-server nas-ip 10.59.100.2
```

radius-server retransmit

Description

Configures global timeout for RADIUS.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
radius-server retransmit <retransmit>
no radius-server retransmit
```

Parameters

Name	Description	Type
retransmit	attempts	Integer

Examples

```
sonic(config)# radius-server timeout 3
```

radius-server statistics

Description

Configures global statistics collection for RADIUS.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
radius-server statistics <enable>
```

Parameters

Name	Description	Type
enable	enable or disable	Select [enable(enable) disable(disable)]

Examples

```
sonic(config)# radius-server statistics enable
```

radius-server timeout

Description

Configures global timeout for RADIUS.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
radius-server timeout <timeout>
no radius-server timeout
```

Parameters

Name	Description	Type
timeout	seconds	Integer

Examples

```
sonic(config)# radius-server timeout 3
```

rd

Description

This command specifies the route-distinguisher to be attached to routes exported from current VRF into EVPN

Parent Commands (Modes)

```
vni <vninum>
```

Syntax

```
rd <rdvalue>
no rd <rdvalue>
```

Parameters

Name	Description	Type
rdvalue	A.B.C.D:NN or ASN:NN	String

Usage Guidelines

```
[no] rd {route-distinguisher}
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100 vrf Vrf1
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# vni 100
sonic(config-router-bgp-af-vni)# rd 11:11
```

rd

Description

This command specifies the route-distinguisher to be attached to routes exported from current VRF into EVPN

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
rd <rdvalue>
no rd <rdvalue>
```

Parameters

Name	Description	Type
rdvalue	A.B.C.D:NN or ASN:NN	String

Usage Guidelines

```
[no] rd {route-distinguisher}
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100 vrf Vrf1
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# rd 11:11
```

read-quanta

Description

This command configures the maximum number of BGP packets to read from peer socket in one cycle of I/O

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
read-quanta <rdval>
no read-quanta
```

Parameters

Name	Description	Type
rdval		Integer

Usage Guidelines

BGP packets are read off the wire one at a time in a loop. This setting controls how many iterations the loop runs for. It is best to leave this setting on the default.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# read-quanta 6
```

reboot

Description

```
reboot
```

Syntax

```
reboot [ force <_force> ] [ noprompt <_noprompt> ]
```

Parameters

Name	Description	Type
_force	[yes/no]	Select [yes no]
_noprompt	[yes/no]	Select [yes no]

receive-interval

Description

Configure packet receive interval for Bidirectional Forwarding detection(BFD) peer.

Parent Commands (Modes)

```
peer <peer_ipv4>
peer <peer_ipv6>
peer [ interface ] <interfacename>
peer [ local-address ] <local_ipv4>
peer [ local-address ] <local_ipv6>
peer [ multihop ]
peer [ vrf ] <vrfname>
```

Syntax

```
receive-interval <receive_interval>
```

Parameters

Name	Description	Type
receive_interval		Integer

Usage Guidelines

This command can be used to configure desired packet receive interval from BFD peer, default value is 300 milliseconds.

Examples

```
device()#configure terminal
device(config)#bfd
device(conf-bfd)# peer 192.168.0.5 interface Ethernet0
device(conf-bfd-peer)# receive-interval 200
```

receive-interval

Description

Configure packet receive interval for Bidirectional Forwarding detection(BFD) peer.

Parent Commands (Modes)

```
profile <profilename>
```

Syntax

```
receive-interval <receive_interval>
```

Parameters

Name	Description	Type
receive_interval		Integer

Usage Guidelines

This command can be used to configure desired packet receive interval from BFD peer, default value is 300 milliseconds.

Examples

```
device()#configure terminal
device(config)#bfd
device(conf-bfd)# profile fast
device(conf-bfd-profile)# receive-interval 200
```

redistribute

Description

Redistribute information from another routing protocol to BGP. User will have an option to apply a route-map to control the routes that can be redistributed into BGP.

Parent Commands (Modes)

address-family ipv4 unicast

Syntax

```
redistribute { connected | kernel | static | ospf } [ route-map <route-map-name> ] [ metric <metvalue> ]
no redistribute { connected | kernel | static | ospf } [ route-map <route-map-name> ] [ metric <metvalue> ]
```

Parameters

Name	Description	Type
route-map-name	WORD	String
metvalue	(0-4294967295)	Integer

Usage Guidelines

User can provide a route-map while enabling redistribution of routes to control the routes that goes into BGP. User can also use metric option to set the default metric for the redistributed routes.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# redistribute connected
```

redistribute

Description

Redistribute information from another routing protocol

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
redistribute { connected | kernel | static | ospfv3 } [ route-map <route-map-name> ] [ metric
    <metvalue> ]
no redistribute { connected | kernel | static | ospfv3 } [ route-map <route-map-name> ] [
    metric <metvalue> ]
```

Parameters

Name	Description	Type
route-map-name	WORD	String
metvalue	(0-4294967295)	Integer

redistribute

Description

Configures route redistribution into OSPFv2 router.

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
redistribute { { bgp { [ metric <bgpmetricval> ] } { [ metric-type <bgpmertrictype> ] } { [ route-map <bgproutemapname> ] } } | { connected { [ metric <connmetricval> ] } { [ metric-type <connmertrictype> ] } { [ route-map <connroutemapname> ] } } | { static { [ metric <staticmetricval> ] } { [ metric-type <staticmertrictype> ] } { [ route-map <staticroutemapname> ] } } | { kernel { [ metric <kernelmetricval> ] } { [ metric-type <kernelmertrictype> ] } { [ route-map <kernelroutemapname> ] } } }
```

```
no redistribute { { bgp [ metric ] [ metric-type ] [ route-map ] } | { connected [ metric ] [ metric-type ] [ route-map ] } | { static [ metric ] [ metric-type ] [ route-map ] } | { kernel [ metric ] [ metric-type ] [ route-map ] } }
```

Parameters

Name	Description	Type
bgpmetricval		Integer
bgpmertrictype		Integer
bgproutemapname	WORD	String
connmetricval		Integer
connmertrictype		Integer
connroutemapname	WORD	String
staticmetricval		Integer
staticmertrictype		Integer
staticroutemapname	WORD	String
kernelmetricval		Integer
kernelmertrictype		Integer
kernelroutemapname	WORD	String

Usage Guidelines

Use this command to configure route redistribution into OSPFv2 router. Redistributed route's metric and metric-type can be modified using this commands. Route map can also be aslo be applied to redistributed route. Unconfiguring any of the redistribute attribute for a protocol will unconfigure all attributes.

Examples

```
sonic-clt(config-router-ospf)# redistribute bgp
sonic-clt(config-router-ospf)# redistribute bgp route-map bgpospfrmap
sonic-clt(config-router-ospf)# redistribute static metric 10 metrict-type 1 routemap redist_rmap
sonic-clt(config-router-ospf)# redistribute connected metric 10 metrict-type 1
```

Features this CLI belongs to

- OSPFv2 ## refresh ### Description

Configures OSPFv2 LSA refresh interval.

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
refresh timer <refreshtimer>
no refresh timer
```

Parameters

Name	Description	Type
refreshtimer		Integer

Usage Guidelines

Use this command to configure OSPFv2 LSA refresh interval.

Examples

```
sonic-cli(config-router-ospf)# refresh timer 20
```

Features this CLI belongs to

- OSPFv2 ## remark ### Description

Set an ACL remark or description.

Parent Commands (Modes)

```
mac access-list <access-list-name>
```

Syntax

```
remark <remark-val>
no remark
```

Parameters

Name	Description	Type
remark-val	String (Max: 256 characters)	String

Usage Guidelines

Remark with spaces should be mentioned in double quotes.

Examples

```
sonic(config-mac-acl)# remark"Example ACL remark"
```

remark

Description

Set an ACL remark or description.

Parent Commands (Modes)

```
ip access-list <access-list-name>
```

Syntax

```
remark <remark-val>
no remark
```

Parameters

Name	Description	Type
remark-val	String (Max: 256 characters)	String

Usage Guidelines

Remark with spaces should be mentioned in double quotes.

Examples

```
sonic(config-mac-acl)# remark"Example ACL remark"
```

remark

Description

Set an ACL remark or description.

Parent Commands (Modes)

```
ipv6 access-list <access-list-name>
```

Syntax

```
remark <remark-val>
no remark
```

Parameters

Name	Description	Type
remark-val	String (Max: 256 characters)	String

Usage Guidelines

Remark with spaces should be mentioned in double quotes.

Examples

```
sonic(config-mac-acl)# remark"Example ACL remark"
```

remark

Description

Set an ACL remark or description.

Parent Commands (Modes)

```
ip host-access-list <access-list-name>
```

Syntax

```
remark <remark-val>
no remark
```

Parameters

Name	Description	Type
remark-val	String (Max: 256 characters)	String

Usage Guidelines

Remark with spaces should be mentioned in double quotes.

Examples

```
sonic(config-mac-acl)# remark"Example ACL remark"
```

remark

Description

Set an ACL remark or description.

Parent Commands (Modes)

```
ipv6 host-access-list <access-list-name>
```

Syntax

```
remark <remark-val>
no remark
```

Parameters

Name	Description	Type
remark-val	String (Max: 256 characters)	String

Usage Guidelines

Remark with spaces should be mentioned in double quotes.

Examples

```
sonic(config-mac-acl)# remark"Example ACL remark"
```

remote-as

Description

This command configure the remote-as number for a BGP neighbor. This command also can tag a neighbor (dynamic) as internal (iBGP) or external (eBGP)

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
remote-as { internal | external | <as-num-dot> }
no remote-as { [ internal ] | [ external ] | [ <as-num-dot> ] }
```

Parameters

Name	Description	Type
as-num-dot	1-4294967295	Integer

Usage Guidelines

remote-as configuration for a BGP neighbor is mandatory. User must configure remote-as right after creating the BGP neighbor. User can either specify the remote AS number or can specify whether a neighbor is internal or external.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# remote-as 65100
```

remote-as

Description

This command configures the remote-as number for a BGP peer-group. This command also can tag a peer-group as internal (iBGP) or external (eBGP)

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
remote-as { internal | external | <as-num-dot> }
no remote-as { [ internal ] | [ external ] | [ <as-num-dot> ] }
```

Parameters

Name	Description	Type
as-num-dot	1-4294967295	Integer

Usage Guidelines

This command configures remote-as number for a BGP peer-group. User can either specify the remote AS number or can specify whether a peer-group is internal or external.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp-pg)# remote-as 65200
```

remove-private-as

Description

This command configures BGP to remove private AS numbers from as-path in outbound BGP updates to neighbor

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
remove-private-as [ all ] [ replace-as ]
no remove-private-as [ all ] [ replace-as ]
```

Usage Guidelines

Use this command at the boundary of your BGP network to remove the internal/private AS numbers from outbound route updates. User can optionally choose to replace private AS number by local AS number.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# remove-private-as all
```

remove-private-as

Description

This command configures BGP to remove private AS numbers from as-path in outbound BGP updates to neighbors in a peer-group

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
remove-private-as [ all ] [ replace-as ]
no remove-private-as [ all ] [ replace-as ]
```

Usage Guidelines

Use this command at the boundary of your BGP network to remove the internal/private AS numbers from outbound route updates. User can optionally choose to replace private AS number by local AS number.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# address-family ipv4 unicast
sonic(config-router-bgp Pg-af)# remove-private-as all
```

remove-private-as

Description

Remove private ASNs in outbound updates

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
remove-private-as [ all ] [ replace-as ]
no remove-private-as [ all ] [ replace-as ]
```

remove-private-as

Description

Remove private ASNs in outbound updates

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
remove-private-as [ all ] [ replace-as ]
no remove-private-as [ all ] [ replace-as ]
```

renew dhcp-lease

Description

Renew DHCP lease

Syntax

```
renew dhcp-lease interface { Management <mgmt-if-id> }
```

Parameters

Name	Description	Type
mgmt-if-id		Integer

report-interval

Description

This command is used to set report interval for all sensors within subscription.

Parent Commands (Modes)

```
subscription <sub-id>
```

Syntax

```
report-interval <report_interval_val>
```

Parameters

Name	Description	Type
report_interval_val	1-4294967295	Integer

Usage Guidelines

Use this command to set report interval for all sensors within subscription.

Examples

```
sonic(config-telemetry)# subscription s1
sonic(conf-tm-sub-s1)# report-interval 5000
```

report-type

Description

This command is used to set report type for the current subscription.

Parent Commands (Modes)

```
subscription <sub-id>
```

Syntax

```
report-type <report_type_val>
```

Parameters

Name	Description	Type
report_type_val	WORD	String

Usage Guidelines

Use this command to set report type for the current subscription; can be either of periodic, stream or once.

Examples

```
sonic(config-telemetry)# subscription s1
sonic(conf-tm-sub-s1)# report-type periodic
```

request-data-size

Description

```
Configure ICMP request data size
```

Parent Commands (Modes)

```
icmp-echo <addr>
```

Syntax

```
request-data-size <size>
no request-data-size
```

Parameters

Name	Description	Type
size		Integer

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-10)# icmp-echo 30.30.1.2
sonic(conf-ipsla-10-icmp)# request-data-size 128
```

route-map

Description

This command configures policy for BGP neighbor using a route-map. The policy can be applied in INBOUND or OUTBOUND direction

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
route-map <route-name-str> { in | out }
no route-map <route-name-str> { [ in ] | [ out ] ] }
```

Parameters

Name	Description	Type
route-name-str	WORD	String

Usage Guidelines

Use this command to configure policy for BGP neighbor. The policy can be applied in inbound or outbound direction. The policy will dictate if a subset of routes needs to be filtered out or/and if attributes of some routes needs to be modified

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# route-map rmap_filter_intra_routes in
```

route-map

Description

This command configures policy for BGP neighbors in peer-group using a route-map. The policy can be applied in INBOUND or OUTBOUND direction

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
route-map <route-name-str> { in | out }
no route-map <route-name-str> { [ in ] | [ out ] ] }
```

Parameters

Name	Description	Type
route-name-str	WORD	String

Usage Guidelines

Use this command to configure policy for BGP peer-group. The policy can be applied in inbound or outbound direction. The policy will dictate if a subset of routes needs to be filtered out or/and if attributes of some routes needs to be modified

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# address-family ipv4 unicast
sonic(config-router-bgp Pg Af)# route-map RM_Blk_192 in
```

route-map

Description

This command configures policy for BGP neighbor using a route-map. The policy can be applied in INBOUND or OUTBOUND direction

Parent Commands (Modes)

address-family l2vpn evpn

Syntax

```
route-map <route-name-str> { in | out }
no route-map <route-name-str> { [ in ] | [ out ] }
```

Parameters

Name	Description	Type
route-name-str	WORD	String

Usage Guidelines

Use this command to configure policy for BGP neighbor. The policy can be applied in inbound or outbound direction. The policy will dictate if a subset of routes needs to be filtered out or/and if attributes of some routes needs to be modified

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)# route-map rmap_filter_intra_routes in
```

route-map

Description

This command configures policy for BGP neighbors in peer-group using a route-map. The policy can be applied in INBOUND or OUTBOUND direction

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
route-map <route-name-str> { in | out }
no route-map <route-name-str> { [ in ] | [ out ] } }
```

Parameters

Name	Description	Type
route-name-str	WORD	String

Usage Guidelines

Use this command to configure policy for BGP peer-group. The policy can be applied in inbound or outbound direction. The policy will dictate if a subset of routes needs to be filtered out or/and if attributes of some routes needs to be modified

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp- pg)# address-family l2vpn evpn
sonic(config-router-bgp- pg- af)# route-map RM_Blk_192 in
```

route-map

Description

Name of the route map

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
route-map <route-name-str> { in | out }
no route-map <route-name-str> { [ in ] | [ out ] ] }
```

Parameters

Name	Description	Type
route-name-str	WORD	String

route-map

Description

Name of the route map

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
route-map <route-name-str> { in | out }
no route-map <route-name-str> { [ in ] | [ out ] ] }
```

Parameters

Name	Description	Type
route-name-str	WORD	String

route-map

Description

Configure routing policies

Parent Commands (Modes)

configure terminal

Syntax

```
route-map <route-map-name> { permit | deny } <seq-nu>
no route-map <route-map-name> { { [ permit <seq-nu> ] } | { [ deny <seq-nu> ] } }
```

Parameters

Name	Description	Type
route-map-name	WORD	String
seq-nu		Integer

route-map delay-timer

Description

This command sets the route-map change processing delay interval in seconds.

Parent Commands (Modes)

router bgp <as-num-dot> { [vrf <vrf-name>] }

Syntax

```
route-map delay-timer <delaytm>
no route-map delay-timer
```

Parameters

Name	Description	Type
delaytm		Integer

Usage Guidelines

Change in route-map may require BGP RIB to get re-processed to reflect the change in policy. This command set the interval in seconds to wait before processing route-map change.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# route-map delay-timer 60
```

route-reflector allow-outbound-policy

Description

This command allows to set the outbound policy for route reflector neighbors.

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
route-reflector allow-outbound-policy
no route-reflector allow-outbound-policy
```

Usage Guidelines

Use this command to allow users to set outbound policy for route reflector neighbors.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# route-reflector allow-outbound-policy
```

route-reflector-client

Description

This command configures a BGP neighbor as route reflector client.

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
route-reflector-client
no route-reflector-client
```

Usage Guidelines

Use this command to configure an IBGP neighbor a route reflector client. This command will implicitly make the local router a route reflector server.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# route-reflector-client
```

route-reflector-client

Description

This command configures BGP neighbors in a peer-group as route reflector client.

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
route-reflector-client  
no route-reflector-client
```

Usage Guidelines

Use this command to configure an IBGP peer-group route reflector client.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Int  
sonic(config-router-bgp Pg)# address-family ipv4 unicast  
sonic(config-router-bgp Pg#af)# route-reflector-client
```

route-reflector-client

Description

This command configures a BGP neighbor as route reflector client.

Parent Commands (Modes)

```
address-family 12vpn evpn
```

Syntax

```
route-reflector-client  
no route-reflector-client
```

Usage Guidelines

Use this command to configure an IBGP neighbor a route reflector client.
This command will implicitly make the local router a route reflector server.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 20.20.20.2  
sonic(config-router-bgp-neighbor)# remote-as 300  
sonic(config-router-bgp-neighbor)# address-family 12vpn evpn  
sonic(config-router-bgp-neighbor-af)# route-reflector-client
```

route-reflector-client

Description

This command configures BGP neighbors in a peer-group as route reflector client.

Parent Commands (Modes)

address-family l2vpn evpn

Syntax

```
route-reflector-client  
no route-reflector-client
```

Usage Guidelines

Use this command to configure an IBGP peer-group route reflector client.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Int  
sonic(config-router-bgp Pg)# address-family l2vpn evpn  
sonic(config-router-bgp Pg-Af)# route-reflector-client
```

route-reflector-client

Description

Configure a neighbor as Route Reflector client

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
route-reflector-client  
no route-reflector-client
```

route-reflector-client

Description

Configure a neighbor as Route Reflector client

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
route-reflector-client  
no route-reflector-client
```

route-scale

Description

This CLI is used to route scale configuration

Parent Commands (Modes)

switch-resource

Syntax

```
route-scale routes max  
no route-scale routes
```

Usage Guidelines

This CLI is used to route scale configuration

Examples

```
sonic(config-switch-resource)# route-scale routes max
```

route-scale hosts

Description

This CLI is used to host scale configuration

Parent Commands (Modes)

```
switch-resource
```

Syntax

```
route-scale hosts layer2-layer3  
no route-scale hosts
```

Usage Guidelines

```
This CLI is used to host scale configuration
```

Examples

```
sonic(config-switch-resource)# route-scale hosts L2_L3
```

route-server-client

Description

```
This command configures a BGP neighbor a route server client.
```

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
route-server-client  
no route-server-client
```

Usage Guidelines

```
Use this command to configure an IBGP neighbor a route server client.
```

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 20.20.20.2  
sonic(config-router-bgp-neighbor)# remote-as 300  
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast  
sonic(config-router-bgp-neighbor-af)# route-server-client
```

route-server-client

Description

This command configures BGP neighbors in a peer-group route server client.

Parent Commands (Modes)

address-family ipv4 unicast

Syntax

```
route-server-client  
no route-server-client
```

Usage Guidelines

Use this command to configure an IBGP peer-group route server client.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# peer-group PG_Int  
sonic(config-router-bgp Pg)# address-family ipv4 unicast  
sonic(config-router-bgp Pg-af)# route-server-client
```

route-server-client

Description

This command configures a BGP neighbor a route server client.

Parent Commands (Modes)

address-family l2vpn evpn

Syntax

```
route-server-client  
no route-server-client
```

Usage Guidelines

Use this command to configure an IBGP neighbor a route server client.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn
sonic(config-router-bgp-neighbor-af)# route-server-client
```

route-server-client

Description

This command configures BGP neighbors in a peer-group route server client.

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
route-server-client
no route-server-client
```

Usage Guidelines

Use this command to configure an IBGP peer-group route server client.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Int
sonic(config-router-bgp-pg)# address-family l2vpn evpn
sonic(config-router-bgp-pg-af)# route-server-client
```

route-server-client

Description

Configure a neighbor as Route Server client

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
route-server-client  
no route-server-client
```

route-server-client

Description

Configure a neighbor as Route Server client

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
route-server-client  
no route-server-client
```

route-target

Description

This command specifies the route-target or a community to be attached while exporting routes from current vrf for a specific VNI.

This command also allows to specific route-target to be matched when importing routes into current vrf for a specific VNI

Parent Commands (Modes)

```
vni <vninum>
```

Syntax

```
route-target { { both { <rt> | auto } } | { import { <rt> | auto } } | { export { <rt> | auto } } }  
no route-target { { both { <rt> | auto } } | { import { <rt> | auto } } | { export { <rt> | auto } } }
```

Parameters

Name	Description	Type
rt	A.B.C.D:NN or NN:NN or MMMM:NN or NN:MMMM	String

Usage Guidelines

```
[no] route-target import|export|both {route-target-value}
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100 vrf Vrf1
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# vni 100
sonic(config-router-bgp-af-vni)# route-target import 11:11
sonic(config-router-bgp-af-vni)# route-target export 22:22
sonic(config-router-bgp-af-vni)# route-target both 33:33
sonic(config-router-bgp-af)# route-target both auto
sonic(config-router-bgp-af)# route-target import *:33
```

route-target

Description

This command specifies the route-target or a community to be attached while exporting routes from current vrf.

This command also allows to specific route-target to be matched when importing routes into current vrf

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
route-target { { both { <rt> | auto } } | { import { <rt> | auto } } | { export { <rt> | auto } } }
no route-target { { both { <rt> | auto } } | { import { <rt> | auto } } | { export { <rt> | auto } } }
```

Parameters

Name	Description	Type
rt	A.B.C.D:NN or NN:NN or MMMM:NN or NN:MMMM	String

Usage Guidelines

```
[no] route-target import|export|both {route-target-value}
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100 vrf Vrf1
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# route-target import 11:11
sonic(config-router-bgp-af)# route-target export 22:22
sonic(config-router-bgp-af)# route-target both 33:33
sonic(config-router-bgp-af)# route-target both auto
sonic(config-router-bgp-af)# route-target import *:33
```

router bgp

Description

This command creates an instance of BGP routing protocol in a VRF.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
no router bgp [ vrf <vrf-name> ]
```

Parameters

Name	Description	Type
as-num-dot	1-4294967295	Integer
vrf-name	WORD	String

Usage Guidelines

Use this config to create BGP routing instance in a VRF. If vrf key is not supplied by user, default-vrf is assumed. Only one instance of BGP protocol can be created in a VRF. Attempt to create more than one instance will result in command execution failure.

If a BGP instance already exists, executing this command with same AS number will simply enter into the "router bgp ..." configuration mode of the CLI.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 65300
```

router ospf

Description

Configures OSPFv2 router within a VRF

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
router ospf [ vrf <vrf-name> ]  
no router ospf [ vrf <vrf-name> ]
```

Parameters

Name	Description	Type
vrf-name	WORD	String

Usage Guidelines

Use this command to configure an OSPFv2 router. User can optionally specify VRF on which the router has to be configured. If VRF name is not specified then the command is considered for VRF default. Upon successful configuration CLI mode will be changed to config-router-ospf.

Technical details on OSPFv2 support is also available at
<http://docs.frrouting.org/en/latest/>

Examples

```
sonic-cli(config)# router ospf  
sonic-cli(config-router-ospf)#  
      or  
sonic-cli(config)# router ospf vrf Vrf-blue  
sonic-cli(config-router-ospf)#[
```

Features this CLI belongs to

- OSPFv2 ## router ospf6 ### Description

Configures OSPFv3 router within a VRF

Parent Commands (Modes)

configure terminal

Syntax

```
router ospf6 [ vrf <vrf-name> ]
no router ospf6 [ vrf <vrf-name> ]
```

Parameters

Name	Description	Type
vrf-name	WORD	String

Usage Guidelines

Use this command configure an OSPFv3 router. User can optionally specify VRF on which the router have to be configured. If VRF name is not specified then the command is considered for VRF default. Upon successful configuration CLI mode will be changed to config-router-ospf6.

Technical details on OSPFv3 support is also available at
<http://docs.frrouting.org/en/latest/>

Examples

```
sonic-cli(config)# router ospf6
sonic-cli(config-router-ospf6)#
      or
sonic-cli(config)# router ospf6 vrf Vrf-blue
sonic-cli(config-router-ospf6)#{
```

Features this CLI belongs to

- OSPFv3 ## router-id ### Description

This command configures router ID for an instance of BGP protocol

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
router-id <ip-addr>
no router-id
```

Parameters

Name	Description	Type
ip-addr	A.B.C.D	String

Usage Guidelines

Use this command to configure router ID for an instance of BGP protocol. Router ID configuration is optional for user. BGP automatically picks up one of the interface IP address as router ID if not configured explicitly by user.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# router-id 163.134.6.97
```

router-id

Description

This command configures router ID for an instance of BGP protocol

Syntax

```
router-id <ip-addr>
```

Parameters

Name	Description	Type
ip-addr	A.B.C.D	String

Usage Guidelines

Use this command to configure router ID for an instance of BGP protocol. Router ID configuration is optional for user. BGP automatically picks up one of the interface IP address as router ID if not configured explicitly by user.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# router-id 163.134.6.97
```

run

Description

Invoke commands

Syntax

```
run shell
```

sampler

Description

This command configures any configured sampler information.

Parent Commands (Modes)

```
tam
```

Syntax

```
sampler <name> rate <sampler_rate>
no sampler <name>
```

Parameters

Name	Description	Type
name	WORD	String
sampler_rate	1-4294967295	Integer

Usage Guidelines

TAM infrastructure supports setting up a sampling configuration and refer to this sampler configuration from the application which support sampling for the traffic. This command configures any configured sampler information.

Examples

```
sonic(config-tam)# sampler s34 rate 1000
sonic# show tam samplers
Name      Sample Rate
-----
s1        1
s34       1000
sonic#
```

scheduler-policy

Description

This command is used to configure scheduler policy of CPU interface.

Parent Commands (Modes)

```
interface CPU
```

Syntax

```
scheduler-policy <sp_name>
no scheduler-policy
```

Parameters

Name	Description	Type
sp_name	WORD	String

Usage Guidelines

Use this command to configure scheduler policy of CPU interface.

Examples

```
sonic(conf-if-CPU)# scheduler-policy scheduler.cpu
```

scheduler-policy

Description

Scheduler Policy configuration

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
scheduler-policy <sp_name>
no scheduler-policy
```

Parameters

Name	Description	Type
sp_name	WORD	String

send-community

Description

This command configures BGP to send community to a neighbor

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
send-community { standard | extended | both | large | all | none }
no send-community
```

Usage Guidelines

Use this command to enable sending of community attribute to a BGP neighbor. The command option provides the flexibility to enable sending of standard, extended, large communities.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# send-community
```

send-community

Description

This command configures BGP to send community to neighbors in a peer-group

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
send-community { standard | extended | both | large | all | none }
no send-community
```

Usage Guidelines

Use this command to enable sending of community attribute to BGP neighbors in a peer-group. The command option provides the flexibility to enable sending of standard, extended, large communities.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Int
sonic(config-router-bgp Pg)# address-family ipv4 unicast
sonic(config-router-bgp Pg-af)# send-community
```

send-community

Description

Send Community attribute to this neighbor

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
send-community { standard | extended | both | large | all | none }
no send-community
```

send-community

Description

Send Community attribute to this neighbor

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
send-community { standard | extended | both | large | all | none }
no send-community
```

sensor-group

Description

This command is used to create a new telemetry sensor group.

Parent Commands (Modes)

```
telemetry
```

Syntax

```
sensor-group <sg-id>
no sensor-group <sensor_group_id>
```

Parameters

Name	Description	Type
sg-id	WORD	String

Usage Guidelines

Use this command to create a new telemetry sensor group.

Examples

```
sonic# configure terminal
sonic(config)# telemetry
sonic(config-telemetry)# sensor-group sg1
sonic(conf-tm-sensor-sg1)#+
```

seq

Description

Create a MAC ACL Rule.

Parent Commands (Modes)

```
mac access-list <access-list-name>
```

Syntax

```
seq <seq-no> { { deny | permit | { remark <remark-val> } } { { <src-mac-addr> <src-mac-mac> } | src-mac-any | { src-mac-host <src-mac-addr> } } { { <dst-mac-addr> <dst-mac-mask> } | dst-mac-any | { dst-mac-host <dst-mac-addr> } } { [ ethertype-ip ] | [ ethertype-ipv6 ] | [ ethertype-arp ] | [ <ETHERTYPE> ] ] } { [ pcp { pcp-be | pcp-bk | pcp-ee | pcp-ca | pcp-vi | pcp-vo | pcp-ic | pcp-nc | { <pcp-val> { [ pcp-mask <pcp-val-mask> ] } } ] } { [ dei <dei-val> ] } { [ remark <remark-val> ] } }
no seq <seq-no> [ remark ]
```

Parameters

Name	Description	Type
seq-no		Integer
remark-val	String (Max: 256 characters)	String
src-mac-addr	MACADDRESS	String
src-mac-mac	MACADDRESS	String
dst-mac-addr	MACADDRESS	String
dst-mac-mask	MACADDRESS	String
ETHERTYPE	0x600-0xffff	String
pcp-val	0-7	Integer
pcp-val-mask	0-7	Integer
dei-val	0-1	Integer

Usage Guidelines

The rule will be created if there is no existing rule with the same sequence number. ACL Rule can't be updated. To update, the rule must be deleted and added with updated parameters.

Examples

```
sonic(config-mac-acl)# seq 10 permit host 00:00:10:00:00:01 host 00:00:20:00:00:01
```

seq

Description

Create a IPv4 ACL Rule.

Parent Commands (Modes)

```
ip access-list <access-list-name>
```

Syntax

```
seq <seq-no> { { deny | permit | { remark <remark-val> } } { <ip-protocol-val> | icmp | ip |  
    tcp | udp } { <src-ip-prefix> | src-ip-any | { src-ip-host <src-ip> } } { { [ src-eq  
    <src-port1> ] } | { [ src-gt <src-port1> ] } | { [ src-lt <src-port1> ] } | { [ src-range  
    <src-port1> <src-port2> ] } } { <dst-ip-prefix> | dst-ip-any | { dst-ip-host <dst-ip> } }  
    { { [ dst-eq <dst-port1> ] } | { [ dst-gt <dst-port1> ] } | { [ dst-lt <dst-port1> ] } | {  
    [ dst-range <dst-port1> <dst-port2> ] } } { [ dscp { default | cs1 | cs2 | cs3 | cs4 |  
    cs5 | cs6 | cs7 | af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 |  
    af42 | af43 | ef | voice-admit | <dscp-val> } ] } [ tcp-established ] { [ fin ] | [ not-fin  
    ] ] } { [ syn ] | [ not-syn ] ] } { [ rst ] | [ not-rst ] ] } { [ psh ] | [ not-psh ] ] }  
    { [ ack ] | [ not-ack ] ] } { [ urg ] | [ not-urg ] ] } { [ type <TYPE> ] } { [ code <CODE>  
    ] } { [ remark <remark-val> ] } { [ aeth-syndrome <aeth-syndrome-val> ] } }  
no seq <seq-no> [ remark ]
```

Parameters

Name	Description	Type
seq-no		Integer
remark-val	String (Max: 256 characters)	String
ip-protocol-val		Integer
src-ip-prefix	A.B.C.D/mask	String
src-ip	A.B.C.D	String
src-port1		Integer
src-port2		Integer
dst-ip-prefix	A.B.C.D/mask	String
dst-ip	A.B.C.D	String
dst-port1		Integer
dst-port2		Integer
dscp-val		Integer
TYPE		Integer
CODE		Integer
aeth-syndrome-val	Double hexadecimal type for AETH_SYNDROME	String

Usage Guidelines

The rule will be created if there is no existing rule with the same sequence number. ACL Rule can't be updated. To update, the rule must be deleted and added with updated parameters.

Examples

```
sonic(config-ip-acl)# seq 10 permit ip host 10.1.1.1 host 20.1.1.1
```

seq

Description

Create a IPv6 ACL Rule.

Parent Commands (Modes)

```
ipv6 access-list <access-list-name>
```

Syntax

```
seq <seq-no> { { deny | permit | { remark <remark-val> } } { <ip-protocol-val> | icmpv6 | ipv6  
| tcp | udp } { <src-ip-prefix> | src-ip-any | { src-ip-host <src-ip> } } { { [ src-eq  
<src-port1> ] } | { [ src-gt <src-port1> ] } | { [ src-lt <src-port1> ] } | { [ src-range  
<src-port1> <src-port2> ] } } { <dst-ip-prefix> | dst-ip-any | { dst-ip-host <dst-ip> } }  
{ { [ dst-eq <dst-port1> ] } | { [ dst-gt <dst-port1> ] } | { [ dst-lt <dst-port1> ] } | {  
[ dst-range <dst-port1> <dst-port2> ] } } { [ dscp { default | cs1 | cs2 | cs3 | cs4 |  
cs5 | cs6 | cs7 | af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 |  
af42 | af43 | ef | voice-admit | <dscp-val> } ] } [ tcp-established ] { [ fin ] | [ not-fin  
] ] } { [ syn ] | [ not-syn ] ] } { [ rst ] | [ not-rst ] ] } { [ psh ] | [ not-psh ] ] } {  
[ ack ] | [ not-ack ] ] } { [ urg ] | [ not-urg ] ] } { [ type <TYPE> ] } { [ code <CODE>  
] } { [ remark <remark-val> ] } { [ aeth-syndrome <aeth-syndrome-val> ] } }  
no seq <seq-no> [ remark ]
```

Parameters

Name	Description	Type
seq-no		Integer
remark-val	String (Max: 256 characters)	String
ip-protocol-val		Integer
src-ip-prefix	A::B/mask	String
src-ip	A::B	String
src-port1		Integer
src-port2		Integer
dst-ip-prefix	A::B/mask	String
dst-ip	A::B	String
dst-port1		Integer
dst-port2		Integer
dscp-val		Integer
TYPE		Integer
CODE		Integer
aeth-syndrome-val	Double hexadecimal type for AETH_SYNDROME	String

Usage Guidelines

The rule will be created if there is no existing rule with the same sequence number. ACL Rule can't be updated. To update, the rule must be deleted and added with updated parameters.

Examples

```
sonic(config-ip-acl)# seq 100 permit ipv6 host abcd::1 host bcde::1
```

seq

Description

Create a ipv4 host-ACL Rule.

Parent Commands (Modes)

```
ip host-access-list <access-list-name>
```

Syntax

```
seq <seq-no> { { deny | permit } { <ip-protocol-val> | { ip { <src-ip-prefix> | src-ip-any | { src-ip-host <src-ip> } } } { [ dscp { default | cs1 | cs2 | cs3 | cs4 | cs5 | cs6 | cs7 | af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 | af42 | af43 | ef | voice-admit | <dscp-val> } ] } { [ remark <remark-val> ] } { [ aeth-syndrome <aeth-syndrome-val> ] } }  
no seq <seq-no> [ remark ]
```

Parameters

Name	Description	Type
seq-no		Integer
ip-protocol-val		Integer
src-ip-prefix	A.B.C.D/mask	String
src-ip	A.B.C.D	String
dscp-val		Integer
remark-val	String (Max: 256 characters)	String
aeth-syndrome-val	Double hexadecimal type for AETH_SYNDROME	String

Usage Guidelines

The rule will be created if there is no existing rule with the same sequence number. ACL Rule can't be updated. To update, the rule must be deleted and added with updated parameters.

Examples

```
sonic(config-ipv4-host-acl)# seq 100 permit ip 10.10.10.1/32 any
```

seq

Description

Create a ipv6 host-ACL Rule.

Parent Commands (Modes)

```
ipv6 host-access-list <access-list-name>
```

Syntax

```
seq <seq-no> { { deny | permit } { <ip-protocol-val> | { ipv6 { <src-ip-prefix> | src-ip-any | { src-ip-host <src-ip> } } } { [ dscp { default | cs1 | cs2 | cs3 | cs4 | cs5 | cs6 | cs7 | af11 | af12 | af13 | af21 | af22 | af23 | af31 | af32 | af33 | af41 | af42 | af43 | ef | voice-admit | <dscp-val> } ] } { [ remark <remark-val> ] } { [ aeth-syndrome <aeth-syndrome-val> ] } }
```

```
no seq <seq-no> [ remark ]
```

Parameters

Name	Description	Type
seq-no		Integer
ip-protocol-val		Integer
src-ip-prefix	A::B/mask	String
src-ip	A::B	String
dscp-val		Integer
remark-val	String (Max: 256 characters)	String
aeth-syndrome-val	Double hexadecimal type for AETH_SYNDROME	String

Usage Guidelines

The rule will be created if there is no existing rule with the same sequence number. ACL Rule can't be updated. To update, the rule must be deleted and added with updated parameters.

Examples

```
sonic(config-ipv6-host-acl)# seq 100 permit ipv6 fdbd:abcd::1/128 any
```

server-key

Description

This command configures a global shared secret that is used for all dynamic authorization clients that do not have an individual shared secret key configured.

Parent Commands (Modes)

```
aaa server radius dynamic-author
```

Syntax

```
server-key <key> [ encrypted ]
no server-key
```

Parameters

Name	Description	Type
key	WORD	String

Usage Guidelines

This command configures a global shared secret that is used for all dynamic authorization clients that do not have an individual shared secret key configured.

Examples

```
sonic-cli(config)# aaa server radius dynamic-author
sonic-cli(config-radius-da)# server-key U2FsdGVkX18LcQREyGJP/aDuWTi34jJLzdqNy9W5WxI= encrypted
sonic-cli(config-radius-da)#

```

service

Description

Configure services

Parent Commands (Modes)

```
ip host-access-list <access-list-name>
```

Syntax

```
service <service-name>
```

Parameters

Name	Description	Type
service-name	String(Max: 72 characters)	String

service

Description

```
Configure services
```

Parent Commands (Modes)

```
ipv6 host-access-list <access-list-name>
```

Syntax

```
service <service-name>
```

Parameters

Name	Description	Type
service-name	String(Max: 72 characters)	String

service-policy

Description

```
Applies ingress/egress service policy on given interface
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
service-policy type { qos | monitoring | forwarding | copp | acl-copp } { in | out }
    <fbps-policy-name>
no service-policy type { qos | monitoring | forwarding | copp | acl-copp } { in | out }
```

Parameters

Name	Description	Type
fbs-policy-name	WORD	String

Examples

```
sonic(conf-if-Vlan100)# service-policy type forwarding in policy_vrf
```

service-policy

Description

Applies ingress/egress service policy on given interface

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
service-policy type { qos | monitoring | forwarding | copp | acl-copp } { in | out }
    <fbs-policy-name>
no service-policy type { qos | monitoring | forwarding | copp | acl-copp } { in | out }
```

Parameters

Name	Description	Type
fbs-policy-name	WORD	String

Examples

```
sonic(conf-if-Vlan100)# service-policy type forwarding in policy_vrf
```

service-policy

Description

Applies ingress/egress service policy on given interface

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
service-policy type { qos | monitoring | forwarding | copp | acl-copp } { in | out }
    <fbps-policy-name>
no service-policy type { qos | monitoring | forwarding | copp | acl-copp } { in | out }
```

Parameters

Name	Description	Type
fbps-policy-name	WORD	String

Examples

```
sonic(conf-if-Vlan100)# service-policy type forwarding in policy_vrf
```

service-policy

Description

```
Applies ingress/egress service policy on given interface
```

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [
    fallback ] [ fast_rate ]
```

Syntax

```
service-policy type { qos | monitoring | forwarding | copp | acl-copp } { in | out }
    <fbps-policy-name>
no service-policy type { qos | monitoring | forwarding | copp | acl-copp } { in | out }
```

Parameters

Name	Description	Type
fbs-policy-name	WORD	String

Examples

```
sonic(conf-if-Vlan100)# service-policy type forwarding in policy_vrf
```

service-policy

Description

Applies ingress/egress service policy on given interface

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
service-policy type { qos | monitoring | forwarding | copp | acl-copp } { in | out }
    <fbs-policy-name>
no service-policy type { qos | monitoring | forwarding | copp | acl-copp } { in | out }
```

Parameters

Name	Description	Type
fbs-policy-name	WORD	String

Examples

```
sonic(conf-if-Vlan100)# service-policy type forwarding in policy_vrf
```

service-policy

Description

Applies ingress/egress service policy on given interface

Parent Commands (Modes)

```
interface CPU
```

Syntax

```
service-policy type { qos | monitoring | forwarding | copp | acl-copp } { in | out }
<fbps-policy-name>
no service-policy type { qos | monitoring | forwarding | copp | acl-copp } { in | out }
```

Parameters

Name	Description	Type
fbps-policy-name	WORD	String

Examples

```
sonic(conf-if-Vlan100)# service-policy type forwarding in policy_vrf
```

session

Description

This command creates IFA monitoring session associated a previously defined flow-group.

Parent Commands (Modes)

```
ifa
```

Syntax

```
session <session_name> flowgroup <flowgroup_name> [ collector <collector_name> ] [ sampler
<sampler_name> ] node-type <node_type>
no session <session_name>
```

Parameters

Name	Description	Type
session_name	WORD	String
flowgroup_name	WORD	String
collector_name	WORD	String
sampler_name	WORD	String
node_type	IFA Node Role	Select [ingress(INGRESS) egress(EGRESS)]

Usage Guidelines

This command creates IFA monitoring session associated a previously defined flow-group.

Examples

```
sonic(config-tam-ifa)# session ss1 flowgroup f9 sampler s1 node-type ingress
sonic(config-tam-ifa)#
sonic# show tam ifa sessions
Name          Flow Group        Collector        Sampler        Node Type
-----        -----
ss1           f9                  s1              Ingress Node
sonic#
```

session

Description

This command creates drop monitoring session associated a previously defined flow-group.

Parent Commands (Modes)

drop-monitor

Syntax

```
session <session_name> flowgroup <flowgroup_name> collector <collector_name> sampler
      <sampler_name>
no session <session_name>
```

Parameters

Name	Description	Type
session_name	WORD	String
flowgroup_name	WORD	String

Name	Description	Type
collector_name	WORD	String
sampler_name	WORD	String

Usage Guidelines

This command creates drop monitoring session associated a previously defined flow-group.

Examples

```
sonic# configure terminal
sonic(config)# tam
sonic(config-tam)# drop-monitor
sonic(config-tam-dm)# session ss91 flowgroup f9 collector c2 sampler s1
sonic(config-tam-dm)# end
sonic# show tam drop-monitor sessions
Name          Flow Group      Collector      Sampler
-----        -----
ss1           f1             c1            s1
ss2           DEMO           c1            s2
ss91          f9             c2            s1
sonic#
```

session

Description

This command creates tail stamping session associated a previously defined flow-group.

Parent Commands (Modes)

tail-stamping

Syntax

```
session <session_name> flowgroup <flowgroup_name> [ node-type <node_type> ]
no session <session_name>
```

Parameters

Name	Description	Type
session_name	WORD	String
flowgroup_name	WORD	String

Name	Description	Type
node_type	Tail Stamping Node Role	Select [normal(NORMAL) ifa(IFA)]

Usage Guidelines

This command creates tail stamping session associated a previously defined flow-group.

Examples

```
sonic# configure terminal
sonic(config)# tam
sonic(config-tam)# tail-stamping
sonic(config-tam-ts)# session ss66 flowgroup f9 node-type ifa
sonic(config-tam-ts)# end
sonic# show tam tail-stamping sessions
Name          Flow Group      Node Type
-----        -----
ss66          f9            IFA
ss99          f10           Normal
sonic#
```

session-timeout

Description

Configures MLAG session timeout value in seconds

Parent Commands (Modes)

mclag domain <mclag-domain-id>

Syntax

```
session-timeout <ST>
no session-timeout
```

Parameters

Name	Description	Type
ST		Integer

Usage Guidelines

Use this command to change the default MLAG session timeout

Examples

```
sonic-cl(i(config-mlag-domain-100)#session-timeout 100
```

set as-path prepend

Description

Prepend to the as-path

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
set as-path prepend <as-number_list>
no set as-path prepend
```

Parameters

Name	Description	Type
as-number_list	asn list	String

set as-path set

Description

Set the as-path

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
set as-path set <as-path>
no set as-path set
```

Parameters

Name	Description	Type
as-path	AS-path	String

set comm-list

Description

```
Set BGP community list (for deletion)
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
set comm-list <comm_nm> delete
no set comm-list
```

Parameters

Name	Description	Type
comm_nm	String	String

set community

Description

```
BGP community attribute
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
set community { { [ <comm-num1> { [ <comm-num2> { [ <comm-num3> { [ <comm-num4> { [ <comm-num5>
    [ local-as ] [ no-advertise ] [ no-export ] [ no-peer ] [ additive ] ] } [ local-as ] [
    no-advertise ] [ no-export ] [ no-peer ] [ additive ] ] } [ local-as ] [ no-advertise ] [
    no-export ] [ no-peer ] [ additive ] ] } [ local-as ] [ no-advertise ] [ no-export ] [
    no-peer ] [ additive ] ] } [ local-as ] [ no-advertise ] [ no-export ] [ no-peer ] [
    additive ] ] } [ local-as ] [ no-advertise ] [ no-export ] [ no-peer ] [ additive ] ] } |
{ [ no-advertise [ local-as ] [ no-export ] [ no-peer ] [ additive ] ] } | { [ no-export [
local-as ] [ no-advertise ] [ no-peer ] [ additive ] ] } | { [ no-peer [ local-as ] [
no-advertise ] [ no-export ] [ additive ] ] } | [ none ] }
```

```
no set community { { [ <comm-num1> { [ <comm-num2> { [ <comm-num3> { [ <comm-num4> { [
<comm-num5> [ local-as ] [ no-advertise ] [ no-export ] [ no-peer ] [ additive ] ] } [ local-as ] [
no-advertise ] [ no-export ] [ no-peer ] [ additive ] ] } [ local-as ] [ no-advertise ] [
no-export ] [ no-peer ] [ additive ] ] } [ local-as ] [ no-advertise ] [ no-export ] [
no-peer ] [ additive ] ] } | { [ local-as [ no-advertise ] [ no-export ] [ no-peer ] [
additive ] ] } | { [ no-advertise [ local-as ] [ no-export ] [ no-peer ] [ additive ] ] } |
{ [ no-export [ local-as ] [ no-advertise ] [ no-peer ] [ additive ] ] } | { [ no-peer [
local-as ] [ no-advertise ] [ no-export ] [ additive ] ] } | [ none ] }
```

Parameters

Name	Description	Type
comm-num1	AA:NN	String
comm-num2	AA:NN	String
comm-num3	AA:NN	String
comm-num4	AA:NN	String
comm-num5	AA:NN	String

set copp-action

Description

```
Updates class-map match ACL attributes
```

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
set copp-action <copp-action-name>
```

Parameters

Name	Description	Type
copp-action-name	WORD	String

Examples

```
sonic(config)# class-map class_ip_acl match-type acl  
sonic(config-class-map)# match access-group ip ip_acl1
```

set dscp

Description

Configures DSCP Remarketing action for Qos Flow

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
set dscp <dscp-value>  
no set dscp
```

Parameters

Name	Description	Type
dscp-value		Integer

Examples

```
sonic(config)# policy-map policy_qos type qos  
sonic(config-policy-map)# class class_permit_ip priority 10  
sonic(config-policy-map-flow)# set dscp 10
```

set extcommunity

Description

BGP extended community attribute

Parent Commands (Modes)

route-map <route-map-name> { permit | deny } <seq-nu>

Syntax

```
set extcommunity { { rt <value> } | { soo <value> } }
no set extcommunity { { [ rt <value> ] } | { [ soo <value> ] } }
```

Parameters

Name	Description	Type
value	ASN:NN_OR_IP-ADDRESS:NN	String

set interface

Description

Configures ip interface for forwarding flow

Parent Commands (Modes)

class <fbs-class-name> [priority <fbs-flow-priority>]

Syntax

```
set interface { null | <port-id> | <portchannel-id> } [ priority <priority-value> ]
no set interface { null | <port-id> | <portchannel-id> } [ priority <priority-value> ]
```

Parameters

Name	Description	Type
port-id	EthernetNUM	
portchannel-id	PortChannelNUM	
priority-value		Integer

Usage Guidelines

Egress interfaces configuration is valid only if the classifier uses MAC/L2 ACL for match. Only L2 switched traffic will be forwarded to the configured egress interface. Combining egress interface with IPv4 or IPv6 next-hops is not permitted. Drop action(set interface to null) if configured will be of the lowest priority and will be chosen if none of the configured next-hops or egress interfaces can be used for forwarding.

Examples

```
sonic(config)# policy-map policy_vrf type forwarding
sonic(config-policy-map)# class class10 priority 10
sonic(config-policy-map-flow)# set interface Eth 1/9
sonic(config-policy-map-flow)# set interface null
```

set ip

Description

Configures ip next hop for forwarding flow

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
set ip { { next-hop <ip-address> { [ vrf { <vrf-name> | default } ] } } | { next-hop-group
<pbf-nh-grp> } | { replication-group <pbf-repl-grp> } } [ priority <priority-value> ]
no set ip { { next-hop <ip-address> { [ vrf { <vrf-name> | default } ] } } | { next-hop-group
<pbf-nh-grp> } | { replication-group <pbf-repl-grp> } } [ priority <priority-value> ]
```

Parameters

Name	Description	Type
ip-address	A.B.C.D	String
vrf-name	VRF name (prefixed by Vrf, Max: 15 characters)	String
pbf-nh-grp	WORD	String

Name	Description	Type
pbf-repl-grp priority-value	WORD	String Integer

Usage Guidelines

If the VRF name is not specified then it will be derived from the VRF of the interface on which the policy is applied or default will be used for global application.
Priority of the next-hop. Range is 1-65535. Default is 0 ie lowest priority if not configured by the user. The next-hop with the higher priority will be picked up for forwarding first.
If more than 1 next-hops have the same priority then the next-hop which is configured first will be used.

Examples

```
sonic(config)# policy-map policy_vrf type forwarding
sonic(config-policy-map)# class class_permit_ip priority 10
set ip next-hop 12.12.2.2 vrf Vrf-BLUE priority 20
```

set ip

Description

IPv4 information

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
set ip next-hop <ip-addr>
no set ip next-hop <ip-addr>
```

Parameters

Name	Description	Type
ip-addr	A.B.C.D	String

set ipv6

Description

Configures ipv6 next hop for forwarding flow

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
set ipv6 { { next-hop <ip-address> { [ vrf { <vrf-name> | default } ] } } | { next-hop-group <pbf-nh-grp> } | { replication-group <pbf-repl-grp> } } [ priority <priority-value> ]
no set ipv6 { { next-hop <ip-address> { [ vrf { <vrf-name> | default } ] } } | { next-hop-group <pbf-nh-grp> } | { replication-group <pbf-repl-grp> } } [ priority <priority-value> ]
```

Parameters

Name	Description	Type
ip-address	A::B	String
vrf-name	VRF name (prefixed by Vrf, Max: 15 characters)	String
pbf-nh-grp	WORD	String
pbf-repl-grp	WORD	String
priority-value		Integer

Usage Guidelines

If the VRF name is not specified then it will be derived from the VRF of the interface on which the policy is applied or default will be used for global application.
Priority of the next-hop. Range is 1-65535. Default is 0 ie lowest priority if not configured by the user. The next-hop with the higher priority will be picked up for forwarding first.
If more than 1 next-hops have the same priority then the next-hop which is configured first will be used.

Examples

```
sonic(config)# policy-map policy_vrf type forwarding
sonic(config-policy-map)# class class_permit_ipv6 priority 10
sonic(config-policy-map)# set ipv6 next-hop 1211::2 priority 20
```

set ipv6 next-hop

Description

IPv6 next-hop address

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
set ipv6 next-hop { prefer-global | { global <ip-addr> } }  
no set ipv6 next-hop { prefer-global | { global <ip-addr> } }
```

Parameters

Name	Description	Type
ip-addr	A::B	String

set local-preference

Description

```
BGP local preference path attribute
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
set local-preference <pvalue>  
no set local-preference [ <pvalue> ]
```

Parameters

Name	Description	Type
pvalue	(0-4294967295)	Integer

set metric

Description

```
Set metric value action for the routing policy
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
set metric { <metric> | rtt | +rtt | -rtt }
no set metric { [ <metric> ] | [ rtt ] | [ +rtt ] | [ -rtt ] } ]
```

Parameters

Name	Description	Type
metric	(0-4294967295) +(0-4294967295) -(0-4294967295)	String

set mirror-session

Description

```
Configures mirror session name for monitoring flow
```

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
set mirror-session <session-name>
no set mirror-session
```

Parameters

Name	Description	Type
session-name	String	String

Examples

```
sonic(config)# policy-map policy_mirror type monitoring
sonic(config-policy-map)# class class1 priority 10
sonic(config-policy-map-flow)# set mirror-session mirror1
```

set origin

Description

BGP origin code

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
set origin { egp | igrp | incomplete }
no set origin { [ egp ] | [ igrp ] | [ incomplete ] } ]
```

set pcp

Description

Configures PCP Remark action for Qos Flow

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
set pcp <pcp-value>
no set pcp
```

Parameters

Name	Description	Type
pcp-value	0-7	Integer

Examples

```
sonic(config)# policy-map policy_qos type qos
sonic(config-policy-map)# class class_permit_ip priority 10
sonic(config-policy-map-flow)# set pcp 1
```

set traffic-class

Description

Configures set traffic class action for Qos Flow

Parent Commands (Modes)

```
class <fbss-class-name> [ priority <fbss-flow-priority> ]
```

Syntax

```
set traffic-class <tc-value>
no set traffic-class
```

Parameters

Name	Description	Type
tc-value	0-7	Integer

Examples

```
sonic(config)# policy-map policy_qos type qos
sonic(config-policy-map)# class class_permit_ip priority 10
sonic(config-policy-map-flow)#set traffic-class 1
```

set trap-action

Description

Set a CoPP trap action

Parent Commands (Modes)

```
copp-action <copp-action-name>
```

Syntax

```
set trap-action <trap-action-value>
```

Parameters

Name	Description	Type
trap-action-value	Select [drop(DROP) forward(FORWARD) copy(COPY) copy_cancel(COPY_CANCEL) trap(TRAP) log(LOG) deny(DENY) transit(TRANSIT)]	

Examples

```
sonic(config-action)# set trap-action copy  
sonic(config-action)#
```

set trap-priority

Description

```
Set a CoPP trap priority
```

Parent Commands (Modes)

```
copp-action <copp-action-name>
```

Syntax

```
set trap-priority <trap-priority-value>  
no set trap-priority
```

Parameters

Name	Description	Type
trap-priority-value		Integer

Examples

```
sonic(config-action)# set trap-priority 3
sonic(config-action)#

```

set trap-queue

Description

```
Configures CPU Queue for ACL-CoPP flows
```

Parent Commands (Modes)

```
class <fbcs-class-name> [ priority <fbcs-flow-priority> ]
```

Syntax

```
set trap-queue <trap-queue-id-value>
no set trap-queue
```

Parameters

Name	Description	Type
trap-queue-id-value		Integer

Examples

```
sonic(config)# policy-map policy3 type acl-copp
sonic(config-policy-map)# class class0 priority 100
sonic(config-policy-map-flow)# set trap-queue 30
```

set trap-queue

Description

```
Set a CoPP queue id
```

Parent Commands (Modes)

```
copp-action <copp-action-name>
```

Syntax

```
set trap-queue <queue-id-value>
no set trap-queue
```

Parameters

Name	Description	Type
queue-id-value		Integer

Examples

```
sonic(config-action)# set trap-queue 3
sonic(config-action)#

```

set weight

Description

```
BGP weight for routing table
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
set weight <weigh_value>
no set weight
```

Parameters

Name	Description	Type
weigh_value	(0-4294967295)	Integer

sflow agent-id

Description

Configure sFlow agent interface

Parent Commands (Modes)

configure terminal

Syntax

```
sflow agent-id { <phy-if-name> | <vlan-if-name> | <loop-if-name> | <mgmt-if-name> }  
no sflow agent-id
```

Parameters

Name	Description	Type
phy-if-name	EthernetNUM	
vlan-if-name	VlanNUM	
loop-if-name		
mgmt-if-name		

Examples

```
sonic(config)# sflow agent-id Ethernet0  
sonic(config)#
```

sflow collector

Description

Add an sFlow Collector

Parent Commands (Modes)

configure terminal

Syntax

```
sflow collector <ip> [ <port> ] [ vrf <vrf_name> ]  
no sflow collector <ip> [ <port> ] [ vrf <vrf_name> ]
```

Parameters

Name	Description	Type
ip	A.B.C.D or A:B:C:D:E:F:G:H	String
port		Integer
vrf_name	String(Max: 16 characters)	String

Examples

```
sonic(config)# sflow collector 1.1.1.1
sonic(config)#
```

```
sonic(config)# sflow collector 1.1.1.2 port 4451
sonic(config)#
```

```
sonic(config)# sflow collector 1.1.1.2 port 4451 vrf mgmt
sonic(config)#
```

sflow default

Description

Use global sflow interface enable state

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
sflow default
```

sflow enable

Description

Enable sflow

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
sflow enable  
no sflow enable
```

sflow enable

Description

```
Enable sFlow
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
sflow enable  
no sflow enable
```

Examples

```
sonic(config)# sflow enable  
sonic(config)#
```

sflow interface enable

Description

```
Enable sFlow by default for all interfaces
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
sflow interface enable  
no sflow interface enable
```

Examples

```
sonic(config)# sflow interface enable  
sonic(config)#
```

sflow polling-interval

Description

Configure sFlow polling interval

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
sflow polling-interval <interval>  
no sflow polling-interval
```

Parameters

Name	Description	Type
interval	0 to disable	Integer

Examples

```
sonic(config)# sflow polling-interval 44  
sonic(config)#
```

sflow sampling-direction

Description

Set sampling-direction

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
sflow sampling-direction <direction>
no sflow sampling-direction
```

Parameters

Name	Description	Type
direction	Sampling direction	String

sflow sampling-direction

Description

```
Configure the sFlow sampling direction for all interfaces
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
sflow sampling-direction <sampling-direction>
no sflow sampling-direction
```

Parameters

Name	Description	Type
sampling-direction	Sampling direction	String

Examples

```
sonic(config)# sflow sampling-direction rx
sonic(config)#
```

sflow sampling-rate

Description

```
Set sampling-rate
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
sflow sampling-rate <rate>
no sflow sampling-rate
```

Parameters

Name	Description	Type
rate		Integer

sflow sampling-rate

Description

```
Configure the sFlow sampling rate for all interfaces
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
sflow sampling-rate <sampling-rate>
no sflow sampling-rate
```

Parameters

Name	Description	Type
sampling-rate		Integer

Examples

```
sonic(config)# sflow sampling-rate 256  
sonic(config)#
```

sflow source-ip

Description

Configure source ip address for sFlow packets

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
sflow source-ip <src_ip>  
no sflow source-ip
```

Parameters

Name	Description	Type
src_ip	A.B.C.D or A:B:C:D:E:F:G:H	String

Examples

```
sonic(config)# sflow source-ip 1.2.3.4  
sonic(config)#
```

show PortChannel summary

Description

LAG status and configurationn

Syntax

```
show PortChannel summary
```

show Vlan

Description

Display VLAN commands

Syntax

```
show Vlan [ <id> ]
```

Parameters

Name	Description	Type
id		Integer

show aaa

Description

Display AAA info

Syntax

```
show aaa
```

show access-group

Description

Display ACL binding summary

Syntax

```
show access-group
```

Examples

```
Ingress IPV6 access-list ipv6acl-example on Ethernet0
Ingress IP access-list ipacl-example on PortChannel1
Ingress MAC access-list macacl-example on Vlan100
```

Alternate command

```
admin@sonic:~$ show acl table [name]
```

show access-list counters

Description

```
Display current access-list counters
```

Syntax

```
show access-list counters [ access-list <access-list-name> ] [ rule <rule-name> ]
```

Parameters

Name	Description	Type
access-list-name	String	String
rule-name	String	String

show access-lists summary

Description

```
Show access-lists summary
```

Syntax

```
show access-lists summary
```

show alarm acknowledged

Description

```
Display all alarms
```

Syntax

```
show alarm acknowledged
```

show alarm all

Description

Display all alarms

Syntax

```
show alarm all
```

show alarm detail

Description

Active alarm details

Syntax

```
show alarm detail
```

show alarm id

Description

Display alarm

Syntax

```
show alarm id <id>
```

Parameters

Name	Description	Type
id	String	String

show alarm recent

Description

Alarm recent

Syntax

```
show alarm recent <interval>
```

Parameters

Name	Description	Type
interval	Select [5min(5min) 60min(60min) 24hr(24hr)]	

show alarm severity

Description

```
Display all alarms with a given severity
```

Syntax

```
show alarm severity <sev>
```

Parameters

Name	Description	Type
sev	Select [critical(critical) major(major) minor(minor) warning(warning) informational(informational)]	

show alarm summary

Description

```
Display alarm summary
```

Syntax

```
show alarm summary
```

show audit-log

Description

Display audit log

Syntax

```
show audit-log [ <stype> ]
```

Parameters

Name	Description	Type
stype	Select [all(all)]	

show authentication

Description

Show authentication modes

Syntax

```
show authentication
```

show authentication authentication-history

Description

Display the authentication manager authentication history log.

Syntax

```
show authentication authentication-history { all | <port> }
```

Parameters

Name	Description	Type
port	EthernetNUM	

show authentication clients

Description

Shows Authentication Manager clients information.

Syntax

```
show authentication clients { all | <port> }
```

Parameters

Name	Description	Type
port	EthernetNUM	

show authentication interface

Description

Enter client interface to show mab interface.

Syntax

```
show authentication interface { all | <port> }
```

Parameters

Name	Description	Type
port	EthernetNUM	

show authentication rest

Description

Display REST authentication modes

Syntax

```
show authentication rest
```

show authentication telemetry

Description

```
Display telemetry authentication modes
```

Syntax

```
show authentication telemetry
```

show bfd peer

Description

```
Display specific Bidirectional Forwarding detection(BFD) peer with the specified filters..
```

Syntax

```
show bfd peer { <peer_ipv4> | <peer_ipv6> } [ vrf <vrfname> ] [ multihop ] [ local-address { <local_ipv4> | <local_ipv6> } ] [ interface <interfacename> ]
```

Parameters

Name	Description	Type
peer_ipv4	A.B.C.D	String
peer_ipv6	A::B	String
vrfname	WORD	String
local_ipv4	A.B.C.D	String
local_ipv6	A::B	String
interfacename	Interface Type - Ranges	

Examples

```
device# show bfd peer 192.168.2.1 interface Ethernet0
BFD Peers:

    peer 192.168.2.1 vrf default interface Ethernet0
        ID: 106764218
        Remote ID: 3876686491
        Status: up
        Uptime: 0 day(s), 0 hour(s), 51 min(s), 49 sec(s)
        Diagnostics: ok
        Remote diagnostics: ok
        Peer Type: configured
        Local timers:
            Detect-multiplier: 3
            Receive interval: 300ms
            Transmission interval: 300ms
            Echo transmission interval: 0ms
        Remote timers:
            Detect-multiplier: 3
            Receive interval: 300ms
            Transmission interval: 300ms
            Echo transmission interval: 50ms
```

```
device# show bfd peer 192.168.2.1 multihop local-address 192.168.2.2
BFD Peers:

    peer 192.168.2.1 multihop local-address 192.168.2.2 vrf default
        ID: 2900535060
        Remote ID: 0
        Status: down
        Downtime: 0 day(s), 0 hour(s), 0 min(s), 30 sec(s)
        Diagnostics: ok
        Remote diagnostics: ok
        Peer Type: configured
        Local timers:
            Detect-multiplier: 3
            Receive interval: 300ms
            Transmission interval: 300ms
            Echo transmission interval: 60ms
        Remote timers:
            Detect-multiplier: 3
            Receive interval: 1000ms
            Transmission interval: 1000ms
            Echo transmission interval: 0ms
```

show bfd peer counters

Description

Displays counters for the specified Bidirectional Forwarding detection(BFD) peer with the input filters.

Syntax

```
show bfd peer counters { <peer_ipv4> | <peer_ipv6> } [ vrf <vrfname> ] [ multihop ] [ local-address { <local_ipv4> | <local_ipv6> } ] [ interface <interfacename> ]
```

Parameters

Name	Description	Type
peer_ipv4	A.B.C.D	String
peer_ipv6	A::B	String
vriname	WORD	String
local_ipv4	A.B.C.D	String
local_ipv6	A::B	String
interfacename	Interface Type - Ranges	

Examples

```
device# show bfd peer counters 192.168.2.1 interface Ethernet0
peer 192.168.2.1 vrf default interface Ethernet0
    Control packet input: 25 packets
    Control packet output: 25 packets
    Echo packet input: 0 packets
    Echo packet output: 0 packets
    Session up events: 1
    Session down events: 0
    Zebra notifications: 0
```

```
device# show bfd peer counters 192.168.2.1 multihop local-address 192.168.2.2
peer 192.168.2.1 multihop local-address 192.168.2.2 vrf default
    Control packet input: 25 packets
    Control packet output: 25 packets
    Echo packet input: 0 packets
    Echo packet output: 0 packets
    Session up events: 1
    Session down events: 0
    Zebra notifications: 0
```

show bfd peers

Description

```
Displays all Bidirectional Forwarding detection(BFD) peers or counters.
```

Syntax

```
show bfd peers [ vrf { <vrfname> | all } ] { [ brief ] | [ counters ] } ]
```

Parameters

Name	Description	Type
vrfname	WORD	String

Examples

```
device# show bfd peers
BFD Peers:

peer 192.168.2.2 vrf default interface Ethernet0
  ID: 1861362724
  Remote ID: 3644437776
  Status: up
  Uptime: 0 day(s), 0 hour(s), 0 min(s), 6 sec(s)
  Diagnostics: ok
  Remote diagnostics: ok
  Peer Type: configured
  Local timers:
    Detect-multiplier: 3
    Receive interval: 300ms
    Transmission interval: 300ms
    Echo transmission interval: 0ms
  Remote timers:
    Detect-multiplier: 3
    Receive interval: 300ms
    Transmission interval: 300ms
    Echo transmission interval: 50ms
```

```
device# show bfd peers vrf Vrf7
BFD Peers:

peer 192.168.2.2 vrf Vrf7 interface Ethernet0
  ID: 1861362724
  Remote ID: 3644437776
  Status: up
  Uptime: 0 day(s), 0 hour(s), 0 min(s), 6 sec(s)
  Diagnostics: ok
  Remote diagnostics: ok
  Peer Type: configured
  Local timers:
    Detect-multiplier: 3
    Receive interval: 300ms
    Transmission interval: 300ms
```

```
Echo transmission interval: 0ms
Remote timers:
  Detect-multiplier: 3
  Receive interval: 300ms
  Transmission interval: 300ms
  Echo transmission interval: 50ms
```

```
device# show bfd peers counters
BFD Peers:

peer 192.168.2.2 vrf default interface Ethernet0
  Control packet input: 239 packets
  Control packet output: 292 packets
  Echo packet input: 0 packets
  Echo packet output: 0 packets
  Session up events: 1
  Session down events: 0
  Zebra notifications: 0
```

```
device# show bfd peers vrf Vrf7 counters
BFD Peers:

peer 192.168.2.2 vrf Vrf7 interface Ethernet0
  Control packet input: 239 packets
  Control packet output: 292 packets
  Echo packet input: 0 packets
  Echo packet output: 0 packets
  Session up events: 1
  Session down events: 0
  Zebra notifications: 0
```

show bfd profile

Description

Displays all Bidirectional Forwarding detection(BFD) profiles or the one specified.

Syntax

```
show bfd profile [ <profilename> ]
```

Parameters

Name	Description	Type
profilename	WORD	String

Examples

show bgp all

Description

Display BGP information for all address families

Syntax

```
show bgp all [ vrf <vrf-name> ] { { peer-group [ <peer-group-name> ] } | { neighbors { [ <neighbor-ip> ] | { [ interface { Ethernet | PortChannel | Vlan } ] } } } }
```

Parameters

Name	Description	Type
vrf-name	WORD	String
peer-group-name	WORD	String
neighbor-ip	A.B.C.D/A::B	String

show bgp as-path-access-list

Description

This command displays BGP AS Path lists configured on the device.

Syntax

```
show bgp as-path-access-list [ <list-name> ]
```

Parameters

Name	Description	Type
list-name	WORD	String

Usage Guidelines

User configures AS Path lists to use it in route-maps and with neighbors to design BGP routing policies. This command enables users to display the AS Path lists configured on this device. User can provide AS path list name optional CLI key to display only that AS Path list. If AS Path list name is not specified, all AS Path lists will be displayed by this command.

Examples

```
sonic# show bgp as-path-access-list
AS path list asp_private:
  members: ^65000.*6510565109$,65107.*65200
AS path list asp_public:
  members: ^107.*2301.*709$,97.*201
```

show bgp community-list

Description

This command displays BGP community lists configured on the device.

Syntax

```
show bgp community-list [ <list-name> ]
```

Parameters

Name	Description	Type
list-name	WORD	String

Usage Guidelines

User configures community-list to use it in route-maps to design BGP routing policies. This command enables users to display the community lists configured on this device. User can provide community list name optional CLI key to display only that community list. If community list name is not specified, all community lists will be displayed by this command.

Examples

```
sonic# show bgp community-list
Expanded community list CommList_Exp:  match: ANY
  300:500
  800:900
  no-export
Standard community list CommList_RT:  match: ANY
  100:200
  no-export
  no-peer
  65100:3456
```

show bgp ext-community-list

Description

This command displays BGP extended community lists configured on the device.

Syntax

```
show bgp ext-community-list [ <list-name> ]
```

Parameters

Name	Description	Type
list-name	WORD	String

Usage Guidelines

User configures extended community-list to use it in route-maps to design BGP routing policies. This command enables users to display the extended community lists configured on this device. User can provide extended community list name optional CLI key to display only that extended community list. If extended community list name is not specified, all extended community lists will be displayed by this command.

Examples

```
sonic# show bgp ext-community-list
Standard extended community list ExtComm_AllowInt:  match: ALL
  rt:19.32.56.167:65011,rt:31.67.182.214:3001,soo:01:65010,soo:.13.175.21:65101
Standard extended community list ExtComm_BlockExt:  match: ANY
  rt:4020:65104
  soo:9.54.32.165:65200
```

show bgp ipv4

Description

This command displays BGP information including routes, neighbors, peer-group etc.

Syntax

```
show bgp ipv4 unicast [ vrf { <vrf-name> | all } ] [ summary ] { { [ <ip-addr> { [ bestpath ] | [ multipath ] ] } ] } | { [ <ip-prefix> { [ bestpath ] | [ multipath ] ] } ] } | { [ community { <aann> | local-as | no-advertise | no-export | no-peer } [ exact-match ] ] } | { [ route-map <route-map-name> ] } | [ statistics ] | { [ neighbors { { [ <neighbor-ip> { [ routes ] | [ received-routes ] | [ advertised-routes ] ] } ] } | { [ interface { { Ethernet { [ routes ] | [ received-routes ] | [ advertised-routes ] ] } } | { PortChannel { [ routes ] | [ received-routes ] | [ advertised-routes ] ] } } } | { Vlan { [ routes ] | [ received-routes ] | [ advertised-routes ] ] } } ] } | [ summary ] | { [ dampening { dampened-paths | flap-statistics | parameters } ] } ] }
```

Parameters

Name	Description	Type
vrf-name	WORD	String
ip-addr	A.B.C.D	String
ip-prefix	A.B.C.D/mask	String
aann	AA:NN	String
route-map-name	WORD	String
neighbor-ip	A.B.C.D/A::B	String

Usage Guidelines

Use this command to display BGP neighbors, routes, peer-group etc. There are various CLI options available to display various informations from BGP. User can use "vrf" option to display information from a particular VRF instance of BGP. User can also choose ipv4 or ipv6 to display information from either of the address family.

- show bgp ipv4 unicast summary

This command will display BGP global parameters as well as brief information about BGP neighbors

- show bgp ipv4 unicast

This command will show BGP local RIB routes. User can use filtering options on CLI to zoom into a subset of routes that user is interested in

- show bgp ipv4 unicast neighbors

This will display one or all BGP neighbors information in detail

- show bgp all peer-group

This will display one or all BGP peer-group information in detail

Examples

```
leaf4# show bgp ipv4 unicast summary
BGP router identifier 200.9.0.5, local AS number 100
Neighbor      V   AS MsgRcvd MsgSent InQ    OutQ Up/Down State/PfxRcd
14.14.14.1    4   400   8       2       0       0     00:00:43 0

leaf4# show bgp ipv4 unicast
BGP routing table information for VRF default
Router identifier 200.9.0.5, local AS number 100
Route status codes: * - valid, > - best
Origin codes: i - IGP, e - EGP, ? - incomplete
      Network          Next Hop        Metric      LocPref Path
*>  4.4.4.44/32      14.14.14.1      0           400 ??
*>  10.59.128.0/20   14.14.14.1      0           400 ??
*>  13.1.1.0/24     14.14.14.1      0           400 ??
*>  14.14.14.0/24   14.14.14.1      0           400 ??
*>  29.2.2.2/32     14.14.14.1      0           400 ??
*>  192.168.1.0/24  14.14.14.1      0           400 ??
*>  200.0.0.0/24    14.14.14.1      0           400 ??

leaf4# show bgp ipv4 unicast neighbors
BGP neighbor is 14.14.14.1, remote AS 400, local AS 100, external link
  Administratively shut down
  BGP version 4, remote router ID , local router ID
  BGP state = ESTABLISHED, up for 00:01:03
  Hold time is seconds, keepalive interval is 60 seconds, negotiated hold time is 180 seconds
  Minimum time between advertisement runs is seconds
  Neighbor capabilities:
    4 Byte AS: advertised and received
  Message statistics:
    InQ depth is 0
    OutQ depth is 0
              Sent      Rcvd
  Opens:        1        1
  Notifications: 0        0
  Updates:      2        8
  Keepalive:    2        2
  Route Refresh: 0        0
  Capability:   0        0
  Total:        5       11

  Local host: 14.14.14.4, Local port: 46782
  Foreign host: 14.14.14.1, Foreign port: 179
```

show bgp ipv6

Description

IPv6 information

Syntax

```

show bgp ipv6 unicast [ vrf { <vrf-name> | all } ] [ summary ] { { [ <ip-addr> { [ bestpath ] |
[ multipath ] ] } ] } | { [ <ip-prefix> { [ bestpath ] | [ multipath ] ] } ] } | { [
community { <aann> | local-as | no-advertise | no-export | no-peer } [ exact-match ] ] } | {
[ route-map <route-map-name> ] } | [ statistics ] | { [ neighbors { { [ <neighbor-ip> { [
routes ] | [ received-routes ] | [ advertised-routes ] ] } ] } | { [ interface { { Ethernet
{ [ routes ] | [ received-routes ] | [ advertised-routes ] ] } } | { PortChannel { [ routes
] | [ received-routes ] | [ advertised-routes ] ] } } | { Vlan { [ routes ] | [
received-routes ] | [ advertised-routes ] ] } } ] } ] } | [ summary ] | { [ dampening
{ dampened-paths | flap-statistics | parameters } ] } ] }

```

Parameters

Name	Description	Type
vrf-name	WORD	String
ip-addr	A::B	String
ip-prefix	A::B/mask	String
aann	AA:NN	String
route-map-name	WORD	String
neighbor-ip	A.B.C.D/A::B	String

show bgp l2vpn evpn route

Description

This command displays BGP EVPN routes in tabular format.

Syntax

```
show bgp l2vpn evpn route
```

Usage Guidelines

```
show bgp l2vpn evpn route {filters}
```

Examples

```

sonic# show bgp l2vpn evpn route
BGP table version is 2, local router ID is 10.59.142.127
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
EVPN type-1 prefix: [1]:[ESI]:[EthTag]
EVPN type-2 prefix: [2]:[EthTag]:[MAClen]:[MAC]:[IPlen]:[IP]
EVPN type-3 prefix: [3]:[EthTag]:[IPlen]:[OrigIP]
EVPN type-4 prefix: [4]:[ESI]:[IPlen]:[OrigIP]
EVPN type-5 prefix: [5]:[EthTag]:[IPlen]:[IP]

```

Network	Next Hop	Metric	LocPrf	Weight	Path
	Extended Community				
Route Distinguisher: 11:11					
*> [5]:[0]:[0]:[0.0.0.0]	0.0.0.0			32768	i
	ET:8				
*> [5]:[0]:[0]:[::]	0.0.0.0			32768	i
	ET:8				
Route Distinguisher: 22:22					
*> [2]:[0]:[48]:[52:54:00:76:be:f7]:[32]:[2.1.1.1]	1.1.1.1			32768	i
	ET:8 RT:100:268435556	Default Gateway			
*> [2]:[0]:[48]:[52:54:00:cb:f0:e3]	1.1.1.1			32768	i
	ET:8 RT:100:268435556				
*> [2]:[0]:[48]:[52:54:00:cb:f0:e3]:[32]:[2.1.1.2]	1.1.1.1			32768	i
	ET:8 RT:100:268435556				
*> [3]:[0]:[32]:[1.1.1.1]	1.1.1.1			32768	i
	ET:8 RT:100:268435556				
Route Distinguisher: 3.1.1.1:5096					
*> [5]:[0]:[24]:[3.1.1.0]	1.1.1.1	0		32768	?
	ET:8 RT:100:200 Rmac:52:54:00:76:be:f7				
Route Distinguisher: 4.1.1.2:5096					
*> [5]:[0]:[24]:[4.1.1.0]	2.2.2.2	0		0	200 ?
	RT:200:200 ET:8 Rmac:52:54:00:cb:f0:e3				
Route Distinguisher: 10.59.143.68:100					
*> [2]:[0]:[48]:[52:54:00:cb:f0:e3]:[32]:[2.1.1.2]	2.2.2.2			0	200 i
	RT:200:100 ET:8 Default Gateway				
*> [3]:[0]:[32]:[2.2.2.2]	2.2.2.2			0	200 i
	RT:200:100 ET:8				
Displayed 10 prefixes (10 paths)					
sonic#					

show bgp l2vpn evpn route detail

Description

This command displays BGP EVPN routes in detail format.

Syntax

```
show bgp l2vpn evpn route detail
```

Usage Guidelines

```
show bgp l2vpn evpn route detail {filters}
```

Examples

```
sonic# show bgp l2vpn evpn route rd 11:11 type prefix
EVPN type-2 prefix: [2] : [EthTag] : [MAClen] : [MAC]
EVPN type-3 prefix: [3] : [EthTag] : [IPlen] : [OrigIP]
EVPN type-5 prefix: [5] : [EthTag] : [IPlen] : [IP]

BGP routing table entry for 11:11:[5]:[0]:[0]:[0.0.0.0]
Paths: (1 available, best #1)
  Zebra Add: 6d21h36m
  Advertised to non peer-group peers:
    10.1.1.2
  Route [5]:[0]:[0]:[0.0.0.0] VNI 0
  Local
    0.0.0.0 from 0.0.0.0 (10.59.142.127)
      Origin IGP, weight 32768, valid, sourced, local, best (First path received)
      Extended Community: ET:8
      Last update: Wed Feb 12 17:06:15 2020

BGP routing table entry for 11:11:[5]:[0]:[0]:[::]
Paths: (1 available, best #1)
  Zebra Add: 6d21h36m
  Advertised to non peer-group peers:
    10.1.1.2
  Route [5]:[0]:[0]:[::] VNI 0
  Local
    0.0.0.0 from 0.0.0.0 (10.59.142.127)
      Origin IGP, weight 32768, valid, sourced, local, best (First path received)
      Extended Community: ET:8
      Last update: Wed Feb 12 17:06:15 2020

Displayed 2 prefixes (2 paths) with this RD (of requested type)
sonic#
```

show bgp l2vpn evpn route detail type

Description

This command displays BGP EVPN routes of a specified type in detailed format

Syntax

```
show bgp l2vpn evpn route detail type { ead | es | macip | multicast | prefix }
```

Usage Guidelines

```
show bgp l2vpn evpn route detail type ead|es|macip|multicast|prefix
```

Examples

```
Refer example from show bgp l2vpn evpn route detail
```

show bgp l2vpn evpn route rd

Description

```
This command displays BGP EVPN routes for a specific RD in detailed format.
```

Syntax

```
show bgp l2vpn evpn route rd <rdvalue> { { [ mac { <macvalue> { ip <ipvalue> } } ] } | { [ type { ead | es | macip | multicast | prefix } ] } }
```

Parameters

Name	Description	Type
rdvalue	A.B.C.D:NN or ASN:NN	String
macvalue	nn:nn:nn:nn:nn:nn	String
ipvalue	A.B.C.D/A::B	String

Usage Guidelines

```
show bgp l2vpn evpn route rd {rdvalue} {filters}
```

Examples

```
Refer example from show bgp l2vpn evpn route detail
```

show bgp l2vpn evpn route type

Description

```
This command displays BGP EVPN routes of a specified type
```

Syntax

```
show bgp l2vpn evpn route type { ead | es | macip | multicast | prefix }
```

Usage Guidelines

```
show bgp l2vpn evpn route type ead|es|macip|multicast|prefix
```

Examples

Refer example from show bgp l2vpn evpn route

```
show bgp l2vpn evpn route type macip
```

Description

This command displays BGP EVPN routes for route type macip for a specific mac or mac and ip address.

Syntax

```
show bgp l2vpn evpn route type macip [ mac { <macvalue> { [ ip <ipvalue> ] } } ]
```

Parameters

Name	Description	Type
macvalue	nn:nn:nn:nn:nn:nn	String
ipvalue	A.B.C.D/A::B	String

Usage Guidelines

```
show bgp l2vpn evpn route type macip {filters}
```

Examples

Refer example from show bgp l2vpn evpn route type macip {mac mac-addr ip address}

```
show bgp l2vpn evpn route type multicast
```

Description

This command displays BGP EVPN routes for route type multicast for a specific ip address.

Syntax

```
show bgp l2vpn evpn route type multicast [ ip <ipvalue> ]
```

Parameters

Name	Description	Type
ipvalue	A.B.C.D	String

Usage Guidelines

```
show bgp l2vpn evpn route type multicast {filters}
```

Examples

Refer example from show bgp l2vpn evpn route type multicast ip {ip address}

show bgp l2vpn evpn route type prefix

Description

This command displays BGP EVPN routes for route type prefix for a specific ip address.

Syntax

```
show bgp l2vpn evpn route type prefix [ ip <ipvalue> ]
```

Parameters

Name	Description	Type
ipvalue	A.B.C.D/mask or A::B/mask	String

Usage Guidelines

```
show bgp l2vpn evpn route type prefix {filters}
```

Examples

Refer example from show bgp l2vpn evpn route type prefix ip {ip address}

show bgp l2vpn evpn route vni

Description

This command displays BGP EVPN routes for a specified VNI

Syntax

show bgp l2vpn evpn route vni <vninum>

Parameters

Name	Description	Type
vninum	VNI	Integer

Usage Guidelines

show bgp l2vpn evpn route vni <vni-num>

Examples

Refer example from show bgp l2vpn evpn route

show bgp l2vpn evpn summary

Description

This command displays BGP summarized information for BGP L2VPN EVPN address family including
neighbors
with evpn address family activated

Syntax

show bgp l2vpn evpn summary

Usage Guidelines

```
show bgp l2vpn evpn summary
```

Examples

```
sonic# show bgp l2vpn evpn summary
BGP router identifier 10.59.142.127, local AS number 100 vrf-id 0
BGP table version 0

Neighbor      V      AS MsgRcvd MsgSent    TblVer  InQ OutQ Up/Down State/PfxRcd
10.1.1.2      4      200   11338   11337        0      0     0 6d21h29m          3

Total number of neighbors 1
Total number of neighbors established 1
sonic#
```

show bgp l2vpn evpn vni

Description

This command displays VNI information including RD, Route-targets etc.

Syntax

```
show bgp l2vpn evpn vni <vninum>
```

Parameters

Name	Description	Type
vninum	VNI	Integer

Usage Guidelines

```
show bgp l2vpn evpn vni {vni-number}
```

Examples

```
sonic# show bgp l2vpn evpn vni 100
VNI: 100 (known to the kernel)
Type: L2
Tenant-Vrf: default
RD: 22:22
Originator IP: 1.1.1.1
Mcast group: 0.0.0.0
Advertise-gw-macip : Yes
Advertise-svi-macip : No
Import Route Target:
    22:22
    22:23
Export Route Target:
    100:268435556
```

show buffer interface

Description

Display QoS priority group and queue association with QoS buffer profile on interfaces.

Syntax

```
show buffer interface <phy-intf-name> { priority-group | queue }
```

Parameters

Name	Description	Type
phy-intf-name	EthernetNUM	

Usage Guidelines

```
sonic# show buffer interface Ethernet all priority-group
sonic# show buffer interface Ethernet IFNo priority-group
sonic# show buffer interface Ethernet all queue
sonic# show buffer interface Ethernet IFNo queue
```

where IFNo is ethernet interface number.

Examples

```
sonic# show buffer interface ethernet all priority-group
Interface  Priority-group      Profile
Ethernet0      0                ingress_lossy_profile
Ethernet4      3-4              ingress_lossless_profile

sonic# show buffer interface Ethernet0 priority-group
Interface  Priority-group      Profile
Ethernet0      0                ingress_lossy_profile

sonic# show buffer interface ethernet all queue
Interface    Queue      Profile
Ethernet0     0          ingress_lossy_profile
Ethernet4     3-4        ingress_lossless_profile

sonic# show buffer interface Ethernet0 queue
Interface    Queue      Profile
Ethernet0     0          ingress_lossy_profile
```

show buffer pool

Description

Display all QoS buffer pools with their configured parameters.

Syntax

```
show buffer pool
```

Usage Guidelines

```
sonic# show buffer pool
```

Examples

```
sonic# show buffer pool
Pool egress_lossless_pool:
  mode : static
  size : 32575488 bytes
  type : egress
```

show buffer profile

Description

Display all QoS profiles with their configured parameters.

Syntax

```
show buffer profile
```

Usage Guidelines

```
sonic# show buffer profile
```

Examples

```
sonic# show buffer profile
Profile egress_lossless_profile:
  pool          : egress_lossless_pool
  mode          : static
  size          : 0
  static_threshold : 32575488 bytes
```

show buffer-pool

Description

This command is used to show user/persistent watermark counters recorded by the system.

Syntax

```
show buffer-pool { watermark | persistent-watermark }
```

Usage Guidelines

Use this command to display user/persistent watermark counters recorded by the system.

Examples

```
show buffer-pool watermark
show buffer-pool persistent-watermark
```

show class-map

Description

Shows flow based services class-map related information

Syntax

```
show class-map { [ <show-fbs-class-name> ] | [ match-type ] } ] { acl | fields | copp }
```

Parameters

Name	Description	Type
show-fbs-class-name	WORD	String

Usage Guidelines

Class-map match type and class-map name arguments are optional. If match type argument or class-map name not provided command will show all class-map information. Else it shows corresponding class-map information of given type or given name

Examples

```
sonic# show class-map class_permit_ip
Class-map class_permit_ip match-type fields
  Description:
  Match:
  Referenced in flows:
    policy policy_qos at priority 10
    policy policy_vrf at priority 10
```

show clock

Description

Display system date and time

Syntax

```
show clock
```

show config-key password-encrypt

Description

This command indicates if a user configured primary encryption key is in use by the system.

Syntax

```
show config-key password-encrypt
```

Examples

```
sonic# show config-key password-encrypt
Primary encryption key configured : False
```

show configuration

Description

Show current MAC ACL configuration.

Parent Commands (Modes)

```
mac access-list <access-list-name>
```

Syntax

```
show configuration
```

Usage Guidelines

Displays the configuration of the given MAC ACLs.

Examples

```
mac access-list macacl-example
 seq 10 permit host 00:00:10:00:00:01 host 00:00:20:00:00:01
 seq 20 permit host 00:00:10:00:00:02 host 00:00:20:00:00:02
 seq 30 permit host 00:00:10:00:00:03 host 00:00:20:00:00:03
 seq 40 permit host 00:00:10:00:00:04 host 00:00:20:00:00:04
```

show configuration

Description

Show current IPv4 ACL configuration.

Parent Commands (Modes)

```
ip access-list <access-list-name>
```

Syntax

```
show configuration
```

Usage Guidelines

```
Displays the configuration of the current IPv4 ACLs.
```

Examples

```
ip access-list ipacl-example
 seq 10 permit ip host 10.1.1.1 host 20.1.1.1
 seq 20 permit ip host 10.1.1.2 host 20.1.1.2
 seq 30 permit ip host 10.1.1.3 host 20.1.1.3
 seq 40 permit ip host 10.1.1.4 host 20.1.1.4
```

show configuration

Description

```
Show current IPv6 ACL configuration.
```

Parent Commands (Modes)

```
ipv6 access-list <access-list-name>
```

Syntax

```
show configuration
```

Usage Guidelines

```
Displays the configuration of the current IPv6 ACLs.
```

Examples

```
ipv6 access-list ipv6acl-example
 seq 100 permit ipv6 host abcd::1 host bcde::1
 seq 200 permit tcp host abcd::2 host bcde::2
 seq 300 permit udp host abcd::3 host bcde::3
```

show configuration

Description

Show current host-ACL configuration.

Parent Commands (Modes)

`ip host-access-list <access-list-name>`

Syntax

`show configuration`

Usage Guidelines

Displays the configuration of the current CTRPLANE ACLs.

Examples

```
ip host-access-list ipacl-example
  seq 100 permit ip 10.10.10.1/32 any
  seq 200 permit ip 11.11.11.1/32 12.12.12.12/32
  seq 300 permit ip 10.1.1.1/32 20.1.1.1/32
```

show configuration

Description

Show current host-ACL configuration.

Parent Commands (Modes)

`ipv6 host-access-list <access-list-name>`

Syntax

`show configuration`

Usage Guidelines

Displays the configuration of the current host-ACLs.

Examples

```
ctrl access-list ctrlacl-example  
    seq 100 permit ipv6 fdbd:abcd::1/128 any
```

show configuration

Description

```
show configuration
```

Parent Commands (Modes)

```
dropcounters <counter-name>
```

Syntax

```
show configuration
```

show configuration

Description

```
show bgp configuration
```

Parent Commands (Modes)

```
vni <vninum>
```

Syntax

```
show configuration
```

show configuration

Description

```
Shows current policy-map configuration
```

Parent Commands (Modes)

```
policy-map <fbps-policy-name> type { qos | monitoring | forwarding | copp | acl-copp }
```

Syntax

```
show configuration
```

Usage Guidelines

Use this command to display running configurations within current policy-map.

Examples

```
sonic(config-policy-map)# show configuration
policy-map policy_vrf type forwarding
  class class_permit_ipv6 priority 10
    set ipv6 next-hop 1211::2 priority 20
    set ipv6 next-hop 1212::2 vrf Vrf-BLUE priority 30
  class class_permit_ip priority 10
    set ip next-hop 12.12.1.2 vrf default priority 30
    set ip next-hop 12.12.2.2 vrf Vrf-BLUE priority 20
    set ip next-hop 12.12.1.2 priority 10
```

show configuration

Description

Shows configuration of current class-map

Parent Commands (Modes)

```
class-map <fb-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
show configuration
```

Usage Guidelines

Use this command to show the configuration of the current class-map

Examples

```
sonic(config-class-map)#show configuration
class-map class-1 match-type acl
```

show configuration

Description

Shows configuration of current class-map

Parent Commands (Modes)

```
class-map <fbcs-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
show configuration
```

Usage Guidelines

Use this command to show the configuration of the current class-map

Examples

```
sonic(config-class-map)#show configuration  
class-map class-1 match-type fields match-all
```

show configuration

Description

Shows configuration of current class-map

Parent Commands (Modes)

```
class-map <fbcs-class-name> match-type { acl | { fields match-all } | copp }
```

Syntax

```
show configuration
```

Usage Guidelines

Use this command to show the configuration of the current class-map

Examples

```
sonic(config-class-map)#show configuration  
class-map class-1 match-type copp
```

show configuration

Description

```
Shows current policy-map configuration
```

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
show configuration
```

Usage Guidelines

```
Use this command to display running configurations within current policy-map.
```

Examples

```
sonic(config-policy-map-flow)# show configuration  
!  
policy-map policy-qos-test type qos  
description "qos-test-policy"  
class class-qos-test priority 1020  
police cir 512000  
!
```

show configuration

Description

```
Shows current policy-map configuration
```

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
show configuration
```

Usage Guidelines

Use this command to display running configurations within current policy-map.

Examples

```
sonic(config-policy-map-flow)# show configuration
!
policy-map policy-monitoring type monitoring
  description "monitoring-policy"
  class class-monitoring priority 1020
    set mirror-session test
!
```

show configuration

Description

Shows current policy-map configuration

Parent Commands (Modes)

```
class <fbs-class-name> [ priority <fbs-flow-priority> ]
```

Syntax

```
show configuration
```

Usage Guidelines

Use this command to display running configurations within current policy-map.

Examples

```
sonic(config-policy-map-flow)# show configuration
!
policy-map policy-forwarding-test type forwarding
  description "forwarding-policy"
  class class_permit_ip priority 10
    set ip next-hop 12.12.1.2 vrf default priority 30
    set ip next-hop 12.12.2.2 vrf Vrf-BLUE priority 20
    set ip next-hop 12.12.1.2 priority 10
!
```

show configuration

Description

Shows current policy-map configuration

Parent Commands (Modes)

class <fbcs-class-name> [priority <fbcs-flow-priority>]

Syntax

show configuration

Usage Guidelines

Use this command to display running configurations within current policy-map.

Examples

```
sonic(config-policy-map-flow)# show configuration
!
policy-map copp-test-policy type copp
  description "copp-policy-test"
  class copp-system-vrrp priority 0
    set copp-action copp-system-vrrp
!
```

show configuration

Description

Shows current policy-map configuration

Parent Commands (Modes)

class <fbcs-class-name> [priority <fbcs-flow-priority>]

Syntax

show configuration

Usage Guidelines

Use this command to display running configurations within current policy-map.

Examples

```
sonic(config-policy-map-flow)# show configuration
!
policy-map acl-copp-test-policy type acl-copp
description "acl-copp test policy"
class class-acl-copp priority 1020
set trap-queue 1
!
```

show configuration

Description

Shows current pbf ip next-hop group configuration

Parent Commands (Modes)

```
pbf next-hop-group <fbn-nhgrp-name> [ type { ip | ipv6 } ]
```

Syntax

```
show configuration
```

Usage Guidelines

Use this command to display running configurations within current pbf ip next-hop group configuration

Examples

```
sonic(config)# pbf next-hop-group test type ip
sonic(config-pbf-ip-nh-group)# show configuration
!
pbf next-hop-group test type ip
```

show configuration

Description

Shows current pbf ipv6 next-hop group configuration

Parent Commands (Modes)

```
pbf next-hop-group <fbn-nhgrp-name> [ type { ip | ipv6 } ]
```

Syntax

```
show configuration
```

Usage Guidelines

Use this command to display running configurations within current pbf ipv6 next-hop group configuration

Examples

```
sonic(config)# pbf next-hop-group test type ipv6
sonic(config-pbf-ipv6-nh-group)# show configuration
!
pbf next-hop-group test type ipv6
```

show configuration

Description

Shows current pbf ip replication group configuration

Parent Commands (Modes)

```
pbf replication-group <fbn-replgrp-name> [ type { ip | ipv6 } ]
```

Syntax

```
show configuration
```

Usage Guidelines

Use this command to display running configurations within current pbf ip replication group configuration

Examples

```
sonic(config)# pbf replication-group test type ip
sonic(config-pbf-ip-repl-group)# show configuration
!
pbf replication-group test type ip
```

show configuration

Description

Shows current pbf ipv6 replication group configuration

Parent Commands (Modes)

pbf replication-group <fbn-replgrp-name> [type { ip | ipv6 }]

Syntax

show configuration

Usage Guidelines

Use this command to display running configurations within current pbf ipv6 replication group configuration

Examples

```
sonic(config)# pbf replication-group test type ipv6
sonic(config-pbf-ipv6-repl-group)# show configuration
!
pbf replication-group test type ipv6
```

show configuration

Description

Sh

Parent Commands (Modes)

link state track <grp-name>

Syntax

show configuration

Usage Guidelines

Use this command to display running configurations within current link state track group

Examples

```
sonic(config-link-track)# show configuration  
link state track track1  
description mlag  
downstream all-mLAG  
threshold type percentage up 10
```

show configuration

Description

```
show configuration
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
show configuration
```

show configuration

Description

```
show configuration
```

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
show configuration
```

show configuration

Description

```
show configuration
```

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
show configuration
```

show configuration

Description

```
show configuration
```

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
show configuration
```

show configuration

Description

```
show configuration
```

Parent Commands (Modes)

```
interface Management <mgmt-if-id>
```

Syntax

```
show configuration
```

show configuration

Description

```
show configuration
```

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
show configuration
```

show configuration

Description

```
show configuration
```

Parent Commands (Modes)

```
interface range create vlan_range_num  
interface range vlan_range_num
```

Syntax

```
show configuration
```

show configuration

Description

```
show configuration
```

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]  
interface range po_range_num
```

Syntax

```
show configuration
```

show configuration

Description

```
show configuration
```

Parent Commands (Modes)

```
interface CPU
```

Syntax

```
show configuration
```

show configuration

Description

```
Displays current NAT configuration
```

Parent Commands (Modes)

```
nat
```

Syntax

```
show configuration
```

Usage Guidelines

```
Use this command to display running configurations.
```

Features this CLI belongs to

- NAT

show configuration

Description

```
show tc-dscp-map configuration
```

Parent Commands (Modes)

```
qos map tc-dscp <name>
```

Syntax

```
show configuration
```

show configuration

Description

Display BGP configuration

Parent Commands (Modes)

address-family ipv4 unicast

Syntax

show configuration

show configuration

Description

Display BGP neighbor IPv4 configuration

Parent Commands (Modes)

address-family ipv4 unicast

Syntax

show configuration

show configuration

Description

show bgp peer group IPv4 configuration

Parent Commands (Modes)

address-family ipv4 unicast

Syntax

show configuration

show configuration

Description

```
Display scheduler-policy configuration
```

Parent Commands (Modes)

```
qos scheduler-policy <name>
```

Syntax

```
show configuration
```

show configuration

Description

```
Display scheduler_policy queue configuration
```

Syntax

```
show configuration
```

show configuration

Description

```
Display scheduler_policy port configuration
```

Syntax

```
show configuration
```

show configuration

Description

```
show bgp configuration
```

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
show configuration
```

show configuration

Description

```
show bgp neighbor 12vpn evpn configuration
```

Parent Commands (Modes)

```
address-family 12vpn evpn
```

Syntax

```
show configuration
```

show configuration

Description

```
show bgp peer group 12vpn evpn configuration
```

Parent Commands (Modes)

```
address-family 12vpn evpn
```

Syntax

```
show configuration
```

show configuration

Description

```
Show configuration
```

Parent Commands (Modes)

```
mclag domain <mclag-domain-id>
```

Syntax

```
show configuration
```

show configuration

Description

```
Displays current IP SLA configuration
```

Parent Commands (Modes)

```
ip sla <sla-id>
```

Syntax

```
show configuration
```

Usage Guidelines

```
Use this command to display running configurations.
```

Features this CLI belongs to

- IPSLA

show configuration

Description

```
Displays current IP SLA configuration
```

Parent Commands (Modes)

```
icmp-echo <addr>
```

Syntax

```
show configuration
```

Usage Guidelines

```
Use this command to display running configurations.
```

Features this CLI belongs to

- IPSLA

show configuration

Description

Displays current IP SLA configuration

Parent Commands (Modes)

tcp-connect <addr> port <portno>

Syntax

show configuration

Usage Guidelines

Use this command to display running configurations.

Features this CLI belongs to

- IPSLA

show configuration

Description

show network policy running configuration

Parent Commands (Modes)

network-policy profile <np_num>

Syntax

show configuration

show configuration

Description

Displays current DAS configuration

Parent Commands (Modes)

aaa server radius dynamic-author

Syntax

show configuration

Usage Guidelines

Use this command to DAS display running configurations.

Examples

```
sonic-cli(config)# aaa server radius dynamic-author
sonic-cli(config-radius-da)# show configuration
!
aaa server radius dynamic-author
auth-type session-key
port 8989
client 7.7.7.7 server-key M2FsdGVkX17LcQREyGJP/aDuWTi34jJLzdqNy9W5WxI= encrypted
ignore server-key
server-key U2FsdGVkX18LcQREyGJP/aDuWTi34jJLzdqNy9W5WxI= encrypted
sonic-cli(config-radius-da)#

```

show configuration

Description

Display tc-pg-map configuration

Parent Commands (Modes)

qos map tc-pg <name>

Syntax

show configuration

show configuration

Description

Displays Bidirectional Forwarding detection(BFD) configurations.

Parent Commands (Modes)

bfd

Syntax

show configuration

Examples

```
device# configure terminal
device(config)# bfd
device(conf-bfd)# show configuration
!
bfd
profile fast
  receive-interval 150
  transmit-interval 150

peer 192.168.2.1 interface Ethernet0
  detect-multiplier 5
  echo-interval 200
  echo-mode
  receive-interval 200
  transmit-interval 200
!
peer 192.168.2.1 multihop local-address 192.168.2.2
  detect-multiplier 4
  receive-interval 150
  transmit-interval 150
!
```

show configuration

Description

Displays Bidirectional Forwarding detection(BFD) profile configurations.

Parent Commands (Modes)

profile <profilename>

Syntax

show configuration

Examples

```
device# configure terminal
device(config)# bfd
device(conf-bfd-profile)#profile test
device(conf-bfd-profile)# show configuration
!
profile test
  detect-multiplier 4
  echo-interval 123
  echo-mode
  receive-interval 123
  shutdown
  transmit-interval 123
  minimum-ttl 250
  passive-mode
```

show configuration

Description

```
Display dot1p-tc-map configuration
```

Parent Commands (Modes)

```
qos map dot1p-tc <name>
```

Syntax

```
show configuration
```

show configuration

Description

```
show pfc-priority-queue-map configuration
```

Parent Commands (Modes)

```
qos map pfc-priority-queue <name>
```

Syntax

```
show configuration
```

show configuration

Description

```
show configuration
```

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
show configuration
```

show configuration

Description

```
Display BGP configuration
```

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
show configuration
```

show configuration

Description

```
Display BGP neighbor IPv6 configuration
```

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
show configuration
```

show configuration

Description

```
Display BGP peer group IPv6 configuration
```

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
show configuration
```

show configuration

Description

```
Displays current OSPFv2 router configuration
```

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
show configuration
```

Usage Guidelines

```
Use this command to display running configurations within current OSPFv2 router.
```

Examples

```
sonic-clia(config-router-ospf)# show configuration
!
router ospf
  ospf router-id 10.1.1.1
  network 10.10.3.0/24 area 0.0.0.1
  network 10.10.4.0/24 area 0.0.0.1
```

Features this CLI belongs to

- OSPFv2

show configuration

Description

```
show tc-dot1p-map configuration
```

Parent Commands (Modes)

```
qos map tc-dot1p <name>
```

Syntax

```
show configuration
```

show configuration

Description

```
Display route-map configuration
```

Parent Commands (Modes)

```
route-map <route-map-name> { permit | deny } <seq-nu>
```

Syntax

```
show configuration
```

show configuration

Description

```
Current configuration
```

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
show configuration
```

show configuration

Description

Current configuration

Parent Commands (Modes)

vrrp ipv6

Syntax

show configuration

show configuration

Description

Current configuration

Parent Commands (Modes)

vrrp ipv4

Syntax

show configuration

show configuration

Description

Current configuration

Parent Commands (Modes)

vrrp ipv6

Syntax

show configuration

show configuration

Description

Current configuration

Parent Commands (Modes)

vrrp ipv4

Syntax

show configuration

show configuration

Description

Current configuration

Parent Commands (Modes)

vrrp ipv6

Syntax

show configuration

show configuration

Description

Current configuration

Parent Commands (Modes)

vrrp ipv4

Syntax

show configuration

show configuration

Description

Current configuration

Parent Commands (Modes)

vrrp ipv6

Syntax

show configuration

show configuration

Description

Displays current switch-resource configuration

Parent Commands (Modes)

switch-resource

Syntax

show configuration

Usage Guidelines

Use this command to display running configurations.

Features this CLI belongs to

- Switch-resource

show configuration

Description

This command is used to show current TAM configuration.

Parent Commands (Modes)

tam

Syntax

```
show configuration
```

Usage Guidelines

Use this command to display TAM running configuration.

Examples

```
sonic-cli(config-tam)# show configuration
!
!
tam
  switch-id 3232
  enterprise-id 434
  collector c1 ip 1.1.1.1 port 1111 protocol UDP
  sampler s1 rate 1
  sampler s2 rate 655
  sampler s4 rate 65550
  sampler s5 rate 999999999
  flow-group f1 src-ip 10.1.1.10/24 dst-ip 30.1.1.10/24 protocol TCP 14-src-port 8080 priority
    100
  flow-group f2 src-ip 10.1.1.10/32 dst-ip 30.1.1.10/32 protocol UDP priority 100
!
drop-monitor
  aging-interval 23
!
ifa
  session ifa1 flowgroup f1 collector c1 node-type EGRESS
```

show configuration

Description

This command is used to show Telemetry Destination Group configuration.

Parent Commands (Modes)

```
destination-group <dg-id>
```

Syntax

```
show configuration
```

Usage Guidelines

Use this command to display Telemetry Destination Group running configuration.

Examples

```
sonic-cli(configure-telemetry-dg)# show configuration
!
!
destination-group dg1
    ip address 1.1.1.1 port 1111 src-ip 192.168.122.11 vrf mgmt
```

show configuration

Description

This command is used to show current IFA configuration.

Parent Commands (Modes)

```
ifa
```

Syntax

```
show configuration
```

Usage Guidelines

Use this command to display IFA running configuration.

Examples

```
sonic-cli(config-tam-ifaf)# show configuration
!
!
ifa
    session ifa1 flowgroup f1 collector c1 node-type EGRESS
```

show configuration

Description

This command is used to show current drop monitor configuration.

Parent Commands (Modes)

```
drop-monitor
```

Syntax

```
show configuration
```

Usage Guidelines

Use this command to display drop monitor running configuration.

Examples

```
sonic-cli(config-tam-dm)# show configuration
!
!
drop-monitor
  aging-interval 23
```

show configuration

Description

This command is used to show current Tailstamping configuration.

Parent Commands (Modes)

```
tail-stamping
```

Syntax

```
show configuration
```

Usage Guidelines

Use this command to display Tailstamping running configuration.

Examples

```
sonic-cli(config-tam-ts)# show configuration
!
!
tail-stamping
  session ss1 flowgroup f1 node-type normal
```

show configuration

Description

Display BGP configuration

Parent Commands (Modes)

router bgp <as-num-dot> { [vrf <vrf-name>] }

Syntax

show configuration

show configuration

Description

Display BGP nbr configuration

Parent Commands (Modes)

neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }

Syntax

show configuration

show configuration

Description

Display BGP peer group configuration

Parent Commands (Modes)

peer-group <template-str>

Syntax

show configuration

show configuration

Description

```
show configuration
```

Parent Commands (Modes)

```
mirror-session <session-name>
```

Syntax

```
show configuration
```

show configuration

Description

```
Display dscp-tc-map configuration
```

Parent Commands (Modes)

```
qos map dscp-tc <name>
```

Syntax

```
show configuration
```

show configuration

Description

```
Display tc-queue-map configuration
```

Parent Commands (Modes)

```
qos map tc-queue <name>
```

Syntax

```
show configuration
```

show configuration

Description

```
Show VXLAN configuration
```

Parent Commands (Modes)

```
interface vxlan <vxlan-if-name>
```

Syntax

```
show configuration
```

show configuration

Description

```
Display wred-policy configuration
```

Parent Commands (Modes)

```
qos wred-policy <name>
```

Syntax

```
show configuration
```

show core config

Description

```
Display the coredump configuration. Use this command to display if the coredump feature is administratively enabled or disabled.
```

Syntax

```
show core config
```

Usage Guidelines

```
sonic# show core config
```

Examples

```
sonic# show core config  
Coredump : Enabled
```

Features this CLI belongs to

- COREDUMP

Alternate command

```
show cores config
```

show core info

Description

Use this command to display detailed information about a crash that has occurred in the system.
This command takes processid or executable name as input to search and display the corresponding crash information. If multiple core files are found which satisfy the match condition, information of all core files is displayed.

The following information about matching core files is displayed:

- Time: The time of the crash, as reported by the kernel in UTC
- Executable: The full path to the application executable that has crashed"
- Core File: The file name of the application core dump of the executable that has crashed
- PID: The identifier of the process that crashed
- User ID: The user identifier of the process that crashed
- Group ID: The group identifier of the process that crashed
- Signal: The signal that caused the process to crash, when applicable
- Command Line: The command line arguments of the process that crashed
- Boot ID: The unique identifier of the local system that is generated and set on each system boot up event
- Machine ID: The unique machine identifier of the local system that is set during installation
- Core File Found: Indicates whether the captured core file exists on local disk or has been removed
- Crash Message: A copy of the application stack trace information of the process crashed

Syntax

```
show core info <key>
```

Parameters

Name	Description	Type
key	String	String

Usage Guidelines

```
sonic# show core info clish
```

Examples

Features this CLI belongs to

- COREDUMP

Alternate command

show cores info

show core list

Description

Use this command to list a summary of the core files generated by the kernel. The following information about each core file is also displayed.

- TIME The time of the crash, as reported by the kernel in UTC
- PID: The identifier of the process that crashed

- SIG: The signal that caused the process to crash, when applicable
- COREFILE: Indicates whether the captured core file exists on local disk or has been removed
- EXE: The application executable that has crashed

Syntax

```
show core list
```

Usage Guidelines

```
sonic# show core list
```

Examples

```
sonic# show core list
      TIME          PID SIG COREFILE EXE
2020-05-16 11:54:33    26480 11 present clish
2020-05-15 01:25:16     6195 11 present crashme
2020-05-15 00:45:28    13604 11 present crashme
2020-05-14 02:11:11     3197 11 present crashme
2020-05-13 01:10:56    17844 11 missing crashme
2020-05-13 01:10:55    17728 11 present crashme
```

Features this CLI belongs to

- COREDUMP

Alternate command

```
show cores list
```

show counters configuration

Description

```
Display counters configuration
```

Syntax

```
show counters configuration
```

show crm

Description

Display CRM information

Syntax

```
show crm { summary | { resources { { acl { group | table } } | all | dnat | fdb | ipmc | { ipv4
    { neighbor | nexthop | route } } | { ipv6 { neighbor | nexthop | route } } | { nexthop { {
        group { member | object } } } | snat } } | { thresholds { { acl { group | table } } | all
    | dnat | fdb | ipmc | { ipv4 { neighbor | nexthop | route } } | { ipv6 { neighbor | nexthop
    | route } } | { nexthop { { group { member | object } } } } | snat } } }
```

show database map

Description

Use this command to display a summary of the databases currently in use in SONiC. The following information

about each database is also displayed:

- ID: Numeric database identifier.
- Name: Database name string used to refer to the database in the sonic-db-cli command and Database connector APIs.
- Instance: Redis instance this database is part of
- TCP Port: The TCP Port used to connect to the Redis instance which this database is part of.
- Unix Socket Path: The unix socket path used to connect to the Redis instance which this database is part of.

Syntax

```
show database map
```

Usage Guidelines

```
sonic# show database map
```

Examples

```
sonic# show database map
```

ID	Name	Instance	TCP Port	Unix Socket Path
0	APPL_DB	redis2	26379	/var/run/redis/redis2.sock
1	ASIC_DB	redis3	36379	/var/run/redis/redis3.sock
2	COUNTERS_DB	redis	6379	/var/run/redis/redis.sock
3	LOGLEVEL_DB	redis	6379	/var/run/redis/redis.sock
4	CONFIG_DB	redis	6379	/var/run/redis/redis.sock
5	PFC_WD_DB	redis	6379	/var/run/redis/redis.sock
6	STATE_DB	redis	6379	/var/run/redis/redis.sock
7	SNMP_OVERLAY_DB	redis	6379	/var/run/redis/redis.sock
8	ERROR_DB	redis	6379	/var/run/redis/redis.sock

show dot1x

Description

Display dot1x information.

Syntax

```
show dot1x
```

show dot1x detail

Description

Display the details of the configuration for the specified port.

Syntax

```
show dot1x detail { all | <port> }
```

Parameters

Name	Description	Type
port	EthernetNUM	

show dropcounters capabilities

Description

show command to display dropcounters capabilities

Syntax

```
show dropcounters capabilities
```

Usage Guidelines

```
sonic# show dropcounters capabilities
```

Examples

```

-
sonic# show dropcounters capabilities
Counter Type          Total
-----
PORT_INGRESS_DROPS      3

PORT_INGRESS_DROPS
IP_HEADER_ERROR
FDB_AND_BLACKHOLE_DISCARDS

```

show dropcounters configuration

Description

show command to display dropcounters configuration

Syntax

```
show dropcounters configuration
```

Usage Guidelines

```
sonic# show dropcounters configuration
```

Examples

```

sonic# show dropcounters configuration
Counter   Alias    Group   Type           Mirror      Reasons
-----  -----  -----
DEBUG_0   RX_LEGIT  LEGIT   PORT_INGRESS_DROPS  Session1  SMAC_EQUALS_DMAC
DEBUG_1   TX_LEGIT  None    SWITCH_EGRESS_DROPS NA          INGRESS_VLAN_FILTER
                                                EGRESS_VLAN_FILTER

```

show dropcounters configuration detail

Description

Display detailed dropcounters configuration

Syntax

```
show dropcounters configuration detail
```

show errdisable link-flap

Description

Shows error disable recovery information incase of excessive link flaps.

Syntax

```
show errdisable link-flap
```

Usage Guidelines

Use this command to check status of error disable recovery for excessive link flaps.

Examples

```
sonic#show errdisable link-flap
Interface  Flap-threshold  Sampling-time  Recovery-timeout Status
-----
Ethernet0  10            3              30           Err-disabled
Ethernet4  10            3              60           On
```

Features this CLI belongs to

- ERRDISABLE

show errdisable recovery

Description

Shows error disable recovery information

Syntax

```
show errdisable recovery
```

Usage Guidelines

Use this command to check status of error disable recovery for all supported feature

Examples

```
show errdisable recovery
Errdisable Cause      Status
-----
udld                  enabled
bpduguard            disabled
Timeout for Auto-recovery: 300 seconds
```

Features this CLI belongs to

- ERRDISABLE

show event details

Description

```
Detailed view of events
```

Syntax

```
show event details
```

show event id

Description

```
Display event
```

Syntax

```
show event id <id>
```

Parameters

Name	Description	Type
id	String	String

show event profile

Description

```
Display active Event Profile
```

Syntax

```
show event profile
```

show event recent

Description

```
Event recent
```

Syntax

```
show event recent <interval>
```

Parameters

Name	Description	Type
interval	Select [5min(5min) 60min(60min) 24hr(24hr)]	

show event severity

Description

```
Display all events with a given severity
```

Syntax

```
show event severity <sev>
```

Parameters

Name	Description	Type
sev	Select [critical(critical) major(major) minor(minor) warning(warning) informational(informational)]	

show event summary

Description

Display event summary

Syntax

```
show event summary
```

show evpn

Description

Display EVPN information summary

Syntax

```
show evpn
```

show evpn arp-cache vni

Description

VxLAN Network Identifier

Syntax

```
show evpn arp-cache vni <vninum> { { [ ip <ipvalue> ] } | [ duplicate ] | { [ vtep <vtепvalue> ] } }
```

Parameters

Name	Description	Type
vninum	VNI	Integer
ipvalue	A.B.C.D/A::B	String
vtepvalue	A.B.C.D	String

show evpn arp-cache vni all

Description

Display all VNIs

Syntax

```
show evpn arp-cache vni all { [ detail ] | [ duplicate ] } ]
```

show evpn mac vni

Description

```
VxLAN Network Identifier
```

Syntax

```
show evpn mac vni <vninum> { { [ mac <macvalue> ] } | [ duplicate ] | { [ vtep <vtepvalue> ] } }
```

Parameters

Name	Description	Type
vninum	VNI	Integer
macvalue	nn:nn:nn:nn:nn:nn	String
vtepvalue	A.B.C.D	String

show evpn mac vni all

Description

```
Display all VNIs
```

Syntax

```
show evpn mac vni all { [ detail ] | [ duplicate ] | { [ vtep <vtepvalue> ] } } ]
```

Parameters

Name	Description	Type
vtepvalue	A.B.C.D	String

```
show evpn next-hops vni
```

Description

VxLAN Network Identifier

Syntax

```
show evpn next-hops vni <vninum> { [ ip <ipvalue> ] }
```

Parameters

Name	Description	Type
vninum	VNI	Integer
ipvalue	A.B.C.D/A::B	String

```
show evpn next-hops vni all
```

Description

Display all VNIs

Syntax

```
show evpn next-hops vni all
```

```
show evpn rmac vni
```

Description

VxLAN Network Identifier

Syntax

```
show evpn rmac vni <vninum> { [ mac <macvalue> ] }
```

Parameters

Name	Description	Type
vninum	VNI	Integer
macvalue	nn:nn:nn:nn:nn:nn	String

show evpn rmac vni all

Description

All VNIs

Syntax

```
show evpn rmac vni all
```

show evpn vni

Description

Display VxLAN Network Identifier

Syntax

```
show evpn vni <vninum>
```

Parameters

Name	Description	Type
vni	VNI	Integer

show evpn vni detail

Description

Display detailed information on each VNI

Syntax

```
show evpn vni detail
```

show grpc

Description

Displays GRPC configuration.

Syntax

```
show grpc
```

Usage Guidelines

```
sonic-cl# show grpc
```

Examples

```
sonic-cl(config)# show grpc
-----
GRPC Global Configuration
-----
Port: 22
```

show histogram memory

Description

Display histogram information

Syntax

```
show histogram memory { system | docker | process } [ verbose ] [ stime <_stime> ] [ etime <_etime> ] [ filter <filtername> ] [ analyze <_analyze> ]
```

Parameters

Name	Description	Type
_stime	Time in ISO format	String
_etime	Time in ISO format	String
filtername	WORD	String
_analyze	WORD	String

show hosts

Description

Display IP name servers

Syntax

`show hosts`

show in-memory-logging

Description

Display in-memory-logging information

Syntax

`show in-memory-logging`

show in-memory-logging count

Description

Total number of in-memory-logging

Syntax

`show in-memory-logging count`

show in-memory-logging lines

Description

Output of last NUM lines

Syntax

`show in-memory-logging lines [<lines>]`

Parameters

Name	Description	Type
lines	port	Integer

show interface

Description

Display interface information

Syntax

```
show interface { { advertise [ <phy-if-range> ] } | { fec { status [ <phy-if-range> ] } } } | { link-training [ <phy-if-range> ] } | { unreliable-los { status [ <phy-if-range> ] } } } | { counters { [ rate ] | [ <phy-if-name> ] | [ <po-name> ] | { [ rif { [ <phy-if-name> ] | [ <po-name> ] | [ <vlan-name> ] | [ <phy-sub-name> ] | [ <po-sub-name> ] ] ] } } } } | { Eth [ <iface_num> ] } | { Ethernet [ <iface_num> ] } | <iface_range_num> | { PortChannel [ <po-id> ] } | { Management [ <mgmt-if-id> ] } | <vlan_range_num> | { Loopback [ <lo-id> ] } | { status [ <if-reason> ] } | { description [ <iftype> ] } | { phy { { [ status [ <phy-if-name> ] ] } | { [ counters [ <phy-if-name> ] ] } } } | { dropcounters { [ <phy-if-name> ] ] } } | { transceiver [ <phy-if-name> ] { [ dom ] | { [ diagnostics { capability | status } ] } } } [ summary ] [ table ] } | debounce }
```

Parameters

Name	Description	Type
phy-if-range	Interface Ethernet Range	
phy-if-name	EthernetNUM	
po-name	Interface PortChannel Range	
vlan-name	Interface Vlan Range	
phy-sub-name	EthernetX.Y	
po-sub-name	Interface Type	
iface_num	EthernetSLOTPORT	
iface_range_num	Interface Ethernet Range	
po-id	PortChannel Range	
mgmt-if-id		Integer
vlan_range_num	Interface Vlan Range	
lo-id		Integer
if-reason	Interface status reason	Select [admin-down(admin_down) err-disabled(err_disabled) phy-link-down(phy_link_down) oper-up(oper_up) all-links-down(all_links_down) lacp-fail(lacp_fail) min-links(min_links)]

Name	Description	Type
iftype	Interface Type	

show interface breakout

Description

Show information related to dynamic port breakout.

Syntax

```
show interface breakout { { [ dependencies { port <slotport> } ] } | [ modes ] | { [ port
<slotport> ] } | [ resources ] }
```

Parameters

Name	Description	Type
slotport	Front-panel port (slot/port)	

Usage Guidelines

Use this command to show,

- the dependent configurations on a port.
`show interface breakout dependencies port front-panel-port`
- breakout modes supported by the device.
`show interface breakout modes`
- breakout resource limits supported by the device and it's current usage.
`show interface breakout resources`
- the current breakout configuration, member ports and status.
`show interface breakout port front-panel-port`
`show interface breakout port`

Examples

```
sonic# show interface breakout modes
```

Port	Interface	Supported Modes	Default Mode
1/1	Ethernet0	1x100G, 1x40G, 4x25G, 4x10G	1x100G
1/3	Ethernet8	1x100G, 1x40G, 4x25G, 4x10G	1x100G
1/5	Ethernet16	1x100G, 1x40G, 4x25G, 4x10G	1x100G
1/7	Ethernet24	1x100G, 1x40G, 4x25G, 4x10G	1x100G
1/9	Ethernet32	1x100G, 1x40G, 4x25G, 4x10G	1x100G
1/11	Ethernet40	1x100G, 1x40G, 4x25G, 4x10G	1x100G

```

1/13  Ethernet48  1x100G, 1x40G, 4x25G, 4x10G      1x100G
1/15  Ethernet56  1x100G, 1x40G, 4x25G, 4x10G      1x100G
1/17  Ethernet64  1x100G, 1x40G, 4x25G, 4x10G      1x100G
1/19  Ethernet72  1x100G, 1x40G, 4x25G, 4x10G      1x100G
1/21  Ethernet80  1x100G, 1x40G, 4x25G, 4x10G      1x100G
1/23  Ethernet88  1x100G, 1x40G, 4x25G, 4x10G      1x100G
1/25  Ethernet96  1x100G, 1x40G, 4x25G, 4x10G      1x100G
1/27  Ethernet104 1x100G, 1x40G, 4x25G, 4x10G      1x100G
1/29  Ethernet112 1x100G, 1x40G, 4x25G, 4x10G      1x100G
1/31  Ethernet120 1x100G, 1x40G, 4x25G, 4x10G      1x100G
sonic#
sonic# show interface breakout dependencies port 1/1
-----
Dependent Configurations
-----
VLAN|Vlan100
VLAN_MEMBER|Vlan100|Ethernet2
sonic#
sonic# show interface breakout port 1/1
-----
Port  Breakout Mode  Status        Interfaces
-----
1/1   4x25G       Completed      Ethernet0
                           Ethernet1
                           Ethernet2
                           Ethernet3
sonic#
sonic# show interface breakout
-----
Port  Breakout Mode  Status        Interfaces
-----
1/1   4x25G       Completed      Ethernet0
                           Ethernet1
                           Ethernet2
                           Ethernet3
1/3   4x10G       Completed      Ethernet8
                           Ethernet9
                           Ethernet10
                           Ethernet11
sonic#

```

show ip access-group

Description

Display IPv4 ACL binding summary

Syntax

```
show ip access-group
```

Examples

```
Ingress IP access-list ipacl-example on PortChannel1
```

Alternate command

```
admin@sonic:~$ show acl table [name]
```

show ip access-lists

Description

```
Show IPv4 ACL rules and statistics
```

Syntax

```
show ip access-lists [ <access-list-name> { { [ interface { Ethernet | <PortChannel> | <Vlan> | <eth-sub-if-id> | <po-sub-if-id> } ] } | [ Switch ] ] }
```

Parameters

Name	Description	Type
access-list-name	WORD	String
PortChannel	PortChannelNUM	
Vlan	VlanNUM	
eth-sub-if-id	EthernetX.Y	
po-sub-if-id	PortChannelNUM	

Usage Guidelines

ACL name and interface names are optional. If ACL name is not specified then all IPv4 ACLs will be displayed. ACL statistics will be shown only if the ACL is applied globally or to any interface.

Examples

```
ip access-list ipacl-example
  seq 10 permit ip host 10.1.1.1 host 20.1.1.1 (0 packets) [0 bytes]
  seq 20 permit ip host 10.1.1.2 host 20.1.1.2 (0 packets) [0 bytes]
  seq 30 permit ip host 10.1.1.3 host 20.1.1.3 (0 packets) [0 bytes]
  seq 40 permit ip host 10.1.1.4 host 20.1.1.4 (0 packets) [0 bytes]
```

Alternate command

```
admin@sonic:~$ show acl rule [name] [rule_name]
```

show ip arp

Description

This command displays ARP table entries. To filter the output, specify an interface, port channel, or VLAN, an IP address, a MAC address, or a combination of more than one value to match. You can also display total number of ARP entries using 'summary' option.

Syntax

```
show ip arp [ vrf { <vrfname> | mgmt | all } ] { [ <ip-addr> ] | { [ mac-address <mac-addr> ] } | [ summary ] }
```

Parameters

Name	Description	Type
vrfname	VRF name (prefixed by Vrf, Max: 15 characters)	String
ip-addr	A.B.C.D	String
mac-addr	nn:nn:nn:nn:nn:nn	String

Usage Guidelines

```
sonic# show ip arp [interface { Ethernet < port > [summary] | PortChannel < id > [summary] | Vlan < id > [summary] }] [< ipv4-address >] [mac-address < mac >] [summary]
```

Examples

```
sonic# show ip arp
```

Address	Hardware address	Interface	Egress Interface
192.168.1.4	00:01:02:03:44:55	Ethernet8	-
192.168.2.4	00:01:02:03:ab:cd	PortChannel1200	-
192.168.3.6	00:01:02:03:04:05	Vlan100	Ethernet4
10.11.48.254	00:01:e8:8b:44:71	eth0	-
10.14.8.102	00:01:e8:8b:44:71	eth0	-

```

sonic# show ip arp interface Vlan 100
-----
Address      Hardware address   Interface      Egress Interface
-----
192.168.3.6  00:01:02:03:04:05  Vlan100        Ethernet4

sonic# show ip arp interface Management 0
-----
Address      Hardware address   Interface      Egress Interface
-----
10.11.48.254 00:01:e8:8b:44:71  eth0          -
10.14.8.102   00:01:e8:8b:44:71  eth0          -

```

```

sonic# show ip arp 192.168.1.4
-----
Address      Hardware address   Interface      Egress Interface
-----
192.168.1.4  00:01:02:03:44:55  Ethernet8      -

```

```

sonic# show ip arp mac-address 00:01:02:03:ab:cd
-----
Address      Hardware address   Interface      Egress Interface
-----
192.168.2.4  00:01:02:03:ab:cd  PortChannel1200  -

```

show ip arp interface

Description

ARP entries for this interface

Syntax

```

show ip arp interface { { <phy-if-name> [ summary ] } | { po-if-name [ summary ] } | {
    <phy-subif-name> [ summary ] } | { Loopback { <lo-id> [ summary ] } } | { Management {
    <mgmt-if-id> [ summary ] } } | { Vlan { <vlan-id> [ summary ] } } | { Vxlan {
    <vxlan-if-name> [ summary ] } }

```

Parameters

Name	Description	Type
phy-if-name	EthernetNUM	
phy-subif-name	Interface Type	
lo-id		Integer
mgmt-if-id		Integer
vlan-id		Integer

Name	Description	Type
vxlan-if-name	WORD	String

show ip dhcp smart-relay

Description

Displays general information about ipv4 DHCP smart-relay

Syntax

```
show ip dhcp smart-relay
```

Usage Guidelines

```
show ip dhcp smart-relay
```

Examples

```
console#show ip dhcp smart-relay
DHCP smart-relay is Enabled
DHCP smart-relay is enabled on the following VLANs: 20
```

Features this CLI belongs to

- DHCP smart-relay

show ip dhcp snooping

Description

Displays general information about ipv4 DHCP Snooping

Syntax

```
show ip dhcp snooping
```

Usage Guidelines

```
show ip dhcp snooping
```

Examples

```
console#show ip dhcp snooping

  snooping is Enabled
  snooping source MAC verification is disabled
  snooping is enabled on the following VLANs: 10,20

rface   Trusted
-----
rnet1   Yes
console#
```

Features this CLI belongs to

- DHCP Snooping

show ip dhcp snooping binding

Description

Show DHCPv4 Snooping binding database

Syntax

```
show ip dhcp snooping binding
```

show ip dhcp snooping statistics

Description

Show DHCPv4 Snooping statistics

Syntax

```
show ip dhcp snooping statistics
```

show ip dhcp snooping statistics detail

Description

Show DHCPv4 Snooping deatiled statistics

Syntax

```
show ip dhcp snooping statistics detail
```

show ip dhcp-relay

Description

```
Display IP DHCP relay information
```

Syntax

```
show ip dhcp-relay { [ brief ] | { [ detailed { [ <intfName1> ] | [ <pchName1> ] | [ <vlanName1> ] } ] } | { [ statistics { [ <intfName> ] | [ <pchName> ] | [ <vlanName> ] } ] }
```

Parameters

Name	Description	Type
intfName1	EthernetX.Y	
pchName1	PortChannelNUM	
vlanName1	VlanNUM	
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	

show ip forward-protocol

Description

```
Displays IP helper global information.
```

Syntax

```
show ip forward-protocol
```

Usage Guidelines

```
show ip forward-protocol
```

Examples

```
sonic# show ip forward-protocol
UDP forwarding : Enabled
UDP rate limit : 2000 pps
UDP forwarding enabled on the ports: TFTP, DNS, NTP, TACACS, 12200, 12202
UDP forwarding disabled on the ports: NetBios-Name-Server, NetBios-Datagram-Server
sonic#
```

Features this CLI belongs to

- IP Helper

Alternate command

```
show ip forward_protocol config
```

show ip helper-address

Description

Displays IP helper server addresses configured on interface.

Syntax

```
show ip helper-address [ <iface> ]
```

Parameters

Name	Description	Type
iface	Interface Type - Ranges	

Usage Guidelines

```
show ip helper-address [ <interface-name> ]
```

Examples

```
sonic# show ip helper-address
Interface      Vrf      Relay Address
-----
Vlan200          4.4.4.4
Ethernet0        1.1.1.1
                  Vrf1    2.2.2.2
Vlan100          3.3.3.3
Ethernet4        5.5.5.5
                  7.7.7.7
sonic#
```

```
sonic# show ip helper-address Ethernet0
Interface      Vrf      Relay Address
-----
Ethernet0      Vrf1    2.2.2.2
                  1.1.1.1
sonic#
```

Features this CLI belongs to

- IP Helper

Alternate command

```
show ip helper_address config
```

show ip helper-address statistics

Description

Displays IP helper packet counters and statistics on interface.

Syntax

```
show ip helper-address statistics [ <iface> ]
```

Parameters

Name	Description	Type
iface	Interface Type - Ranges	

Usage Guidelines

```
show ip helper-address statistics [ <interface-name> ]
```

Examples

```
sonic# show ip helper-address statistics
Ethernet0
-----
    Packets received          : 0
    Packets relayed           : 24
    Packets dropped           : 0
    Invalid TTL packets       : 0
    All ones broadcast packets received : 0
    Net directed broadcast packets received : 0
Vlan100
-----
    Packets received          : 45678
    Packets relayed           : 0
    Packets dropped           : 0
    Invalid TTL packets       : 0
    All ones broadcast packets received : 0
    Net directed broadcast packets received : 0
Ethernet4
-----
    Packets received          : 0
    Packets relayed           : 0
    Packets dropped           : 24444
    Invalid TTL packets       : 0
    All ones broadcast packets received : 0
    Net directed broadcast packets received : 0
sonic#
```

```
sonic# show ip helper-address statistics Vlan100
    Packets received          : 45678
    Packets relayed           : 0
    Packets dropped           : 0
    Invalid TTL packets       : 0
    All ones broadcast packets received : 0
    Net directed broadcast packets received : 0
sonic#
```

Features this CLI belongs to

- IP Helper

Alternate command

```
show ip helper_address statistics
```

show ip host-access-lists

Description

Show IPv4 host ACL rules

Syntax

```
show ip host-access-lists [ <access-list-name> { { [ interface { Ethernet | <PortChannel> | <Vlan> | <eth-sub-if-id> | <po-sub-if-id> } ] } | [ Switch ] ] }
```

Parameters

Name	Description	Type
access-list-name	WORD	String
PortChannel	PortChannelNUM	
Vlan	VlanNUM	
eth-sub-if-id	EthernetX.Y	
po-sub-if-id	PortChannelNUM	

Usage Guidelines

Host ACL name and interface names are optional. If host ACL name is not specified then all IPv4 host ACLs will be displayed.

Examples

```
ip host-access-list ipacl-example
  seq 10 permit ip host 10.1.1.1
  seq 20 permit ip host 10.1.1.2
  seq 30 permit ip host 10.1.1.3 aeth-syndrome 0x63/0x63
  seq 40 permit ip host 10.1.1.4 aeth-syndrome 0x63/0x63
```

show ip interfaces

Description

IP information of interfaces

Syntax

```
show ip interfaces
```

show ip load-share

Description

Displays load share information

Syntax

```
show ip load-share
```

Usage Guidelines

Use this command to display ip load-share information

Examples

```
sonic-cl# show ip load-share
```

Features this CLI belongs to

- HASH

show ip ospf

Description

Display show related ospf information for a specific vrf

Syntax

```
show ip ospf [ vrf { <vrf-name> } ] { [ border-routers ] | { [ database { { [ asbr-summary [ <lsid> ] { { [ adv-router <advrouter> ] } | [ self-originate ] ] } ] } | { [ external [ <lsid> ] { { [ adv-router <advrouter> ] } | [ self-originate ] ] } ] } | [ max-age ] | { [ network [ <lsid> ] { { [ adv-router <advrouter> ] } | [ self-originate ] ] } ] } | { [ nssa-external [ <lsid> ] { { [ adv-router <advrouter> ] } | [ self-originate ] ] } ] } | { [ opaque-area [ <lsid> ] { { [ adv-router <advrouter> ] } | [ self-originate ] ] } ] } | { [ opaque-as [ <lsid> ] { { [ adv-router <advrouter> ] } | [ self-originate ] ] } ] } | { [ opaque-link [ <lsid> ] { { [ adv-router <advrouter> ] } | [ self-originate ] ] } ] } | { [ router [ <lsid> ] { { [ adv-router <advrouter> ] } | [ self-originate ] ] } ] } | { [ self-originate ] | { [ summary [ <lsid> ] { { [ adv-router <advrouter> ] } | [ self-originate ] ] } ] } | { [ interface [ traffic ] [ <interfacename> ] ] } ] } | { [ route ] | { [ neighbor { [ <neighip> ] | [ all ] | [ <interfacename> ] ] } [ detail ] ] } | { [ graceful-restart { helper { [ detail ] ] } ] } ] }
```

Parameters

Name	Description	Type
vrf-name	WORD	String
lsid	A.B.C.D	String
advrouter	A.B.C.D	String
interfacename	Interface Type - Ranges	
neighip	A.B.C.D	String

Usage Guidelines

Use this command to display global, neighbors, route, interfaces etc information related to an OSPFv2 router. User can optionally specify VRF on which the router have to be configured. If VRF name is not specified then the command is considered for VRF default.

Technical details on OSPFv2 support is also available at
<http://docs.frrouting.org/en/latest/>

Examples

```
device# show ip ospf
OSPF Routing Process, Router ID: 1.1.1.1
Supports only single TOS (TOS0) routes
This implementation conforms to RFC2328
RFC1583Compatibility flag is enabled
OpaqueCapability flag is disabled
Initial SPF scheduling delay 0 millisec(s)
Minimum hold time between consecutive SPFs 50 millisec(s)
Maximum hold time between consecutive SPFs 5000 millisec(s)
Hold time multiplier is currently 1
time is 92031756
SPF algorithm last executed 1065d4h22m ago
Last SPF duration 0.0s
SPF timer is inactive
LSA minimum interval 5000 msecs
LSA minimum arrival 1000 msecs
Write Multiplier set to 20
Refresh timer 10 secs
Number of external LSA 0. Checksum Sum 0x0
Number of opaque AS LSA 0. Checksum Sum 0x0
Number of areas attached to this router: 2
Area ID: 0.0.0.0 (Backbone)
    Number of interfaces in this area: Total: 1 , Active: 1
    Number of fully adjacent neighbors in this area: 1
    Area has no authentication
    SPF algorithm executed 8 times
    Number of LSA 3
    Number of router LSA 2. Checksum Sum 0x40f64b4000000000
    Number of network LSA 1. Checksum Sum 0x40d5adc000000000
    Number of summary LSA 0. Checksum Sum 0x0
    Number of ASBR summary LSA 0. Checksum Sum 0x0
    Number of NSSA LSA 0. Checksum Sum 0x0
```

```

Number of opaque link LSA . Checksum Sum 0x
Number of opaque area LSA 0. Checksum Sum 0x0
Area ID: 0.0.0.1
Number of interfaces in this area: Total: 1 , Active: 1
Number of fully adjacent neighbors in this area: 0
Area has no authentication
SPF algorithm executed 1 times
Number of LSA 2
Number of router LSA 0. Checksum Sum 0x0
Number of network LSA 0. Checksum Sum 0x0
Number of summary LSA 2. Checksum Sum 0x40f1f61000000000
Number of ASBR summary LSA 0. Checksum Sum 0x0
Number of NSSA LSA 0. Checksum Sum 0x0
Number of opaque link LSA . Checksum Sum 0x
Number of opaque area LSA 0. Checksum Sum 0x0

```

```
sonic# show ip ospf neighbor | no-more
```

Neighbor ID DBsmL	Pri State	Dead Time Address	Interface	RXmtL	RqstL
10.59.142.247 0	1 Full/Backup	37.343s 64.1.1.2	Ethernet64:64.1.1.1	0	0

```
Leaf1# show ip ospf neighbor Ethernet66 | no-more
```

Neighbor ID DBsmL	Pri State	Dead Time Address	Interface	RXmtL	RqstL
2.2.2.2 0	1 Full/Backup	38.245s 64.1.1.2	Ethernet66:64.1.1.1	0	0

```
sonic# show ip ospf neighbor detail | no-more
```

```

Neighbor 10.59.142.247, interface address 64.1.1.2
In the area 0.0.0.0 via interface Ethernet64
Neighbor priority is 1, State is Full, 6 state changes
Most recent state change statistics:
  Progressive change 7h3m25s ago
  DR is 64.1.1.1, BDR is 64.1.1.2
  Options 2 *|-|-|-|-|E|-|
  Dead timer due in 30.687s
  Database Summary List 0
  Link State Request List 0
  Link State Retransmission List 0
  Thread Inactivity Timer on
  Thread Database Description Retransmission off
  Thread Link State Request Retransmission on
  Thread Link State Update Retransmission on

```

```
Leaf1# show ip ospf neighbor 2.2.2.2 | no-more
```

```

Neighbor 2.2.2.2, interface address 64.1.1.2
In the area 0.0.0.0 via interface Ethernet66
Neighbor priority is 1, State is Full, 5 state changes
Most recent state change statistics:
  Progressive change 0h1m11s ago
  DR is 64.1.1.1, BDR is 64.1.1.2
  Options 2 *|-|-|-|-|E|-|
  Dead timer due in 33.203s
  Database Summary List 0

```

```

Link State Request List 0
Link State Retransmission List 0
Thread Inactivity Timer on
Thread Database Description Retransmision off
Thread Link State Request Retransmission on
Thread Link State Update Retransmission on

Neighbor 2.2.2.2, interface address 65.1.1.2
  In the area 0.0.0.1 via interface Ethernet67
  Neighbor priority is 1, State is Full, 5 state changes
  Most recent state change statistics:
    Progressive change 0h1m10s ago
  DR is 65.1.1.1, BDR is 65.1.1.2
  Options 2 *|-|-|-|-|E|-|
  Dead timer due in 34.590s
  Database Summary List 0
  Link State Request List 0
  Link State Retransmission List 0
  Thread Inactivity Timer on
  Thread Database Description Retransmision off
  Thread Link State Request Retransmission on
  Thread Link State Update Retransmission on

```

```

sonic# show ip ospf interface | no-more
VRF Name: default
Ethernet64 is up
  ifindex 128, MTU 9100 bytes, BW 25000 Mbit UP,BROADCAST,RUNNING,MULTICAST
  Internet Address 64.1.1.1/24, Broadcast 64.1.1.255, Area 0.0.0.0
  MTU mismatch detection: enabled
  Router ID 10.59.143.131, Network Type BROADCAST, Cost: 4
  Transmit Delay is 1 sec, State DR, Priority 1
  Backup Designated Router (ID) 10.59.142.247, Interface Address 64.1.1.2
  Saved Network-LSA sequence number 0x8000000f
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 9.023s
  Neighbor Count is 1, Adjacent neighbor count is 1

```

```

Leaf1# show ip ospf interface Ethernet67 | no-more
VRF Name: default
Ethernet67 is up
  ifindex 926, MTU 9100 bytes, BW 25000 Mbit UP,BROADCAST,RUNNING,MULTICAST
  Internet Address 65.1.1.1/24, Broadcast 65.1.1.255, Area 0.0.0.1
  MTU mismatch detection: enabled
  Router ID 1.1.1.1, Network Type BROADCAST, Cost: 4
  Transmit Delay is 1 sec, State DR, Priority 1
  Backup Designated Router (ID) 2.2.2.2, Interface Address 65.1.1.2
  Multicast group memberships: OSPFAllRouters OSPFDesignatedRouters
  Timer intervals configured, Hello 10s, Dead 40s, Wait 40s, Retransmit 5
    Hello due in 7.957s
  Neighbor Count is 1, Adjacent neighbor count is 1

```

```
sonic# show ip ospf interface traffic | no-more
```

Interface	HELLO Rx/Tx	DB-Desc Rx/Tx	LS-Req Rx/Tx	LS-Update Rx/Tx	LS-Ack Rx/Tx
-----------	----------------	------------------	-----------------	--------------------	-----------------

Ethernet64	2563/2563	3/3	1/1	17/30	29/16
Leaf1# show ip ospf interface traffic Ethernet67 no-more					
Interface	HELLO Rx/Tx	DB-Desc Rx/Tx	LS-Req Rx/Tx	LS-Update Rx/Tx	LS-Ack Rx/Tx
Ethernet67	19/22	2/3	1/1	3/3	2/2

```
sonic# show ip ospf vrf Vrf-Blue route | no-more
VRF Name: Vrf-Blue
===== OSPF network routing table =====
N 10.1.1.1/32      [10] area: 0.0.0.0
                           directly attached to Loopback0
N 10.1.1.2/32      [14] area: 0.0.0.0
                           via 10.10.3.2, Ethernet64
                           via 10.10.8.2, Ethernet65.8
N 10.10.3.0/24     [4] area: 0.0.0.0
                           directly attached to Ethernet64
N 10.10.4.0/24     [10] area: 0.0.0.0
                           directly attached to Vlan2
N 10.10.5.0/24     [10] area: 0.0.0.0
                           directly attached to Vlan3
N 10.10.6.0/24     [10] area: 0.0.0.0
                           directly attached to PortChannel1
N 10.10.8.0/24     [4] area: 0.0.0.0
                           directly attached to Ethernet65.8

===== OSPF router routing table =====
R 10.1.1.2          [4] area: 0.0.0.0, ASBR
                           via 10.10.3.2, Ethernet64
                           via 10.10.8.2, Ethernet65.8

===== OSPF external routing table =====
N E2 18.18.18.0/24   [4/20] tag: 0
                           via 10.10.3.2, Ethernet64
                           via 10.10.8.2, Ethernet65.8
```

```
sonic# show ip ospf database | no-more
VRF Name: default

        OSPF Router with ID (10.59.143.131)

        Router Link States (Area 0.0.0.0)

Link ID      ADV Router      Age  Seq#      CkSum  Link count
10.59.142.247 10.59.142      1682 0x80000011 0xb56b 1
10.59.143.131 10.59.143      1498 0x80000011 0xdc2c 1

        Net Link States (Area 0.0.0.0)

Link ID      ADV Router      Age  Seq#      CkSum
64.1.1.1      10.59.143.131 1538 0x8000000f 0x1c70
```

```
sonic# show ip ospf database router | no-more
VRF Name: default

    OSPF Router with ID (10.59.143.131)

        Router Link States (Area 0.0.0.0)

    LS age: 1709
    Options: 0x2 : *|-|-|-|-|E|-
    LS Flags: 0x6
    Flags: 0x0 :
    LS Type: router-LSA
    Link State ID: 10.59.142.247
    Advertising Router: 10.59.142
    LS Seq Number: 80000011
    Checksum: 0xb56b
    Length: 36

    Number of Links: 1

    Link connected to: a Transit Network
        (Link ID) Designated Router address: 64.1.1.1
        (Link Data) Router Interface address: 64.1.1.2
        Number of TOS metrics: 0
            TOS 0 Metric: 4

    LS age: 1525
    Options: 0x2 : *|-|-|-|-|E|-
    LS Flags: 0x3
    Flags: 0x0 :
    LS Type: router-LSA
    Link State ID: 10.59.143.131
    Advertising Router: 10.59.143
    LS Seq Number: 80000011
    Checksum: 0xdc2c
    Length: 36

    Number of Links: 1

    Link connected to: a Transit Network
        (Link ID) Designated Router address: 64.1.1.1
        (Link Data) Router Interface address: 64.1.1.1
        Number of TOS metrics: 0
            TOS 0 Metric: 4
```

```
sonic# show ip ospf database network | no-more
VRF Name: default

    OSPF Router with ID (10.59.143.131)

        Net Link States (Area 0.0.0.0)

    LS age: 1602
    Options: 0x2 : *|-|-|-|-|E|-
    LS Flags: 0x3
    LS Type: network-LSA
```

```
Link State ID: 64.1.1.1 (address of Designated Router)
Advertising Router: 10.59.143.131
LS Seq Number: 8000000f
Checksum: 0x1c70
Length: 32

Network Mask: /24
Attached Router: 10.59.142.247

Attached Router: 10.59.143.131
```

```
Leaf1# show ip ospf database summary | no-more
VRF Name: default

OSPF Router with ID (1.1.1.1)

Summary Link States (Area 0.0.0.0)

LS age: 468
Options: 0x2 : *|-|-|-|-|E|-|
LS Flags: 0x11
LS Type: summary-LSA
Link State ID: 65.1.1.0 (summary Network Number)
Advertising Router: 1.1.1.1
LS Seq Number: 80000001
Checksum: 0x0e04
Length: 28

Network Mask: /24
TOS: 0 Metric: 4

LS age: 429
Options: 0x2 : *|-|-|-|-|E|-|
LS Flags: 0x6
LS Type: summary-LSA
Link State ID: 65.1.1.0 (summary Network Number)
Advertising Router: 2.2.2.2
LS Seq Number: 80000002
Checksum: 0xed1f
Length: 28

Network Mask: /24
TOS: 0 Metric: 4
```

Summary Link States (Area 0.0.0.1)

```
LS age: 468
Options: 0x2 : *|-|-|-|-|E|-|
LS Flags: 0x11
LS Type: summary-LSA
Link State ID: 64.1.1.0 (summary Network Number)
Advertising Router: 1.1.1.1
LS Seq Number: 80000001
Checksum: 0x1bf7
Length: 28
```

```
Network Mask: /24
TOS: 0 Metric: 4

LS age: 429
Options: 0x2 : *|-|-|-|-|E|-
LS Flags: 0x6
LS Type: summary-LSA
Link State ID: 64.1.1.0 (summary Network Number)
Advertising Router: 2.2.2.2
LS Seq Number: 80000002
Checksum: 0xfa13
Length: 28

Network Mask: /24
TOS: 0 Metric: 4
```

```
Leaf1# show ip ospf database asbr-summary | no-more
VRF Name: default

    OSPF Router with ID (1.1.1.1)

        ASBR-Summary Link States (Area 0.0.0.0)

LS age: 38
Options: 0x2 : *|-|-|-|-|E|-
LS Type: summary-LSA
Link State ID: 2.2.2.2 (AS Boundary Router address)
Advertising Router: 1.1.1.1
LS Seq Number: 80000001
Checksum: 0xb41
Length: 28

Network Mask: /0
TOS: 0 Metric: 4
```

```
Leaf1# show ip ospf database external | no-more
VRF Name: default

    OSPF Router with ID (1.1.1.1)

        AS External Link States

LS age: 52
Options: 0x2 : *|-|-|-|-|E|-
LS Flags: 0x6
LS Type: AS-external-LSA
Link State ID: 25.1.1.1 (External Network Number)
Advertising Router: 2.2.2.2
LS Seq Number: 80000001
Checksum: 0x0892
Length: 36

Network Mask: /32
Metric Type: 2 (Larger than any link state path)
TOS: 0
```

```
Metric: 20
Forward Address: 0.0.0.0
External Route Tag: 0
```

```
Leaf1# show ip ospf database max-age
      OSPF Router with ID (1.1.1.1)
      MaxAge Link States:
```

```
Leaf1# show ip ospf database self-originate | no-more
VRF Name: default
```

```
      OSPF Router with ID (1.1.1.1)

      Router Link States (Area 0.0.0.0)

Link ID      ADV Router      Age  Seq#      CkSum  Link count
1.1.1.1      1.1.1.1       777  0x80000004 0x7b42 1

      Net Link States (Area 0.0.0.0)

Link ID      ADV Router      Age  Seq#      CkSum
64.1.1.1     1.1.1.1       777  0x80000001 0x8581

      Summary Link States (Area 0.0.0.0)

Link ID      ADV Router      Age  Seq#      CkSum  Route
65.1.1.0     1.1.1.1       816  0x80000001 0x0e04 65.1.1.0/24

      ASBR-Summary Link States (Area 0.0.0.0)

Link ID      ADV Router      Age  Seq#      CkSum
2.2.2.2      1.1.1.1       360  0x80000001 0x0b41

      Router Link States (Area 0.0.0.1)

Link ID      ADV Router      Age  Seq#      CkSum  Link count
1.1.1.1      1.1.1.1       776  0x80000004 0x8d2e 1

      Net Link States (Area 0.0.0.1)

Link ID      ADV Router      Age  Seq#      CkSum
65.1.1.1     1.1.1.1       776  0x80000001 0x788d

      Summary Link States (Area 0.0.0.1)

Link ID      ADV Router      Age  Seq#      CkSum  Route
64.1.1.0     1.1.1.1       816  0x80000001 0x1bf7 64.1.1.0/24
```

```
Leaf1# show ip ospf database network adv-router 1.1.1.1 | no-more
VRF Name: default
```

```
      OSPF Router with ID (1.1.1.1)
```

```

Net Link States (Area 0.0.0.0)

LS age: 886
Options: 0x2 : *|-|-|-|-|E|-
LS Flags: 0x3
LS Type: network-LSA
Link State ID: 64.1.1.1 (address of Designated Router)
Advertising Router: 1.1.1.1
LS Seq Number: 80000001
Checksum: 0x8581
Length: 32

Network Mask: /24
Attached Router: 1.1.1.1

Attached Router: 2.2.2.2

Net Link States (Area 0.0.0.1)

LS age: 886
Options: 0x2 : *|-|-|-|-|E|-
LS Flags: 0x3
LS Type: network-LSA
Link State ID: 65.1.1.1 (address of Designated Router)
Advertising Router: 1.1.1.1
LS Seq Number: 80000001
Checksum: 0x788d
Length: 32

Network Mask: /24
Attached Router: 1.1.1.1

Attached Router: 2.2.2.2

```

```

sonic# show ip ospf graceful-restart helper detail

OSPF Router with ID (192.168.10.2)

Graceful restart helper support enabled.
Strict LSA check is enabled.
Helper supported for planned restarts only.
Supported Graceful restart interval: 500(in seconds).

```

Features this CLI belongs to

- OSPFv2

show ip prefix-list

Description

```
Display IPv4 prefix-list
```

Syntax

```
show ip prefix-list [ <list-name> ]
```

Parameters

Name	Description	Type
list-name	WORD	String

show ip route

Description

```
IP route information
```

Syntax

```
show ip route [ vrf { { all summary } | <vrfname> } ] { [ <address> ] | [ <prefix> ] | [ summary ] | [ bgp ] | [ connected ] | [ static ] | [ ospf ] } ]
```

Parameters

Name	Description	Type
vrfname	WORD	String
address	A.B.C.D	String
prefix	A.B.C.D/mask	String

show ip sla

Description

```
Displays IP SLA information
```

Syntax

```
show ip sla [ <id> [ history ] ]
```

Parameters

Name	Description	Type
id		Integer

Usage Guidelines

Use this command to display IP SLA summary information of all instances on a system, to display IP SLA detailed information of a particular instance or history information of a particular instance

Examples

```
sonic-cli(config)# show ip sla
      or
sonic-cli(config)# show ip sla 10
      or
sonic-cli(config)# show ip sla 10 history
```

SLA#	Type	State	Target	VRF	Transitions	Last change
10	ICMP-echo	Up	30.30.1.2	default	1	00:06:41 ago
20	TCP-connect	Up	40.40.1.2(100)	default	1	00:05:40 ago

```
sonic# show ip sla 10
IP SLA Operation Number: 10
Type of Operation: ICMP-echo
ICMP destination IP address: 30.30.1.2
ICMP source IP address: 30.30.1.1
ICMP source interface:
ICMP request data size: 32
ICMP Time-To-Live(TTL): 0
ICMP Type-of-Service(ToS): 0
Source VRF: default
Operation frequency (sec): 30
Operation timeout (sec): 5
Operation threshold: 3
Operation state: Up
Operation state transitions: 1
Operation last state change: 00:08:39 ago
ICMP Echo Request counter: 107
ICMP Echo Reply counter: 107
ICMP Error counter: 0
ICMP Invalid responses: 0
```

```
sonic# show ip sla 20
Type of Operation: TCP-connect
TCP destination IP address: 40.40.1.2
TCP destination port: 100
TCP source IP address: 40.40.1.1
TCP source port: 200
TCP source interface:
TCP Time-To-Live(TTL): 0
TCP Type-of-Service(ToS): 0
Source VRF: default
Operation frequency (sec): 30
Operation timeout (sec): 3
Operation threshold: 3
Operation state: Up
Operation state transitions: 1
Operation last state change: 00:28:59 ago
TCP connect request counter: 58
TCP connect success counter: 58
TCP connect error counter: 0
```

```
sonic# show ip sla 10 history
Timestamp          Event
-----
Fri Sep 25 22:54:36 2020 State changed to: Up
Fri Sep 25 22:54:21 2020 Started
```

Features this CLI belongs to

- IPSLA

show ip static-anycast-gateway

Description

Displays IPv4 static-anycast-gateway information.

Syntax

```
show ip static-anycast-gateway
```

Examples

```
sonic# show ip static-anycast-gateway
Configured Anycast Gateway MAC address: 00:22:33:44:55:66
IPv4 Anycast Gateway MAC address: enable
Total number of gateway: 3
Total number of gateway admin UP: 3
Total number of gateway oper UP: 3
Interfaces      Gateway Address      Vrf      Admin/Oper
-----      -----      -----      -----
```

Vlan3	30.30.1.1/24	up/up
Vlan5	50.0.0.1/24	up/up
Vlan54	54.1.1.1/24	up/up

show ip vrf

Description

Display configuration information for a specified VRF and its associated interfaces or all VRF and their associated interfaces.

Syntax

```
show ip vrf [ <vrf-name> ]
```

Parameters

Name	Description	Type
vrf-name	WORD	String

Usage Guidelines

```
sonic# show ip vrf
```

Examples

```
sonic# show ip vrf
VRF-NAME           INTERFACES
-----
mgmt              eth0
Vrf_red           Ethernet16
                  Ethernet8
```

```
sonic# show ip vrf Vrf_red
VRF-NAME           INTERFACES
-----
Vrf_red           Ethernet16
                  Ethernet8
```

show ip vrf mgmt

Description

Display configuration information for management VRF.

Syntax

```
show ip vrf mgmt
```

Usage Guidelines

```
sonic# show ip vrf mgmt
```

Examples

```
sonic# show ip vrf mgmt
VRF-NAME           INTERFACES
-----
mgmt              eth0
```

show ipv6 access-group

Description

Display IPv6 ACL binding summary

Syntax

```
show ipv6 access-group
```

Examples

```
Ingress IPV6 access-list ipv6acl-example on Ethernet0
```

Alternate command

```
admin@sonic:~$ show acl table [name]
```

show ipv6 access-lists

Description

Show IPv6 ACL rules and statistics

Syntax

```
show ipv6 access-lists [ <access-list-name> { { [ interface { Ethernet | <PortChannel> | <Vlan>
    | <eth-sub-if-id> | <po-sub-if-id> } ] } | [ Switch ] ] }
```

Parameters

Name	Description	Type
access-list-name	WORD	String
PortChannel	PortChannelNUM	
Vlan	VlanNUM	
eth-sub-if-id	EthernetX.Y	
po-sub-if-id	PortChannelNUM	

Usage Guidelines

ACL name and interface names are optional. If ACL name is not specified then all IPv6 ACLs will be displayed. ACL statistics will be shown only if the ACL is applied globally or to any interface.

Examples

```
ipv6 access-list ipv6acl-example
  seq 100 permit ipv6 host abcd::1 host bcde::1 (0 packets) [0 bytes]
  seq 200 permit tcp host abcd::2 host bcde::2 (0 packets) [0 bytes]
  seq 300 permit udp host abcd::3 host bcde::3 (0 packets) [0 bytes]
```

Alternate command

```
admin@sonic:~$ show acl rule [name] [rule_name]
```

show ipv6 dhcp snooping

Description

Show DHCP Snooping IPv6 global configuration

Syntax

```
show ipv6 dhcp snooping
```

```
show ipv6 dhcp snooping binding
```

Description

```
Show DHCP Snooping IPv6 binding database
```

Syntax

```
show ipv6 dhcp snooping binding
```

```
show ipv6 dhcp snooping statistics
```

Description

```
Show DHCPv6 Snooping statistics
```

Syntax

```
show ipv6 dhcp snooping statistics
```

```
show ipv6 dhcp snooping statistics detail
```

Description

```
Show DHCPv6 Snooping statistics
```

Syntax

```
show ipv6 dhcp snooping statistics detail
```

```
show ipv6 dhcp-relay
```

Description

```
Display IPv6 DHCP relay information
```

Syntax

```
show ipv6 dhcp-relay { [ brief ] | { [ detailed { [ <intfName1> ] | [ <pchName1> ] | [ <vlanName1> ] } ] } | { [ statistics { [ <intfName> ] | [ <pchName> ] | [ <vlanName> ] } ] }
```

Parameters

Name	Description	Type
intfName1	EthernetX.Y	
pchName1	PortChannelNUM	
vlanName1	VlanNUM	
intfName	EthernetX.Y	
pchName	PortChannelNUM	
vlanName	VlanNUM	

show ipv6 host-access-lists

Description

Show IPv6 host ACL rules

Syntax

```
show ipv6 host-access-lists [ <access-list-name> { { [ interface { Ethernet | <PortChannel> | <Vlan> | <eth-sub-if-id> | <po-sub-if-id> } ] } | [ Switch ] ] }
```

Parameters

Name	Description	Type
access-list-name	WORD	String
PortChannel	PortChannelNUM	
Vlan	VlanNUM	
eth-sub-if-id	EthernetX.Y	
po-sub-if-id	PortChannelNUM	

Usage Guidelines

Host ACL name and interface names are optional. If host ACL name is not specified then all IPv6 host ACLs will be displayed.

Examples

```
ipv6 host-access-list ipv6acl-example
  seq 100 permit ipv6 host abcd::1
  seq 200 permit tcp host abcd::2
  seq 300 permit udp host abcd::3 aeth-syndrome 0x63/0x63
```

show ipv6 interfaces

Description

```
IPv6 info of interfaces
```

Syntax

```
show ipv6 interfaces
```

show ipv6 neighbors

Description

This command displays NDP table entries. To filter the output, specify an interface, port channel, or VLAN, an IPv6 address, a MAC address, or a combination of more than one value to match. You can also display total number of ARP entries using 'summary' option.

Syntax

```
show ipv6 neighbors [ vrf { <vrfname> | mgmt | all } ] { [ <ip-addr> ] | { [ mac-address <mac-addr> ] } | [ summary ] } ]
```

Parameters

Name	Description	Type
vrfname	VRF name (prefixed by Vrf, Max: 15 characters)	String
ip-addr	A::B	String
mac-addr	nn:nn:nn:nn:nn:nn	String

Usage Guidelines

```
sonic# show ipv6 neighbors [interface { Ethernet < port > [summary] | PortChannel < id > [summary] | Vlan < id > [summary] }] [< ipv6-address >] [mac-address < mac >] [summary]
```

Examples

```
sonic# show ipv6 neighbors
```

Address	Hardware address	Interface	Egress Interface
20::1	00:01:02:03:44:55	Ethernet8	-
20::2	00:01:02:03:ab:cd	PortChannel200	-
20::3	00:01:02:03:04:05	Vlan100	Ethernet4
fe80::e6f0:4ff:fe79:34c7	00:01:e8:8b:44:71	eth0	-

```
sonic# show ipv6 neighbors Vlan 100
```

Address	Hardware address	Interface	Egress Interface
20::3	00:01:02:03:04:05	Vlan100	Ethernet4
sonic# show ipv6 neighbors interface Management 0			
Address	Hardware address	Interface	Egress Interface
fe80::e6f0:4ff:fe79:34c7	00:01:e8:8b:44:71	eth0	-

```
sonic# show ipv6 neighbors 20::2
```

Address	Hardware address	Interface	Egress Interface
20::2	00:01:02:03:ab:cd	PortChannel200	-

```
sonic# show ipv6 neighbors mac-address 00:01:02:03:04:05
```

Address	Hardware address	Interface	Egress Interface
20::3	00:01:02:03:04:05	Vlan100	Ethernet4

show ipv6 neighbors interface

Description

```
NDP entries for this interface
```

Syntax

```
show ipv6 neighbors interface { { [ <phy-if-name> [ summary ] ] } | { [ <subif-name> [ summary ] ] } | { Loopback { <lo-id> [ summary ] } } | { Management { <mgmt-if-id> [ summary ] } } | { PortChannel { <lag-id> [ summary ] } } | { Vlan { <vlan-id> [ summary ] } } | { Vxlan { <vxlan-if-name> [ summary ] } } }
```

Parameters

Name	Description	Type
phy-if-name	EthernetNUM	
subif-name	Interface Type	
lo-id		Integer
mgmt-if-id		Integer
lag-id		Integer
vlan-id		Integer
vxlan-if-name	WORD	String

show ipv6 ospf6

Description

Show OSPFv3 router information

Syntax

```
show ipv6 ospf6 [ vrf { <vrf-name> | all } ] { { [ area { <areaid> { spf tree } } ] } | { [ border-routers { [ <router-id> ] | [ detail ] ] } ] } | { [ interface { { [ traffic { [ prefix { { [ <route-bestmatch> { [ match [ detail ] ] } ] } ] } | { [ <route> { [ match [ detail ] ] } ] } ] } | { [ <interfacename> { [ prefix { { [ <route-bestmatch> { [ match [ detail ] ] } ] } ] } | { [ <route> { [ match [ detail ] ] } ] } ] } | { [ linkstate { { router <router-id> } | { network { <router-id> <link-state-id> } } | detail } ] } | [ redistribute ] | { [ route { [ <address> ] | { [ <prefix> { { [ match [ detail ] ] } | [ longer ] ] } ] } | { [ external-1 [ detail ] ] } | { [ external-2 [ detail ] ] } ] } | { [ inter-area [ detail ] ] } | { [ intra-area [ detail ] ] } | [ summary ] ] } ] } | { [ simulate { spf-tree { <router-id> { area <area-id> } } } ] } | { [ neighbor { [ <router-id> ] | [ detail ] | [ drchoice ] ] } ] } | { [ spf tree ] } | { [ summary-address [ detail ] ] } ] }
```

Parameters

Name	Description	Type
vrf-name	WORD	String
areaid	A.B.C.D	String
router-id	A.B.C.D	String
route-bestmatch	X:X::X:X	String
route	X:X::X:X/M	String
interfacename	Interface Type - Ranges	
link-state-id	A.B.C.D	String
address	X:X::X:X	String
prefix	X:X::X:X/M	String
area-id	A.B.C.D	String

```
show ipv6 ospf6 vrf
```

Description

Show OSPF6 VRFs

Syntax

```
show ipv6 ospf6 vrf
```

```
show ipv6 prefix-list
```

Description

Display IPv6 prefix-list

Syntax

```
show ipv6 prefix-list [ <list-name> ]
```

Parameters

Name	Description	Type
list-name	WORD	String

```
show ipv6 route
```

Description

IP route information

Syntax

```
show ipv6 route [ vrf { { all summary } | <vrfname> } ] { [ <address> ] | [ <prefix> ] | [ summary ] | [ bgp ] | [ connected ] | [ static ] } ]
```

Parameters

Name	Description	Type
vrfname	WORD	String
address	A::B	String
prefix	A::B/mask	String

show ipv6 static-anycast-gateway

Description

Displays IPv6 static-anycast-gateway information.

Syntax

```
show ipv6 static-anycast-gateway
```

Examples

```
sonic# show ipv6 static-anycast-gateway
Configured Anycast Gateway MAC address: 00:22:33:44:55:66
IPv6 Anycast Gateway MAC address: enable
Total number of gateway: 2
Total number of gateway admin UP: 2
Total number of gateway oper UP: 2
Interfaces Gateway Address      Vrf      Admin/Oper
-----  -----  -----
Vlan3    30::1/64                up/up
Vlan5    50::1/64                up/up
```

show kdump files

Description

Show the kdump kernel core dump files which are stored locally.

Syntax

```
show kdump files
```

Usage Guidelines

Use this command to show the kdump kernel core dump files which are stored locally.

Examples

```
sonic# show kdump files
Record Key          Filename
-----
1 202002101809 /var/crash/202002101809/dmesg.202002101809
                  /var/crash/202002101809/kdump.202002101809
```

Features this CLI belongs to

- KDUMP

Alternate command

```
show kdump files
```

show kdump log

Description

Show a kdump kernel core dump file kernel log from a file stored locally.

Syntax

```
show kdump log <record> [ <lines> ]
```

Parameters

Name	Description	Type
record	kdump dump file record number	Integer
lines		Integer

Usage Guidelines

Use this command to show a kdump kernel core dump file kernel log from a file stored locally. The mandatory parameter is the number of the kernel core dump files which are stored locally. The optional parameter is the number of lines displayed (20 is the default number of lines displayed).

Examples

```
sonic# show kdump log 1 5
File: /var/crash/202002101809/dmesg.202002101809
[326785.222049]  [<fffffffffa0c0484e>] ? entry_SYSCALL_64_after_swapgs+0x58/0xc6
[326785.229926] Code: 41 5c 41 5d 41 5e 41 5f e9 6c 2f cf ff 66 2e 0f 1f 84 00 00 00 00 00 00 66
    90 0f 1f 44 00 00 c7 05 29 28 a8 00 01 00 00 00 0f ae f8 <c6> 04 25 00 00 00 00 01 c3 0f 1f
    44 00 00 0f 1f 44 00 00 53 8d
[326785.251451] RIP  [<fffffffffa0a2a562>] sysrq_handle_crash+0x12/0x20
[326785.258463] RSP <fffffafd2c6523e78>
[326785.262453] CR2: 0000000000000000
```

In this example, we show the kernel log for the first kernel core dump file stored locally. We display only the first 5 lines of the log.

Features this CLI belongs to

- KDUMP

Alternate command

```
show kdump log
```

show kdump memory

Description

Show the amount of memory reserved and allocated for kdump operation.

Syntax

```
show kdump memory
```

Usage Guidelines

Use this command to show the configured amount of memory reserved for kdump operation. It also displays the actual memory allocated.

Examples

```
sonic# show kdump memory
Memory Reserved: 0M-2G:256M,2G-4G:256M,4G-8G:384M,8G-:448M
Memory Allocated: 448M
```

Features this CLI belongs to

- KDUMP

Alternate command

```
show kdump memory
```

show kdump num-dumps

Description

Show the maximum number of kernel core dump files which can be stored locally.

Syntax

```
show kdump num-dumps
```

Usage Guidelines

Use this command to show the maximum number of kernel core dump files which can be stored locally.

Examples

```
sonic# show kdump num-dumps
Maximum number of Kernel Core files Stored: 3
```

Features this CLI belongs to

- KDUMP

Alternate command

```
show kdump num_dumps
```

show kdump status

Description

Show the status of kdump operation.

Syntax

```
show kdump status
```

Usage Guidelines

Use this command to show the status of kdump operation.

Examples

```
sonic# show kdump status
Kdump Administrative Mode: Enabled
Kdump Operational State: Ready
Memory Reserved: 512M
Maximum number of Kernel Core files Stored: 3
Record Key          Filename
-----
1 202002101809 /var/crash/202002101809/dmesg.202002101809
                  /var/crash/202002101809/kdump.202002101809
```

Features this CLI belongs to

- KDUMP

Alternate command

```
show kdump status
```

show link state tracking

Description

Show the link state tracking group operational state information.

Syntax

```
show link state tracking [ <grp-name> ]
```

Parameters

Name	Description	Type
grp-name	name	String

Usage Guidelines

Link state tracking group name can be of maximum 63 characters. The name must begin with A-Z, a-z or 0-9. Underscore and hyphens can be used except as the first character.
If the group name is not specified then a summary of all configured groups will be displayed.

Examples

```
sonic# show link state tracking FooBar
Name: FooBar
Description: Example description
Timeout: 120 seconds
Upstream Interfaces:
    Ethernet0 (Up)
    Ethernet4 (Up)
    Vlan100 (Up)
Downstream Interfaces:
    PortChannel1 (Up)
    PortChannel2 (Up)
    Ethernet4 (Up)
```

Alternate command

```
admin@sonic:~$ show linktrack group <name>
```

show lldp neighbor

Description

Shows LLDP neighbor information in detail

Syntax

```
show lldp neighbor [ <ifname> ]
```

Parameters

Name	Description	Type
ifname	EthernetNUM	

Usage Guidelines

This command is useful to view the LLDP neighbor information in detail

Examples

```
sonic-cl# show lldp neighbor
-----
LLDP Neighbors
-----
Interface: Ethernet64,via: LLDP
Chassis:
    ChassisID: 80:a2:35:26:48:5e
    SysName: Leaf9
    SysDescr: Debian GNU/Linux 9 (stretch) Linux 4.9.0-11-2-amd64 #1 SMP Debian
               4.9.189-3+deb9u2 (2019-11-11) x86_64
    MgmtIP: 10.59.132.165
    MgmtIP: 10.59.132.165
    Capability: MAC_BRIDGE, ON
    Capability: ROUTER, ON
Port
    PortID: hundredGigE53
    PortDescr: Ethernet64
-----
```

show lldp statistics

Description

Shows LLDP statistics information

Syntax

```
show lldp statistics [ <ifname> ]
```

Parameters

Name	Description	Type
ifname	EthernetNUM	

Usage Guidelines

This command is useful to view the LLDP statistics information

Examples

```
sonic-cl# show lldp statistics
```

```
LLDP Statistics
```

```
-----
```

```
Interface: Ethernet0
    Transmitted      : 10
    Received        : 12
    Discarded       : 1
    Unrecognized TLV : 0
    Ageout          : 0
```

show lldp table

Description

```
Shows LLDP neighbor information in brief
```

Syntax

```
show lldp table
```

Usage Guidelines

```
This command is useful to view the LLDP neighbor information in brief
```

Examples

```
sonic-cl# show lldp table
```

LocalPort	RemoteDevice	RemotePortID	Capability	RemotePortDescr
Ethernet64	Leaf9	hundredGigE53	BR	Ethernet64

show locator-led

Description

```
Show Locator LED State
```

Syntax

```
show locator-led chassis
```

show logging

Description

Display logging information

Syntax

```
show logging
```

show logging count

Description

Total number of logging

Syntax

```
show logging count
```

show logging filter

Description

```
filter logs
```

Syntax

```
show logging filter { { since <since_date> { { [ level <log_level> ] } | { [ severity <log_severity> ] } } } | { level <log_level> { [ since <since_date> ] } } | { severity <log_severity> { [ since <since_date> ] } } | { type <log_type> { [ since <since_date> ] } { { [ level <log_level> ] } | { [ severity <log_severity> ] } } } }
```

Parameters

Name	Description	Type
since_date	String	String
log_level	DEBUG or INFO or WARNING or NOTICE or ERR or CRIT	Select [DEBUG INFO WARNING NOTICE ERR CRIT]
log_severity	DEBUG or INFO or WARNING or NOTICE or ERR or CRIT	Select [DEBUG INFO WARNING NOTICE ERR CRIT]
log_type	syslog or inmem or all	Select [syslog inmem all]

show logging lines

Description

Output of last NUM lines

Syntax

```
show logging lines [ <lines> ]
```

Parameters

Name	Description	Type
lines		Integer

show logging profiles

Description

Shows list of available logging profiles

Syntax

```
show logging profiles
```

Examples

```
sonic# show logging profiles
-----
Profile      Active
-----
default      true
1            false
2            false
sonic#
```

show logging servers

Description

Shows list of remote syslog servers are configured

Syntax

```
show logging servers
```

Examples

```
sonic# show logging servers
```

HOST	PORT	SOURCE-INTERFACE	VRF
30.1.1.1	514	-	Vrf2
40.1.1.1	514	-	-

show mab

Description

Show mab information.

Syntax

```
show mab
```

show mab interface

Description

Enter client interface to show mab interface.

Syntax

```
show mab interface <port>
```

Parameters

Name	Description	Type
port	EthernetNUM	

show mac access-group

Description

Display MAC ACL binding summary

Syntax

```
show mac access-group
```

Examples

```
Ingress MAC access-list macacl-example on Vlan100
```

Alternate command

```
admin@sonic:~$ show acl table [name]
```

show mac access-lists

Description

Show MAC ACL rules and statistics

Syntax

```
show mac access-lists [ <access-list-name> { { [ interface { Ethernet | <PortChannel> | <Vlan>
    | <eth-sub-if-id> | <po-sub-if-id> } ] } | [ Switch ] } ]
```

Parameters

Name	Description	Type
access-list-name	WORD	String
PortChannel	PortChannelNUM	
Vlan	VlanNUM	
eth-sub-if-id	EthernetX.Y	
po-sub-if-id	PortChannelNUM	

Usage Guidelines

ACL name and interface names are optional. If ACL name is not specified then all MAC ACLs will be displayed. ACL statistics will be shown only if the ACL is applied globally or to any interface.

Examples

```
mac access-list macacl-example
  seq 10 permit host 00:00:10:00:00:01 host 00:00:20:00:00:01 (10 packets) [1000 bytes]
  seq 20 permit host 00:00:10:00:00:02 host 00:00:20:00:00:02 (20 packets) [2000 bytes]
  seq 30 permit host 00:00:10:00:00:03 host 00:00:20:00:00:03 (30 packets) [3000 bytes]
  seq 40 permit host 00:00:10:00:00:04 host 00:00:20:00:00:04 (40 packets) [4000 bytes]
```

Alternate command

```
admin@sonic:~$ show acl rule [name] [rule_name]
```

show mac address-table

Description

```
MAC address-table
```

Syntax

```
show mac address-table
```

show mac address-table Vlan

Description

```
Display MAC address-table for VLAN
```

Syntax

```
show mac address-table Vlan <vlan-id>
```

Parameters

Name	Description	Type
vlan-id		Integer

show mac address-table address

Description

Display MAC address-table address for MAC address

Syntax

```
show mac address-table address <mac-addr>
```

Parameters

Name	Description	Type
mac-addr	nn:nn:nn:nn:nn:nn	String

show mac address-table aging-time

Description

MAC aging-time

Syntax

```
show mac address-table aging-time
```

show mac address-table count

Description

Count keyword

Syntax

```
show mac address-table count
```

show mac address-table dynamic

Description

MAC address-table for dynamic commands

Syntax

```
show mac address-table dynamic { { [ address <mac-addr> ] } | [ Vlan ] | { [ interface {  
    <phy-if-name> | <PortChannel> } ] } } ]
```

Parameters

Name	Description	Type
mac-addr	nn:nn:nn:nn:nn:nn	String
phy-if-name	EthernetNUM	
PortChannel	PortChannelNUM	

show mac address-table interface

Description

Display MAC address-table for interfaces

Syntax

```
show mac address-table interface { <phy-if-name> | <PortChannel> }
```

Parameters

Name	Description	Type
phy-if-name	EthernetNUM	
PortChannel	PortChannelNUM	

show mac address-table static

Description

MAC address-table for static commands

Syntax

```
show mac address-table static { { [ address <mac-addr> ] } | [ Vlan ] | { [ interface {  
    <phy-if-name> | <PortChannel> } ] } } ]
```

Parameters

Name	Description	Type
mac-addr	nn:nn:nn:nn:nn:nn	String
phy-if-name	EthernetNUM	
PortChannel	PortChannelNUM	

show mac dampening

Description

Display MAC dampening configuration

Syntax

```
show mac dampening
```

show mac dampening-disabled-ports

Description

Display MAC dampening-disabled-ports

Syntax

```
show mac dampening-disabled-ports
```

show mac move-policy

Description

Display MAC move policy configuration

Syntax

```
show mac move-policy configuration
```

show mac move-policy status

Description

```
Display the current in action for VLAN members/port
```

Syntax

```
show mac move-policy status
```

show mclag brief

Description

```
Show command to display MLAG domain summary
```

Syntax

```
show mclag brief
```

Usage Guidelines

```
sonic-cli# show mclag brief
```

Examples

```
sonic-cli# show mclag brief

Domain ID          : 10
Role               :
Session Status     :
Peer Link Status   :
Source Address     : 1.1.1.1
Peer Address       :
Peer Link          : Eth1/12/1
Keepalive Interval : 1 secs
Session Timeout    : 30 secs
Delay Restore      : 300 secs
System Mac          :
Mclag System Mac   : 00:bb:bb:bb:cc:cc

Number of MLAG Interfaces:1
-----
MLAG Interface      Local/Remote Status
-----
PortChannel10        down/down
```

show mclag interface

Description

Show command to display MLAG interface information

Syntax

```
show mlag interface <ifid> <domain_id>
```

Parameters

Name	Description	Type
ifid		Integer
domain_id		Integer

Usage Guidelines

```
sonic-cli# show mlag interface PortchannelId DomainId
```

Examples

```
sonic-cli# show mlag 10 100
Local/Remote Status : down/down
TrafficDisable       : No
IsolateWithPeerLink : No
```

show mlag mac remote

Description

Show command to display MLAG remote mac count

Syntax

```
show mlag mac remote { [ <vlan-if-id> ] | [ <phy-if-name> ] | [ <PortChannel> ] }
```

Parameters

Name	Description	Type
vlan-if-id	VlanNUM	
phy-if-name	EthernetNUM	
PortChannel	PortChannelNUM	

Usage Guidelines

```
sonic-cl# show mclag mac remote
```

Examples

```
sonic-cl# show mclag mac remote
=====
Vlan      Mac          Port      Type
=====
Vlan300   00:0a:0a:0a:0a:0a  PortChannel10  static
Vlan400   00:0a:0a:0a:0b:0b  Ethernet12    dynamic
Vlan400   00:0a:0a:0a:0a:0a  PortChannel101  dynamic
```

show mclag mac remote count

Description

Show command to display MCLAG remote mac count

Syntax

```
show mclag mac remote count
```

Usage Guidelines

```
sonic-cl# show mclag remote mac count
```

Examples

```
sonic-cl# show mclag remote mac count
```

show mclag peer-gateway-interfaces

Description

Show command to display Vlan interfaces on which MCLAG Peer Gateway is configured.

Syntax

```
show mclag peer-gateway-interfaces
```

Usage Guidelines

```
sonic-cli# show mclag peer-gateway-interfaces
```

Examples

```
sonic-cli# show mclag peer-gateway-interfaces

Interface Name
=====
Vlan10
=====
Total count : 1
=====
```

show mclag separate-ip-interfaces

Description

Show command to display Vlan interfaces on which MCLAG separate IP is configured.

Syntax

```
show mclag separate-ip-interfaces
```

Usage Guidelines

```
sonic-cli# show mclag separate-ip-interfaces
```

Examples

```
sonic-cli# show mclag separate-ip-interfaces

Interface Name
=====
Vlan10
=====
Total count : 1
=====
```

show mirror-session

Description

Display configured mirror sessions

Syntax

```
show mirror-session [ <session-name> ]
```

Parameters

Name	Description	Type
session-name	String(Max: 24 characters)	String

Examples

```
sonic# show mirror-session
ERSPAN Sessions
-----
Name      Status     SRC-IP          DST-IP          GRE    DSCP   TTL   Queue   Policer
      SRC-Port   Direction
-----
Mirror2   active    11.1.1.1        10.1.1.1       0x88ee 10    10    10
      Ethernet4      rx
SPAN Sessions
-----
Name      Status     DST-Port        SRC-Port        Direction
-----
Mirror1  active    Ethernet0      Ethernet4      rx
```

show nat

Description

Display NAT info

Syntax

```
show nat { { translations [ count ] } | statistics | { config { [ static ] | [ pool ] | [
  bindings ] | [ globalvalues ] | [ zones ] ] } } }
```

show neighbor-suppress-status

Description

Display ARP and ND suppression status

Syntax

```
show neighbor-suppress-status [ <id> ]
```

Parameters

Name	Description	Type
id		Integer

show ntp associations

Description

Display associated NTP servers and their properties.

Syntax

```
show ntp associations
```

Usage Guidelines

```
sonic# show ntp associations
```

Examples

```
sonic# show ntp associations
remote                refid      st  t  when   poll   reach  delay offset
jitter
-----
*10.11.0.1            10.14.1.1    4   u  44     64     255   0.258 17915.100
  151.732
+10.11.0.2            10.11.0.1    5   u  48     64     255   0.306 17774.800
  86.586
-----
* master (synced), # master (unsynced), + selected, - candidate, ~ configured
```

show ntp global

Description

Display global configuration for NTP, which includes VRF configuration and source interface configuration.

Syntax

```
show ntp global
```

Usage Guidelines

```
sonic# show ntp global
```

Examples

```
sonic# show ntp global
-----
NTP Global Configuration
-----
NTP source-interface:    Ethernet8
NTP vrf:                 default
```

show ntp server

Description

Display configured NTP servers.

Syntax

```
show ntp server
```

Usage Guidelines

```
sonic# show ntp server
```

Examples

```
sonic# show ntp server
-----
NTP Servers
-----
10.11.0.1
10.11.0.2
```

show object-groups

Description

Show Object Groups

Syntax

```
show object-groups { [ <show-object-group-name> ] | [ type ] } ] network
```

Parameters

Name	Description	Type
show-object-group-name	String(Max: 72 characters)	String

Usage Guidelines

Object group Type and name are optional. If Object group name and Object group type are not specified then all the object groups will be displayed. All the object group of a given type will be displayed if the object group type is specified. If the Object group name is specified then the specific object group will be displayed.

Examples

```
object-group object-group-example type network
    description: Object group example
    network-object host 1.1.1.1
```

show pbf next-hop-group

Description

Display group information

Syntax

```
show pbf next-hop-group { [ <show-fbs-group-name> ] | [ type ] } ] { ip | ipv6 }
```

Parameters

Name	Description	Type
show-fbs-group-name	WORD	String

show pbf next-hop-group status Switch

Description

Display group status

Syntax

```
show pbf next-hop-group status Switch { [ <show-fbs-group-name> ] | [ type ] } ] { ip | ipv6 }
```

Parameters

Name	Description	Type
show-fbs-group-name	WORD	String

show pbf next-hop-group status interface

Description

Display group status

Syntax

```
show pbf next-hop-group status interface { <eth-if-id> | <po-if-id> | <vlan-if-id> |
<eth-sub-if-id> | <po-sub-if-id> } { [ <show-fbs-group-name> ] | [ type ] } ] { ip | ipv6 }
```

Parameters

Name	Description	Type
eth-if-id	EthernetNUM	

Name	Description	Type
po-if-id	PortChannelNUM	
vlan-if-id	VlanNUM	
eth-sub-if-id	EthernetX.Y	
po-sub-if-id	PortChannelNUM	
show-fbs-group-name	WORD	String

show pbf replication-group

Description

Display group information

Syntax

```
show pbf replication-group { [ <show-fbs-group-name> ] | [ type ] } ] { ip | ipv6 }
```

Parameters

Name	Description	Type
show-fbs-group-name	WORD	String

show pbf replication-group status Switch

Description

Display group status

Syntax

```
show pbf replication-group status Switch { [ <show-fbs-group-name> ] | [ type ] } ] { ip | ipv6 }
```

Parameters

Name	Description	Type
show-fbs-group-name	WORD	String

show pbf replication-group status interface

Description

Display group status

Syntax

```
show pbf replication-group status interface { <eth-if-id> | <po-if-id> | <vlan-if-id> |
<eth-sub-if-id> | <po-sub-if-id> } { [ <show-fbs-group-name> ] | [ type ] } ] { ip | ipv6 }
```

Parameters

Name	Description	Type
eth-if-id	EthernetNUM	
po-if-id	PortChannelNUM	
vlan-if-id	VlanNUM	
eth-sub-if-id	EthernetX.Y	
po-sub-if-id	PortChannelNUM	
show-fbs-group-name	WORD	String

show platform environment

Description

Display platform Environment

Syntax

```
show platform environment
```

show platform fanstatus

Description

Display platform fan status

Syntax

```
show platform fanstatus
```

show platform firmware

Description

Display platform firmware

Syntax

show platform firmware

show platform firmware detail

Description

Display detailed firmware information

Syntax

show platform firmware detail

show platform inventory

Description

Display platform inventory information

Syntax

show platform inventory

show platform psustatus

Description

Display platform PSU status

Syntax

show platform psustatus

show platform psusummary

Description

Display platform PSU summary

Syntax

```
show platform psusummary
```

show platform sbstatus

Description

Display secure boot status

Syntax

```
show platform sbstatus
```

show platform syseeprom

Description

Display platform EEPROM information

Syntax

```
show platform syseeprom
```

show platform temperature

Description

Display platform temperature sensors

Syntax

```
show platform temperature
```

show platform temperature detail

Description

Display detailed temperature sensors information

Syntax

```
show platform temperature detail
```

show policy-map

Description

```
Shows flow based services policy-map related information
```

Syntax

```
show policy-map { [ <show-fbs-policy-name> ] | [ type ] } ] { qos | monitoring | forwarding |  
copp | acl-copp }
```

Parameters

Name	Description	Type
show-fbs-policy-name	WORD	String

Usage Guidelines

Policy-map type and policy-map name arguments are optional. If type argument or policy-map name not provided command will show all policy-map information.
Else it show corresponding policy-map information of given type or given name

Examples

```
Policy policy_mirror Type monitoring  
Description:  
Flow class1 at priority 10  
Description:  
set mirror-session mirror1  
Applied to:  
Vlan100 at Ingress

Policy policy_qos Type qos  
Description:  
Flow class_permit_ipv6 at priority 10  
Description:  
police cir 300000000 cbs 300000000 pir 300000000 pbs 300000000  
Flow class_permit_ip at priority 10  
Description:  
police cir 300000000 cbs 300000000 pir 300000000 pbs 300000000  
Applied to:  
Vlan100 at Egress
```

```

Policy policy_vrf Type forwarding
  Description:
    Flow class_permit_ipv6 at priority 10
      Description:
        set ipv6 nexthop 1211::2 priority 20
        set ipv6 nexthop 1212::2 vrf Vrf-BLUE priority 30
    Flow class_permit_ip at priority 10
      Description:
        set ip nexthop 12.12.1.2 vrf default priority 30
        set ip nexthop 12.12.2.2 vrf Vrf-BLUE priority 20
        set ip nexthop 12.12.1.2 priority 10
  Applied to:
    Vlan100 at Ingress
    Switch at Ingress

```

show port-group

Description

Display the list of port groups, it's member ports and valid speeds.
The port-group is not supported in all the hardware platforms.

Syntax

```
show port-group
```

Usage Guidelines

Use this command to know port group members, valid speeds and default speed.
show port-group

Examples

```

show sonic# show port-group
-----
Port-group  Interface range          Valid speeds      Default Speed
-----
1           Ethernet0 - Ethernet11   10G, 25G         25G
2           Ethernet12 - Ethernet23  10G, 25G         25G
3           Ethernet24 - Ethernet35  10G, 25G         25G
4           Ethernet36 - Ethernet47  10G, 25G         25G
sonic#

```

show port-security

Description

Shows Port Mac Security information

Syntax

```
show port-security
```

Usage Guidelines

After enabling PMS on desired ports, use this command to list information of all Port MAC security enabled ports

Examples

```
sonic-cl# show port-security
Secure Port      MaxSecureAddr      CurrentAddr  SecurityViolation  SecurityAction
Ethernet0          11                  11            0                Protect
Ethernet1          15                  5             0                Protect
PortChannel10      5                   4             0                Protect
```

show port-security interface

Description

Shows interface level PMS information on this interface

Syntax

```
show port-security interface <interface-name>
```

Parameters

Name	Description	Type
interface-name	String	String

Usage Guidelines

After enabling PMS on port, use this command to get PMS information for this port

Examples

```
sonic-cli# show port-security interface Ethernet0
Port Security: Enabled
Violation mode: Shutdown
Maximum MAC Addresses: 11
Total MAC Addresses: 11
Security Violation count: 0
```

show priority-flow-control

Description

Display PFC summary

Syntax

```
show priority-flow-control watchdog
```

show priority-group

Description

This command displays priority group shared/headroom watermarks and persistent-watermarks.

Syntax

```
show priority-group { { watermark { { headroom { [ interface { <phy-intf-name> } ] } } | { shared { [ interface { <phy-intf-name> } ] } } | { percentage { { headroom { [ interface { <phy-intf-name> } ] } } } | { shared { [ interface { <phy-intf-name> } ] } } | { persistent-watermark { { headroom { [ interface { <phy-intf-name> } ] } } } | { shared { [ interface { <phy-intf-name> } ] } } | { percentage { { headroom { [ interface { <phy-intf-name> } ] } } } | { shared { [ interface { <phy-intf-name> } ] } } }
```

Parameters

Name	Description	Type
phy-intf-name	EthernetNUM	

Usage Guidelines

Use this command to display priority group watermarks, persistent-watermarks etc.

There are various CLI options available to display information for shared, headroom and interface watermarks.

- show priority-group (watermark|persistent-watermark) (shared | headroom) (interface Ethernet [ifname])

This command will display priority groups shared/headroom watermarks.

Examples

```
sonic# show priority-group watermark shared  
Ingress shared pool watermark per PG:
```

Port	PG0	PG1	PG2	PG3	PG4	PG5	PG6	PG7
Ethernet0	0	0	0	0	0	0	0	0
Ethernet4	0	0	0	0	0	0	0	0
Ethernet8	0	0	0	0	0	0	0	0
Ethernet12	0	0	0	0	0	0	0	0
Ethernet16	0	0	0	0	0	0	0	0

```
sonic# show priority-group persistent-watermark headroom  
Ingress headroom persistent watermark per PG:
```

Port	PG0	PG1	PG2	PG3	PG4	PG5	PG6	PG7
Ethernet0	0	0	0	0	0	0	0	0
Ethernet4	0	0	0	0	0	0	0	0
Ethernet8	0	0	0	0	0	0	0	0
Ethernet12	0	0	0	0	0	0	0	0
Ethernet16	0	0	0	0	0	0	0	0
Ethernet20	0	0	0	0	0	0	0	0
Ethernet24	0	0	0	0	0	0	0	0

show ptp

Description

```
Display PTP status/configuration
```

Syntax

```
show ptp
```

Examples

```
sonic# show ptp
```

Interface	State
Ethernet56	master
Ethernet64	slave

show ptp clock

Description

```
Display PTP clock configuration/status
```

Syntax

```
show ptp clock
```

Examples

```
sonic# show ptp clock
Mode          BC
Domain Profile    ieee1588
Network Transport   UDPv4 unicast
Domain Number      1
Clock Identity     3c2c99.ffe.2d7c35
Priority1          128
Priority2          128
Two Step           Enabled
Slave Only         False
Number Ports       2
Clock Quality:
Clock Class        248
Clock Accuracy      254
Ofst Scaled Log Var 65535
Mean Path Delay     0
Steps Removed       0
Offset from master  0
```

show ptp parent

Description

Display PTP parent status

Syntax

```
show ptp parent
```

Examples

```
sonic# show ptp parent
Parent Clock Identity      3c2c99.ffe.2d7c35
Port Number                 0
Grandmaster Clock Class    248
Grandmaster Off Scaled Log Var 65535
Grandmaster Clock Accuracy 254
Grandmaster Identity        3c2c99.ffe.2d7c35
Grandmaster Priority1       128
Grandmaster Priority2       128
Stats Valid                 False
Observed Off Scaled Log Var 65535
Observed Clock Phase Chg Rate 2147483647
```

show ptp port

Description

Display PTP port status/configuration

Syntax

```
show ptp port <Ethernet>
```

Parameters

Name	Description	Type
Ethernet	EthernetNUM	

Examples

```
sonic# show ptp port Ethernet 64
Port Number          64
Port State           master
Log Min delay Req Intvl   0
Peer Mean Path Delay  0
Log Announce Interval 1
Log Sync Interval    0
Log Min PDelay Req Interval 0
Version Number       2
Unicast Master Table:
                      10.1.1.1
```

show ptp time-property

Description

Display PTP time-property status

Syntax

```
show ptp time-property
```

Examples

```
sonic# show ptp time-property
Curr UTC Offset Vld  False
Curr UTC Offset      37
Leap59               False
Leap61               False
Time Traceable       False
Freq Traceable       False
PTP Timescale        True
```

show qos

Description

Display QoS information

Syntax

```
show qos interface { CPU | { <phy-intf-name> { [ queue <queue-id> ] } { [ priority-flow-control
    { statistics [ queue ] } ] } } | <po-intf-name> | <vlan-intf-name> | <phy-sub-if-name> }
```

Parameters

Name	Description	Type
phy-intf-name	EthernetNUM	
queue-id	All or 0..7	String
po-intf-name	PortChannelNUM	
vlan-intf-name	VlanNUM	
phy-sub-if-name	Interface Type	

show qos map dot1p-tc

Description

Display configured dot1p-tc-map

Syntax

```
show qos map dot1p-tc [ <name> ]
```

Parameters

Name	Description	Type
name	WORD	String

show qos map dscp-tc

Description

```
Display configured dscp-tc-map
```

Syntax

```
show qos map dscp-tc [ <name> ]
```

Parameters

Name	Description	Type
name	WORD	String

show qos map pfc-priority-queue

Description

```
Display configured pfc-priority-queue-map
```

Syntax

```
show qos map pfc-priority-queue [ <name> ]
```

Parameters

Name	Description	Type
name	WORD	String

show qos map tc-dot1p

Description

```
Display configured tc-dot1p-map
```

Syntax

```
show qos map tc-dot1p [ <name> ]
```

Parameters

Name	Description	Type
name	WORD	String

show qos map tc-dscp

Description

Display configured tc-dscp-map

Syntax

show qos map tc-dscp [<name>]

Parameters

Name	Description	Type
name	WORD	String

show qos map tc-pg

Description

Display configured traffic-class-priority-group-map

Syntax

show qos map tc-pg [<name>]

Parameters

Name	Description	Type
name	WORD	String

show qos map tc-queue

Description

Display configured traffic-class-queue-map

Syntax

```
show qos map tc-queue [ <name> ]
```

Parameters

Name	Description	Type
name	WORD	String

show qos scheduler-policy

Description

Display scheduler policy information

Syntax

```
show qos scheduler-policy [ <name> ]
```

Parameters

Name	Description	Type
name	String	String

show qos wred-policy

Description

This command displays WRED policies configured on the device.

Syntax

```
show qos wred-policy [ <name> ]
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

User configures WRED policies to use it in interface queues.
This command enables users to display the WRED policies configured on this device.
User can provide WRED profile name optional CLI key to display only that profile.
If profile name is not specified, all policies will be displayed by this command.

Examples

```
sonic# show qos wred-policy
-----
Profile          : test
-----
Profile          : wred-green
ecn               : ecn_all
green-min-threshold : 100 KBytes
green-max-threshold : 200 KBytes
greetp-drop-rate   : 50
```

show queue

Description

This command displays queue counters, WRED ECN counters, watermarks, persistent-watermarks and breaches etc.

Syntax

```
show queue { { counters { [ interface { { <phy-intf-name> { [ queue <queue-id> ] } } ] } | { CPU { [ queue <queue-id> ] } } } } | { wred-ecn { counters { [ interface { <phy-intf-name> } ] } } } | { watermark { { unicast { [ interface { <phy-intf-name> } ] } } | { multicast { [ interface { <phy-intf-name> } ] } } | { percentage { { unicast { [ interface { <phy-intf-name> } ] } } | { multicast { [ interface { <phy-intf-name> } ] } } } } | { persistent-watermark { { unicast { [ interface { <phy-intf-name> } ] } } } | { multicast { [ interface { <phy-intf-name> } ] } } | { percentage { { unicast { [ interface { <phy-intf-name> } ] } } | { multicast { [ interface { <phy-intf-name> } ] } } } }
```

Parameters

Name	Description	Type
phy-intf-name	EthernetNUM	
queue-id		Integer

Usage Guidelines

Use this command to display queue counters, WRED ECN counters, watermarks, persistent-watermarks and breaches etc.

There are various CLI options available to display information for queue, interface and all interfaces.

- show queue counters (interface Ethernet|CPU [ifname] (queue [id]))
This command will display all queue counters
- show queue (watermark|persistent-watermark) (unicast| multicast|CPU) (interface Ethernet [ifname]))
This command will display all queue watermarks
- show queue wred-ecn counters (interface Ethernet [ifname])
This command will display all queue WRED ECN counters

Examples

```
sonic# show queue counters
-----
Port      TxQ  Counter/pkts  Counter/bytes  Drop/pkts  Drop/bytes
-----
Ethernet0  UC0  0            0              0          0
Ethernet0  UC1  0            0              0          0
Ethernet0  UC2  0            0              0          0
Ethernet0  UC3  0            0              0          0
Ethernet0  UC4  0            0              0          0
Ethernet0  UC5  0            0              0          0
Ethernet0  UC6  0            0              0          0
Ethernet0  UC7  0            0              0          0
Ethernet0  MC8  0            0              0          0
Ethernet0  MC9  0            0              0          0
Ethernet0  MC10 0           0              0          0
Ethernet0  MC11 0           0              0          0
Ethernet0  MC12 0           0              0          0
Ethernet0  MC13 0           0              0          0
Ethernet0  MC14 0           0              0          0
Ethernet0  MC15 0           0              0          0
```

```
sonic# show queue watermark unicast
Egress queue watermark per unicast queue:
```

```
Port      UC0  UC1  UC2  UC3  UC4  UC5  UC6  UC7
-----
Ethernet0  0    0    0    0    0    0    0    0
Ethernet4  0    0    0    0    0    0    0    0
Ethernet8  0    0    0    0    0    0    0    0
Ethernet12 0   0    0    0    0    0    0    0
```

```
Ethernet16 0 0 0 0 0 0 0 0  
Ethernet20 0 0 0 0 0 0 0 0  
Ethernet24 0 0 0 0 0 0 0 0
```

```
sonic# show queue persistent-watermark multicast  
Egress queue persistent watermark per multicast queue:
```

Port	MC8	MC9	MC10	MC11	MC12	MC13	MC14	MC15
Ethernet0	0	0	0	0	0	0	0	0
Ethernet4	0	0	0	0	0	0	0	0
Ethernet8	0	0	0	0	0	0	0	0
Ethernet12	0	0	0	0	0	0	0	0
Ethernet16	0	0	0	0	0	0	0	0
Ethernet20	0	0	0	0	0	0	0	0
Ethernet24	0	0	0	0	0	0	0	0
Ethernet28	0	0	0	0	0	0	0	0

```
sonic# show queue wred-ecn counters interface Eth1/56
```

TxQ	WRED Drops/Pkts	ECN Marked/Pkts	ECN Marked/Bytes
UC0	0	0	0
UC1	0	0	0
UC2	0	837922938	107207451904
UC3	390634	0	0
UC4	400000	0	0
UC5	0	0	0
UC6	0	0	0
UC7	0	0	0
UC8	0	0	0
UC9	0	0	0

show radius-server

Description

Display RADIUS information

Syntax

```
show radius-server
```

show radius-server dynamic-author

Description

This command displays the dynamic authorization server parameters.

Syntax

```
show radius-server dynamic-author
```

Usage Guidelines

This command displays the dynamic authorization server parameters.

Examples

```
sonic-cl# show radius-server dynamic-author

AdminMode..... Disabled
Port..... 7676
Auth Type..... session-key
Global Secret Key..... U2FsdGVkX18LcQREyGJP/aDuWTi34jJLzdqNy9W5WxI=
Ignore Server Key..... Disabled
Ignore Session Key..... Disabled
CoA Bounce-Host-Port..... Accept
CoA Disable-Host-Port..... Accept

-----
Client Address          Secret
-----
7.7.7.7                M2FsdGVkX17LcQREyGJP/aDuWTi34jJLzdqNy9W5WxI=
test                   N/A
```

show radius-server dynamic-author statistics

Description

This command displays the DAS global and per client counters.

Syntax

```
show radius-server dynamic-author statistics [ client { all | <ipaddr_hostname> } ]
```

Parameters

Name	Description	Type
ipaddr_hostname	WORD	String

Usage Guidelines

This command displays the DAS global and per client counters.

Examples

```
sonic-cli# show radius-server dynamic-author statistics

Number of CoA Requests Received..... 79
Number of CoA ACK Responses Sent..... 17
Number of CoA NAK Responses Sent..... 77
Number of CoA Requests Ignored..... 78
Number of CoA Missing/Unsupported Attribute Requests.. 23
Number of CoA Session Context Not Found Requests..... 89
Number of CoA Invalid Attribute Value Requests..... 13
Number of Administratively Prohibited Requests..... 12
sonic-cli#


sonic-cli# show radius-server dynamic-author statistics client all

DAC Address..... 1.1.1.1
Number of CoA Requests Received..... 77
Number of CoA ACK Responses Sent..... 88
Number of CoA NAK Responses Sent..... 66
Number of CoA Requests Ignored..... 7
Number of CoA Missing/Unsupported Attribute Requests.. 3
Number of CoA Session Context Not Found Requests..... 4
Number of CoA Invalid Attribute Value Requests..... 2
Number of Administratively Prohibited Requests..... 8

DAC Address..... test
Number of CoA Requests Received..... 90
Number of CoA ACK Responses Sent..... 32
Number of CoA NAK Responses Sent..... 36
Number of CoA Requests Ignored..... 79
Number of CoA Missing/Unsupported Attribute Requests.. 50
Number of CoA Session Context Not Found Requests..... 28
Number of CoA Invalid Attribute Value Requests..... 39
Number of Administratively Prohibited Requests..... 45
sonic-cli#
```

show reboot-cause

Description

Display cause of most recent reboot

Syntax

```
show reboot-cause
```

show route-map

Description

Display route map information

Syntax

```
show route-map [ <rt-map-name> ]
```

Parameters

Name	Description	Type
rt-map-name	WORD	String

show running-configuration

Description

```
Current operating configuration
```

Syntax

```
show running-configuration
```

show running-configuration bfd

Description

```
Displays Bidirectional Forwarding detection(BFD) configurations.
```

Syntax

```
show running-configuration bfd
```

Examples

```
device# show running-configuration bfd
!
bfd
peer 192.168.2.1 interface Ethernet0
  detect-multiplier 5
  echo-interval 200
  echo-mode
  receive-interval 200
  transmit-interval 200
!
peer 192.168.2.1 multihop local-address 192.168.2.2
  detect-multiplier 4
  receive-interval 150
```

```
transmit-interval 150
!
```

show running-configuration bgp

Description

```
Display current BGP configurations
```

Syntax

```
show running-configuration bgp [ vrf <vrf-name-opt> ]
```

Parameters

Name	Description	Type
vrf-name-opt	WORD	String

show running-configuration bgp as-path-acess-list

Description

```
Display current BGP AS path acess list configuration
```

Syntax

```
show running-configuration bgp as-path-acess-list [ <aspath-list-name> ]
```

Parameters

Name	Description	Type
aspath-list-name	WORD	String

show running-configuration bgp community-list

Description

Display current BGP community-list configuration

Syntax

```
show running-configuration bgp community-list [ <community-list-name> ]
```

Parameters

Name	Description	Type
community-list-name	WORD	String

show running-configuration bgp extcommunity-list

Description

Display current BGP extended community-list configuration

Syntax

```
show running-configuration bgp extcommunity-list [ <extcommunity-list-name> ]
```

Parameters

Name	Description	Type
extcommunity-list-name	WORD	String

show running-configuration bgp neighbor

Description

Display current BGP neighbor configurations

Syntax

```
show running-configuration bgp neighbor vrf { <vrf-name> { [ <ip> ] | { [ interface { Ethernet  
| PortChannel | Vlan } ] } ] } }
```

Parameters

Name	Description	Type
vrf-name	WORD	String
ip	A.B.C.D/A::B	String

show running-configuration bgp peer-group

Description

```
Display current BGP peer group configurations
```

Syntax

```
show running-configuration bgp peer-group vrf { <vrf-name> [ <peer-group-name> ] }
```

Parameters

Name	Description	Type
vrf-name	WORD	String
peer-group-name	WORD	String

show running-configuration class-map

Description

```
Shows running configuration of class-map(s)
```

Syntax

```
show running-configuration class-map [ <show-fbs-class-name> ]
```

Parameters

Name	Description	Type
show-fbs-class-name	WORD	String

Usage Guidelines

Class-map name is optional. If Class-map name provided it will show all running configuration of all class-maps configured

Examples

```
sonic# show running-configuration class-map class_permit_ipv6
class-map class_permit_ipv6 match-type fields match-all
```

show running-configuration dropcounters

Description

Display dropcounters configuration

Syntax

```
show running-configuration dropcounters
```

show running-configuration interface

Description

Display interface information

Syntax

```
show running-configuration interface { [ <port> ] | { [ Eth [ <iface_num> ] ] } | { [ Ethernet
[ <iface_num> ] ] } }
```

Parameters

Name	Description	Type
port	EthernetNUM	
iface_num	EthernetSLOTPORT	

show running-configuration interface Loopback

Description

Display interface loopback information

Syntax

```
show running-configuration interface Loopback [ <lo-id> ]
```

Parameters

Name	Description	Type
lo-id		Integer

show running-configuration interface Management

Description

Display interface management information

Syntax

```
show running-configuration interface Management [ <mgmt-if-id> ]
```

Parameters

Name	Description	Type
mgmt-if-id		Integer

show running-configuration interface PortChannel

Description

Display interface PortChannel information

Syntax

```
show running-configuration interface PortChannel [ <po-id> ]
```

Parameters

Name	Description	Type
po-id		Integer

show running-configuration interface Vlan

Description

```
Display interface VLAN information
```

Syntax

```
show running-configuration interface Vlan [ <vlan-id> ]
```

Parameters

Name	Description	Type
vlan-id		Integer

show running-configuration interface vxlan

Description

```
Display VXLAN configuration
```

Syntax

```
show running-configuration interface vxlan
```

show running-configuration ip access-list

Description

Show current IPv4 ACL configuration.

Syntax

```
show running-configuration ip access-list [ <access-list-name> ]
```

Parameters

Name	Description	Type
access-list-name	String(Max: 72 characters)	String

Usage Guidelines

ACL Name is optional. If no ACL name is specified then configuration for all IPv4 ACLs will be displayed.

Examples

```
ip access-list ipacl-example
 seq 10 permit ip host 10.1.1.1 host 20.1.1.1
 seq 20 permit ip host 10.1.1.2 host 20.1.1.2
 seq 30 permit ip host 10.1.1.3 host 20.1.1.3
 seq 40 permit ip host 10.1.1.4 host 20.1.1.4
```

show running-configuration ip host-access-list

Description

Show current IPv4 host-ACL configuration.

Syntax

```
show running-configuration ip host-access-list [ <access-list-name> ]
```

Parameters

Name	Description	Type
access-list-name	String(Max: 72 characters)	String

Usage Guidelines

ACL Name is optional. If no ACL name is specified then configuration for all ctrl ACLs will be displayed.

Examples

```
ip host-access-list ipacl-example
  service SSH
  seq 10 permit ip 10.1.1.1/32 20.1.1.1/32
  seq 20 permit ip 10.1.1.2/32 20.1.1.2/32
  seq 30 permit ip 10.1.1.3/32 20.1.1.3/32
  seq 40 permit ip 10.1.1.4/32 20.1.1.4/32
```

show running-configuration ip prefix-list

Description

Display current IPv4 prefix-list configuration

Syntax

```
show running-configuration ip prefix-list [ <prefix-list-name> ]
```

Parameters

Name	Description	Type
prefix-list-name	WORD	String

show running-configuration ipv6 access-list

Description

Show current IPv6 ACL configuration.

Syntax

```
show running-configuration ipv6 access-list [ <access-list-name> ]
```

Parameters

Name	Description	Type
access-list-name	String(Max: 72 characters)	String

Usage Guidelines

ACL Name is optional. If no ACL name is specified then configuration for all IPv6 ACLs will be displayed.

Examples

```
ipv6 access-list ipv6acl-example
  seq 100 permit ipv6 host abcd::1 host bcde::1
  seq 200 permit tcp host abcd::2 host bcde::2
  seq 300 permit udp host abcd::3 host bcde::3
```

show running-configuration ipv6 host-access-list

Description

Show current ctrl ACL configuration.

Syntax

```
show running-configuration ipv6 host-access-list [ <access-list-name> ]
```

Parameters

Name	Description	Type
access-list-name	String(Max: 72 characters)	String

Usage Guidelines

ACL Name is optional. If no ACL name is specified then configuration for all ctrl ACLs will be displayed.

Examples

```
ipv6 host-access-list ipv6acl-example
  service SNMP
    seq 10 permit ipv6 fdbd:cc00::2/128 any
    seq 20 permit ipv6 fdbd:cc00::2/128 any
    seq 30 permit ipv6 fdbd:cc00::2/128 any
```

show running-configuration ipv6 prefix-list

Description

Display current IPv6 prefix-list configuration

Syntax

```
show running-configuration ipv6 prefix-list [ <prefix-list-name> ]
```

Parameters

Name	Description	Type
prefix-list-name	WORD	String

show running-configuration link state tracking

Description

Display current link state tracking configuration

Syntax

```
show running-configuration link state tracking [ <show-runn-grp-name> ]
```

Parameters

Name	Description	Type
show-runn-grp-name	name	String

show running-configuration mac access-list

Description

Show current MAC ACL configuration.

Syntax

```
show running-configuration mac access-list [ <access-list-name> ]
```

Parameters

Name	Description	Type
access-list-name	String(Max: 72 characters)	String

Usage Guidelines

ACL Name is optional. If no ACL name is specified then configuration for all MAC ACLs will be displayed.

Examples

```
mac access-list macacl-example
 seq 10 permit host 00:00:10:00:00:01 host 00:00:20:00:00:01
 seq 20 permit host 00:00:10:00:00:02 host 00:00:20:00:00:02
 seq 30 permit host 00:00:10:00:00:03 host 00:00:20:00:00:03
 seq 40 permit host 00:00:10:00:00:04 host 00:00:20:00:00:04
```

show running-configuration mclag

Description

Shows running configuration of mclag domain

Syntax

```
show running-configuration mclag
```

Examples

```
sonic# show running-configuration mclag
mclag domain 89
  source-ip 1.1.1.1
  peer-ip 1.1.1.2
  peer-link PortChannel12
  keepalive-interval 1
  session-timeout 30
```

show running-configuration mirror-session

Description

```
Display mirror session info
```

Syntax

```
show running-configuration mirror-session
```

show running-configuration nat

Description

```
Displays all NAT configurations
```

Syntax

```
show running-configuration nat
```

show running-configuration ospf

Description

```
Display all OSPFv2 router configurations
```

Syntax

```
show running-configuration ospf
```

show running-configuration ospf interface

Description

Displays all OSPFv2 interface configurations

Syntax

`show running-configuration ospf interface`

show running-configuration ospf6

Description

Display all OSPFv3 router configurations

Syntax

`show running-configuration ospf6`

show running-configuration ospf6 interface

Description

Displays all OSPFv3 interface configurations

Syntax

`show running-configuration ospf6 interface`

show running-configuration pac

Description

Shows running-config for Port Access Control

Syntax

`show running-configuration pac`

Usage Guidelines

This command is useful in viewing a summary of running-config for Port Access control

Examples

```
sonic-cl# show running-configuration pac
dot1x system-auth-control
!
interface Ethernet0
    dot1x pae authenticator
    authentication order dot1x mab
    authentication priority dot1x mab
    authentication host-mode multi-auth
    authentication event fail action authorize vlan 30
    authentication event no-response action authorize vlan 30
    mab
!
interface Ethernet10
    dot1x pae authenticator
    authentication order dot1x mab
```

show running-configuration pbf next-hop-group

Description

Display current PBF next-hop groups configuration

Syntax

```
show running-configuration pbf next-hop-group [ <show-fbs-group-name> ]
```

Parameters

Name	Description	Type
show-fbs-group-name	WORD	String

show running-configuration pbf replication-group

Description

Display current PBF replication groups configuration

Syntax

```
show running-configuration pbf replication-group [ <show-fbs-group-name> ]
```

Parameters

Name	Description	Type
show-fbs-group-name	WORD	String

show running-configuration policy-map

Description

Shows running configuration of policy-map(s)

Syntax

```
show running-configuration policy-map [ <show-fbs-policy-name> ]
```

Parameters

Name	Description	Type
show-fbs-policy-name	WORD	String

Usage Guidelines

Policy-map name is optional. If policy-map name provided it will show all running configuration of all policy-maps configured

Examples

```
sonic# show running-configuration policy-map policy_vrf
policy-map policy_vrf type forwarding
  class class_permit_ipv6 priority 10
    set ipv6 next-hop 1211::2 priority 20
    set ipv6 next-hop 1212::2 vrf Vrf-BLUE priority 30
  class class_permit_ip priority 10
    set ip next-hop 12.12.1.2 vrf default priority 30
    set ip next-hop 12.12.2.2 vrf Vrf-BLUE priority 20
    set ip next-hop 12.12.1.2 priority 10
```

show running-configuration route-map

Description

Display current route map configuration

Syntax

```
show running-configuration route-map [ <rt-map-name> [ <seq-nu> ] ]
```

Parameters

Name	Description	Type
rt-map-name	WORD	String
seq-nu		Integer

show running-configuration subinterface

Description

Display running config of subinterfaces or the specified subinterface

Syntax

```
show running-configuration subinterface { [ <subifname> ] | { [ PortChannel <pch_num> ] } }
```

Parameters

Name	Description	Type
subifname	EthernetX.Y	
pch_num	X.Y	String

Usage Guidelines

```
show running-configuration subinterface <subinterface-id>
```

Examples

```
sonic-cli# show running-configuration subinterface
!
interface Ethernet0.10
  encapsulation dot1q vlan-id 100
  mtu 2000
  no shutdown
  ip address 10.10.1.1/24
!
interface PortChannel100.100
  encapsulation dot1q vlan-id 1350
  no shutdown
  ip vrf forwarding Vrf1
sonic-cli#
```

```
sonic-cli# show running-configuration subinterface PortChannel 100.100
!
interface PortChannel100.100
  encapsulation dot1q vlan-id 1350
  no shutdown
  ip vrf forwarding Vrf1
sonic-cli#
```

Features this CLI belongs to

- Subinterface

show running-configuration tam

Description

This command is used to show TAM running configuration.

Syntax

```
show running-configuration tam
```

Usage Guidelines

Use this command to view TAM running configuration.

Examples

```
sonic-cli# show running-configuration tam
!
!
tam
switch-id 3232
enterprise-id 434
```

```
collector c1 ip 1.1.1.1 port 1111 protocol UDP
sampler s1 rate 1
sampler s2 rate 655
sampler s4 rate 65550
sampler s5 rate 999999999
flow-group f1 src-ip 10.1.1.10/24 dst-ip 30.1.1.10/24 protocol TCP l4-src-port 8080 priority
    100
flow-group f2 src-ip 10.1.1.10/32 dst-ip 30.1.1.10/32 protocol UDP priority 100
!
drop-monitor
    aging-interval 23
!
ifa
    session ifa1 flowgroup f1 collector c1 node-type EGRESS
```

show running-configuration telemetry

Description

This command is used to show telemetry running configuration.

Syntax

```
show running-configuration telemetry
```

Usage Guidelines

Use this command to view telemetry running configuration.

Examples

show running-configuration vrf

Description

Show VRF configuration information

Syntax

```
show running-configuration vrf <vrf-name>
```

Parameters

Name	Description	Type
vrf-name	WORD	String

show service-policy

Description

Shows Global/Switch level flow based services applied policies information

Syntax

```
show service-policy { Switch | CtrlPlane } [ type { qos | monitoring | forwarding | copp | acl-copp } ]
```

Usage Guidelines

Examples

```
Sonic# show service-policy Switch
Switch
  Policy policy_vrf type forwarding at ingress
    Description:
      Flow class_permit_ipv6 at priority 10 (Inactive)
        Description:
          set ipv6 nexthop 1211::2 priority 20
          set ipv6 nexthop 1212::2 vrf Vrf-BLUE priority 30
          Packet matches: 0 frames 0 bytes
      Flow class_permit_ip at priority 10 (Inactive)
        Description:
          set ip nexthop 12.12.1.2 vrf default priority 30
          set ip nexthop 12.12.2.2 vrf Vrf-BLUE priority 20
          set ip nexthop 12.12.1.2 priority 10
          Packet matches: 0 frames 0 bytes
```

show service-policy interface

Description

Shows flow based services applied policies information by interface name

Syntax

```
show service-policy interface { <eth-if-id> | <po-if-id> | <vlan-if-id> | <eth-sub-if-id> |  
    <po-sub-if-id> | CPU } [ type { qos | monitoring | forwarding | copp | acl-copp } ]
```

Parameters

Name	Description	Type
eth-if-id	EthernetNUM	
po-if-id	PortChannelNUM	
vlan-if-id	VlanNUM	
eth-sub-if-id	EthernetX.Y	
po-sub-if-id	PortChannelNUM	

Usage Guidelines

policy-map type is optional. If type is not specified then all policies applied to this given interface will be shown. If type is also provided then only corresponding type policies applied on the given interface will be shown

Examples

```
show service-policy interface Vlan 100  
Vlan100  
Policy policy_mirror type monitoring at ingress  
Description:  
    Flow class1 at priority 10 (Active)  
        Description:  
            Packet matches: 0 frames 0 bytes  
Policy policy_vrf type forwarding at ingress  
Description:  
    Flow class_permit_ipv6 at priority 10 (Inactive)  
        Description:  
            set ipv6 nexthop 1211::2 priority 20  
            set ipv6 nexthop 1212::2 vrf Vrf-BLUE priority 30  
            Packet matches: 0 frames 0 bytes  
    Flow class_permit_ip at priority 10 (Active)  
        Description:  
            set ip nexthop 12.12.1.2 vrf default priority 30  
            set ip nexthop 12.12.2.2 vrf Vrf-BLUE priority 20  
            set ip nexthop 12.12.1.2 priority 10  
            Packet matches: 0 frames 0 bytes  
Policy policy_qos type qos at egress  
Description:  
    Flow class_permit_ipv6 at priority 10 (Inactive)  
        Description:  
            police: cir 300000000 cbs 300000000 pir 300000000 pbs 300000000 (Active)  
                type bytes mode color-blind  
                operational cir 0 cbs 0 pir 0 pbs 0  
                green 0 packets 0 bytes action forward
```

```

yellow 0 packets 0 bytes action forward
red 0 packets 0 bytes action drop
Packet matches: 0 frames 0 bytes
Flow class_permit_ip at priority 10 (Inactive)
Description:
police: cir 3000000000 cbs 3000000000 pir 3000000000 pbs 3000000000 (Active)
    type bytes mode color-blind
    operational cir 0 cbs 0 pir 0 pbs 0
    green 0 packets 0 bytes action forward
    yellow 0 packets 0 bytes action forward
    red 0 packets 0 bytes action drop
Packet matches: 0 frames 0 bytes

```

show service-policy policy-map

Description

Shows flow based services applied policies information by policy name

Syntax

```
show service-policy policy-map <fbs-policy-name> { { [ interface { <eth-if-id> | <po-if-id> |
<vlan-if-id> | <eth-sub-if-id> | <po-sub-if-id> | CPU } ] } | [ Switch ] | [ CtrlPlane ] } ]
```

Parameters

Name	Description	Type
fbs-policy-name	WORD	String
eth-if-id	EthernetNUM	
po-if-id	PortChannelNUM	
vlan-if-id	VlanNUM	
eth-sub-if-id	EthernetX.Y	
po-sub-if-id	PortChannelNUM	

Usage Guidelines

policy-map interface argument is optional. If interface is not specified then it shows all services applied interfaces information for given policy. If interface is also specified then it gives service applied information for given policy name and given interface

Examples

```

show service-policy policy-map policy_vrf
Vlan100
Policy policy_vrf type forwarding at ingress
Description:
Flow class_permit_ipv6 at priority 10 (Inactive)
Description:
set ipv6 nexthop 1211::2 priority 20
set ipv6 nexthop 1212::2 vrf Vrf-BLUE priority 30
Packet matches: 0 frames 0 bytes
Flow class_permit_ip at priority 10 (Inactive)
Description:
set ip nexthop 12.12.1.2 vrf default priority 30
set ip nexthop 12.12.2.2 vrf Vrf-BLUE priority 20
set ip nexthop 12.12.1.2 priority 10
Packet matches: 0 frames 0 bytes
Switch
Policy policy_vrf type forwarding at ingress
Description:
Flow class_permit_ipv6 at priority 10 (Inactive)
Description:
set ipv6 nexthop 1211::2 priority 20
set ipv6 nexthop 1212::2 vrf Vrf-BLUE priority 30
Packet matches: 0 frames 0 bytes
Flow class_permit_ip at priority 10 (Inactive)
Description:
set ip nexthop 12.12.1.2 vrf default priority 30
set ip nexthop 12.12.2.2 vrf Vrf-BLUE priority 20
set ip nexthop 12.12.1.2 priority 10
Packet matches: 0 frames 0 bytes

```

show service-policy summary

Description

Shows summary of applied flow based services policies

Syntax

```

show service-policy summary { [ interface { <eth-if-id> | <po-if-id> | <vlan-if-id> |
<eth-sub-if-id> | <po-sub-if-id> | CPU } ] } | [ Switch ] | [ CtrlPlane ] } ] [ type { qos
| monitoring | forwarding | copp | acl-copp } ]

```

Parameters

Name	Description	Type
eth-if-id	EthernetNUM	
po-if-id	PortChannelNUM	
vlan-if-id	VlanNUM	
eth-sub-if-id	EthernetX.Y	
po-sub-if-id	PortChannelNUM	

Usage Guidelines

Interface argument is optional. If interface is not specified it shows all service applied interfaces and their policy information. If interface is specified it shows service applied policy information for given interface. If interface is Switch it shows global/Switch level service policies.

Examples

```
sonic# show service-policy summary
Vlan100
    monitoring policy policy_mirror at ingress
    forwarding policy policy_vrf at ingress
    qos policy policy_qos at egress
Switch
    forwarding policy policy_vrf at ingress
CtrlPlane
    qos policy oob-qos-policy at ingress
```

show sflow

Description

Show global sFlow configuration

Syntax

```
show sflow
```

Examples

```
sonic# show sflow
-----
Global sFlow Information
-----
admin state:      up
polling-interval: 20
agent-id:        default
sonic#
```

show sflow counters

Description

Show sFlow counters

Syntax

```
show sflow counters
```

Examples

```
sonic# show sflow counters
```

Type	Count
Total Samples Packets	135
Total Samples Bytes	135000000
Processed Samples Packets	100
Processed Samples Bytes	100000000
Dropped Samples Packets	5
Dropped Samples Bytes	5000000

```
sonic#
```

show sflow interface

Description

```
Show sFlow interface configuration
```

Syntax

```
show sflow interface
```

Examples

```
sonic# show sflow interface
```

sFlow interface configurations			
Interface	Admin State	Sampling Rate	Sampling Direction
Ethernet0	up	4000	rx
Ethernet1	up	4000	rx
Ethernet2	up	4000	rx
Ethernet3	up	4000	rx
Ethernet4	up	4000	rx
Ethernet5	up	4000	rx
Ethernet6	up	4000	rx
Ethernet7	up	4000	rx
Ethernet8	up	4000	rx
Ethernet9	up	4000	rx
Ethernet10	up	4000	rx
Ethernet11	up	4000	rx
Ethernet12	up	4000	rx
Ethernet13	up	4000	rx
Ethernet14	up	4000	rx
Ethernet15	up	4000	rx
Ethernet16	up	4000	rx

```

Ethernet17      up          4000          rx
Ethernet18      up          4000          rx
Ethernet19      up          4000          rx
Ethernet20      up          4000          rx
Ethernet21      up          4000          rx
--more--
Ethernet22      up          4000          rx
Ethernet23      up          4000          rx
Ethernet24      up          4000          rx
Ethernet25      up          4000          rx
Ethernet26      up          4000          rx
Ethernet27      up          4000          rx
Ethernet28      up          4000          rx
Ethernet29      up          4000          rx
sonic#

```

show snmp-server

Description

Displays the Simple Network Management Protocol (SNMP) server information including the physical location of the switch, the organization responsible for the network, SNMP engine identification, trap status and the agent addresses, if configured. SNMP engine identification is derived from the device MAC address on an initial boot.

Syntax

```
show snmp-server
```

Usage Guidelines

Use this command to view global SNMP server information.

Examples

```

sonic# show snmp-server

Location    : Lab1, Rack-10
Contact     : Broadcom Support
EngineID    : 8000013703525400f6817e
Traps       : enable

Agent Addresses:

      IP Address          UDP Port          Interface
----- 1.2.3.4            161
      1.2.3.4            1024
      1.2.3.5            1024          Ethernet10

```

show snmp-server community

Description

Displays the SNMP communities configured on the switch and the community group association, if configured. Communities are used by SNMPv2 protocol to access the switch.

Syntax

```
show snmp-server community
```

Usage Guidelines

Use this command to view the configured SNMP communities.

Examples

```
sonic# show snmp-server community
```

Community Name	Group Name
comm1	group-lab
comm2	None

show snmp-server group

Description

Displays the SNMP groups configured on the switch. The model and security information indicate the SNMP protocol and security level used to access the system via the group. View names indicate the view that a group provides read, write or trap access to.

Syntax

```
show snmp-server group
```

Usage Guidelines

Use this command to view the configured SNMP groups.

Examples

```
sonic# show snmp-server group
```

Group Name	Model: Security	Read View	Write View	Notify View
group-floor1	v2c: no-auth-no-priv	ro_view	wr_view	None
group-floor2	v3 : auth-priv	r_view	None	None
group-lab	v2c: no-auth-no-priv	None	None	None

show snmp-server host

Description

Displays the SNMP hosts to which the trap or inform messages are sent by the SNMP agent. Timeout indicates the number of seconds before the traps/informs time out when sending to a host.

Retries indicate the number of times the traps/informs are sent after timing out.

Syntax

```
show snmp-server host
```

Usage Guidelines

Use this command to view the configured SNMP hosts.

Examples

```
sonic# show snmp-server host
```

Target Address	Type	Community	Ver	T-Out	Retries
1.2.3.4	trap	comm1	v2c	15	3

Target Address	Type	User Name	Security	T-Out	Retries
1001::1	inform	user1	auth-priv	200	10

show snmp-server user

Description

Displays the SNMPv3 users configured on the switch including any authentication and/or encryption algorithm for the user. The group name indicates a group that defines the SNMPv3 access parameters.

Syntax

```
show snmp-server user
```

Usage Guidelines

Use this command to view the configured SNMPv3 users.

Examples

```
sonic# show snmp-server user
```

User Name	Group Name	Auth	Privacy
user1	group-lab	md5	aes-128
user2	group-floor2	None	None

show snmp-server view

Description

Displays the SNMP views configured on the switch including the OID tree that the view includes or excludes.

Syntax

```
show snmp-server view
```

Usage Guidelines

Use this command to view the configured SNMP views.

Examples

```
sonic# show snmp-server view
```

View Name	OID Tree	Type
view1	1.2.3.4.5.6.7.8.9.1	included
view2	1.2.3.4.5.6.7.8.9.5.1	excluded

show sonic-log-configuration

Description

Display components registered in DB for loglevel severity

Syntax

```
show sonic-log-configuration
```

show ssh

Description

Displays the configurable SSH daemon options.

Syntax

```
show ssh
```

Usage Guidelines

```
sonic-cli# show ssh
```

Examples

```
sonic-cli(config)# show ssh
```

```
-----  
SSH Global Configuration  
-----
```

```
Login Attempts: 6  
Login Timeout: 120  
Port: 22  
TCP Forwarding: yes  
X11 Forwarding: yes
```

show ssh-server vrf

Description

Shows list of VRFs on which ssh server is enabled

Syntax

```
show ssh-server vrf { all | <vrf-name> }
```

Parameters

Name	Description	Type
vrf-name	WORD	String

Examples

```
sonic# show ssh-server vrf all
      VRF Name          Status
-----
  Vrf1                  enable
  Vrf34                 enable
  Vrf5                  enable

sonic# show ssh-server vrf Vrf5
      VRF Name          Status
-----
  Vrf5                  enable
```

show ssh-server vrfs

Description

Display SSH server status for all VRFs

Syntax

```
show ssh-server vrfs
```

show startup-configuration

Description

Contents of startup configuration

Syntax

```
show startup-configuration
```

show storm-control

Description

show command to display all BUM storm-control configurations.

Syntax

```
show storm-control
```

Usage Guidelines

```
sonic-cl# show storm-control
```

Examples

```
sonic-cli# show storm-control
-----
Interface Name      Storm-type        Rate (kbps)
-----
Ethernet0           broadcast          10000
Ethernet0           unknown-unicast   20000
Ethernet0           unknown-multicast 30000
Ethernet4           unknown-unicast   20000
Ethernet4           unknown-multicast 30000
```

show storm-control interface

Description

```
show command to display BUM storm-control configurations on the given interface.
```

Syntax

```
show storm-control interface <Ethernet>
```

Parameters

Name	Description	Type
Ethernet	EthernetNUM	

Usage Guidelines

```
sonic-cli# show storm-control interface Ethernet 4
```

Examples

```
sonic-cli# show storm-control interface Ethernet 4
-----
Interface Name      Storm-type        Rate (kbps)
-----
Ethernet4           unknown-unicast   20000
Ethernet4           unknown-multicast 30000
```

show subinterfaces status

Description

Display subinterface status in tabular format

Syntax

```
show subinterfaces status
```

Usage Guidelines

```
show subinterfaces status
```

Examples

```
sonic-cli# show subinterfaces status
-----
Sub port interface    Speed      MTU      Vlan      Admin      Type
-----
Ethernet0.10          25000     2000     100       up        dot1q-encapsulation
PortChannel100.100    0          9100     1350     up        dot1q-encapsulation
sonic-cli#
```

Features this CLI belongs to

- Subinterface

show switch-profiles

Description

Use this command to list all supported factory default configuration profiles. It also displays the current active default configuration profile. The current active profile name is used to create a startup configuration file when the write erase command is used. The following information about each default configuration profile is displayed.

- Name: Profile name used to refer to a default configuration profile
- Description: A descriptive string

Syntax

```
show switch-profiles
```

Usage Guidelines

```
show factory default profiles
```

Examples

```
sonic# show switch-profiles
Factory Default: 13

Profile Name      Description
-----
12                Layer 2 Switch Configuration
13                Layer 3 Router Configuration
```

show switch-resource drop-monitor

Description

```
Show warm restart
```

Syntax

```
show switch-resource drop-monitor
```

Examples

show switch-resource route-scale

Description

```
Show warm restart
```

Syntax

```
show switch-resource route-scale
```

Examples

show system

Description

```
Display system information
```

Syntax

```
show system
```

show system cpu

Description

```
Display system cpu information
```

Syntax

```
show system cpu
```

show system cpu history

Description

```
Display CPU usage history
```

Syntax

```
show system cpu history
```

show system memory

Description

```
Display system memory information
```

Syntax

```
show system memory
```

show system processes

Description

```
Display system processes
```

Syntax

```
show system processes
```

show system processes cpu

Description

Display system processes sorted by CPU usage

Syntax

`show system processes cpu`

show system processes mem-usage

Description

Display system processes sorted by memory usage

Syntax

`show system processes mem-usage`

show system processes mem-util

Description

Display system processes sorted by memory utilization

Syntax

`show system processes mem-util`

show system processes pid

Description

Display system process information of a particular PID

Syntax

`show system processes pid <pid-no>`

Parameters

Name	Description	Type
pid-no	Process ID	Integer

show system status

Description

```
Display system status
```

Syntax

```
show system status
```

show system vlan

Description

```
Display reserved vlan information
```

Syntax

```
show system vlan reserved
```

show tacacs-server

Description

```
Display TACACS information
```

Syntax

```
show tacacs-server
```

show tacacs-server global

Description

```
Display TACACS global configuration
```

Syntax

```
show tacacs-server global
```

show tacacs-server host

Description

Display TACACS server configuration

Syntax

```
show tacacs-server host [ <address> ]
```

Parameters

Name	Description	Type
address	WORD	String

show tam collectors

Description

This command lists the details for all collectors or for a specific collector.

Syntax

```
show tam collectors [ <name> ]
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

This command lists the details for all collectors or for a specific collector.

Examples

Name	IP Address	Port	Protocol
IFA_Col_i19	192.168.78.121	7071	UDP
MOD_Col_m16	192.168.78.123	7076	UDP

```
# show tam collectors IFA_Col_i19
Name      : IFA_Col_i19
IP Address : 192.168.78.121
Port       : 7071
Protocol   : UDP
```

show tam drop-monitor

Description

This command lists the switch-wide attributes that are in use.

Syntax

```
show tam drop-monitor
```

Usage Guidelines

This command lists the switch-wide attributes that are in use.

Examples

```
Status      : Active
Switch ID   : 2020
Aging Interval : 20
```

show tam drop-monitor sessions

Description

This command lists the details for all drop-monitor sessions or for a specific session.

Syntax

```
show tam drop-monitor sessions [ <name> ]
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

This command lists the details for all drop-monitor sessions or for a specific session. Note that only explicitly configured tuples in the associated flow-group are displayed.

Examples

Name	Flow Group	Collector	Sampler
http_236	tcp_port_236	Col_i19	aggresive
http_239	tcp_port_239	Col_i19	aggresive
http_241	tcp_port_241	Col_i19	aggresive

```
sonic # show tam drop-monitor sessions http_236

Session          : http_236
Flow Group Name : tcp_port_236
  Id            : 4025
  Priority      : 100
  SRC IP        : 13.92.96.32
  DST IP        : 7.72.235.82
  DST L4 Port   : 236
  Ingress Intf : Ethernet20
  Collector     : Col_i19
  Sampler       : aggresive
  Packet Count  : 7656
```

show tam features

Description

This command lists the current status for all TAM features or for a specific feature.

Syntax

```
show tam features { [ ifa ] | [ drop-monitor ] | [ tail-stamping ] } ]
```

Usage Guidelines

This command lists the current status for all TAM features or for a specific feature.

Examples

```
Name      Status  
-----  
ifa      Active  
drop-monitor Active  
  
sonic-clishow tam features ifa  
  
Name      : ifa  
Status    : Active
```

show tam flowgroups

Description

This command lists the details for all flow-groups or for a specific flow-group.

Syntax

```
show tam flowgroups [ <name> ]
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

This command lists the details for all flow-groups or for a specific flow-group. Note that only explicitly configured tuples are displayed.

Examples

```
Flow Group Name   : udp_port_239  
  Id             : 4025  
  Priority       : 100  
  SRC IP         : 10.72.195.23  
  DST L4 Port    : 239  
  Ingress Intf  : Ethernet20  
Packet Count     : 10584  
  
Flow Group Name   : udp_port_241  
  Id             : 4022  
  Priority       : 99
```

```
SRC Port      : 1906
DST L4 Port   : 241
Packet Count   : 8654367

sonic # show tam flow-groups udp_port_239

Flow Group Name    : udp_port_239
  Id              : 4025
  Priority        : 100
  SRC IP          : 10.72.195.23
  DST L4 Port     : 239
  Packet Count    : 10584
```

show tam ifa

Description

This command lists the switch-wide attributes that are in use.

Syntax

```
show tam ifa
```

Usage Guidelines

This command lists the switch-wide attributes that are in use.

Examples

```
Status      : Active
Version     : 2.0
Switch ID   : 2020
Enterprise ID : 2345
```

show tam ifa sessions

Description

This command lists the details for all ifa sessions or for a specific session.

Syntax

```
show tam ifa sessions [ <name> ]
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

This command lists the details for all ifa sessions or for a specific session. Note that only explicitly configured tuples in the associated flow-group are displayed.

Examples

Name	Flow Group	Collector	Sampler	Node Type
http_236	tcp_port_236	-	aggresive	Ingress
http_239	tcp_port_239	-	-	Intermediate
http_241	tcp_port_241	IFA_Col_i19	-	Egress

sonic # show tam ifa sessions http_236

Session : http_236 (Ingress)
Flow Group Name : tcp_port_236
 Id : 4025
 Priority : 100
 SRC IP : 13.92.96.32
 DST IP : 7.72.235.82
 DST L4 Port : 236
 Ingress Intf : Ethernet20
Collector : None
Sampler : aggresive
Packet Count : 7656

show tam samplers

Description

This command lists the details for all samplers or for a specific sampler.

Syntax

```
show tam samplers [ <name> ]
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

This command lists the details for all samplers or for a specific sampler.

Examples

```
Name      Sample Rate
-----
sflow_low   1
aggresive   2000

# show tam samplers aggresive
Name      : aggresive
Sample Rate : 2000
```

show tam switch

Description

This command is used to show TAM device information.

Syntax

```
show tam switch
```

Usage Guidelines

This command displays the configured TAM Device Identifier.

Examples

```
TAM Device information
-----
Switch ID      : 23456
Enterprise ID  : 1234
```

show tam tail-stamping

Description

This command lists the switch-wide attributes that are in use.

Syntax

```
show tam tail-stamping
```

Usage Guidelines

This command lists the switch-wide attributes that are in use.

Examples

Status	:	Active
Switch ID	:	2020

show tam tail-stamping sessions

Description

This command lists the details for all tail-stamping sessions or for a specific session.

Syntax

```
show tam tail-stamping sessions [ <name> ]
```

Parameters

Name	Description	Type
name	WORD	String

Usage Guidelines

This command lists the details for all tail-stamping sessions or for a specific session. Note that only explicitly configured tuples in the associated flow-group are displayed.

Examples

```
Name          Flow Group
-----
http_236      tcp_port_236
http_239      tcp_port_239
http_241      tcp_port_241
```

```
sonic # show tam tail-stamping sessions http_236
```

```
Session        : http_236
Flow Group Name : tcp_port_236
  Id           : 4025
  Priority     : 100
  SRC IP       : 13.92.96.32
  DST IP       : 7.72.235.82
  DST L4 Port  : 236
Packet Count   : 7656
```

show tech-support

Description

```
Collect technical support information
```

Syntax

```
show tech-support
```

show techsupport-export

Description

```
Display Techsupport Export Config
```

Syntax

```
show techsupport-export
```

show telemetry data collector details

Description

```
This command is used to display collection status per sensor path
```

Syntax

```
show telemetry data collector details
```

Usage Guidelines

Use this command to display collection status per sensor path

Examples

sonic# show telemetry data collector details	Row Id	Successful	Failed	Skipped	Sensor Path (Destination Group)
	0	287	0	0	platform/cpu (HS)
	1	286	0	0	openconfig-interfaces:interfaces/interface/Ethernet0/config (HS)
	2	973	0	3	openconfig-interfaces:interfaces/interface/Ethernet0/config (dg2)
	3	0	186	0	openconfig-interfaces:interfaces/interface/Ethernet1/config (dg2)

show telemetry destination-groups

Description

This command is used to show the telemetry destination groups.

Syntax

```
show telemetry destination-groups
```

Usage Guidelines

Use this command to show the telemetry destination groups.

Examples

```
sonic# show telemetry destination-groups
Name:          dg1
Destination addresses: 1.1.1.1:1111
Name:          dg2
Destination addresses: 2.2.2.2:2222,3.3.3.3:3333
```

show telemetry global

Description

This command is used to show the telemetry client global configuration.

Syntax

```
show telemetry global
```

Usage Guidelines

Use this command to show the telemetry client global configuration.

Examples

```
sonic# show telemetry global
encoding:      JSON_IETF
retry_interval: 30
src_ip:        1.1.1.1
unidirectional: True
```

show telemetry sensor-group

Description

This command is used to show created sensor groups.

Syntax

```
show telemetry sensor-group
```

Usage Guidelines

Use this command to list out created sensor groups.

Examples

```
sonic# show telemetry sensor-group
Sensor group ID: sg1
```

show telemetry sensor-path

Description

This command is used to show paths for each of the sensor groups.

Syntax

```
show telemetry sensor-path
```

Usage Guidelines

Use this command to list out paths for each of the sensor groups.

Examples

```
sonic# show telemetry sensor-path
Sensor group ID: sg1
Sensor path:      platform/cpu
Sensor group ID: sg1
Sensor path:      platform/fan
```

show telemetry subscription

Description

This command is used to show created subscriptions.

Syntax

```
show telemetry subscription
```

Usage Guidelines

Use this command to show created subscriptions.

Examples

```
sonic# show telemetry subscription
Name:          s1
Destination group: dg1
Path target:    OC_YANG
Paths:
Sensor group:  sg1
Report interval: 5000
Report type:   periodic
```

show telemetry transport

Description

This command is used to show whether this device is connected successfully with collector

Syntax

```
show telemetry transport
```

Usage Guidelines

Use this command to show whether this device is connected successfully with collector

Examples

Session Id	IP Address	Port	Encoding	Transport	Status
0	127.0.0.1	9081	JSON	gRPC	Connected
1	127.0.0.1	9082	JSON	gRPC	Connected

show threshold breaches

Description

This command is used to show information about threshold breach events recorded by the system.

Syntax

```
show threshold breaches
```

Usage Guidelines

Use this command to display the threshold breach events recorded by the system.

Examples

Event-id	Buffer Time-stamp	Type	Port	Index	Breach Value(%)
2	2020-04-14-11:35:20	priority-group	shared	Ethernet0	7
3	2020-04-17-11:30:20	queue	unicast	Ethernet0	5

show uptime

Description

Display system uptime

Syntax

```
show uptime
```

show users

Description

Display users

Syntax

```
show users
```

show version

Description

Display version information

Syntax

```
show version
```

show vrrp

Description

Displays IPv4 VRRP information

Syntax

```
show vrrp [ interface { <ifname> { vrid <id> } } ]
```

Parameters

Name	Description	Type
ifname	Interface Type - Ranges	
id		Integer

Usage Guidelines

Use this command to display VRRP summary information of all instances on a system or to display VRRP detailed information of a particular instance.

Examples

```
sonic-cli(config)# show vrrp  
      or  
sonic-cli(config)# show vrrp interface Ethernet4 vrid 1
```

```
sonic# show vrrp  
      Interface_Name  VRID   State       VIP           Cfg_Prio   Curr_Prio  
      Ethernet4      1       Master     40.0.0.5      120        120  
      Ethernet8      2       Backup    80.0.0.5      100        100
```

```
sonic# show vrrp interface Ethernet4 vrid 1  
      Ethernet4, VRID 1  
      Version is 2  
      State is Master  
      Virtual IP address:  
          40.0.0.5  
      Virtual MAC address is 0000.5e00.0101  
      Track interface:  
          None  
      Configured Priority is 100, Current Priority is 100  
      Advertisement interval is 1 sec  
      Preemption is enabled
```

Features this CLI belongs to

- VRRP

show vrrp6

Description

Displays IPv6 VRRP information,

Syntax

```
show vrrp6 [ interface { <ifname> { vrid <id> } } ]
```

Parameters

Name	Description	Type
ifname	Interface Type - Ranges	
id		Integer

Usage Guidelines

Use this command to display VRRP summary information of all instances on a system or to display VRRP detailed information of a particular instance.

Examples

```
sonic-clia(config)# show vrrp6
      or
sonic-clia(config)# show vrrp6 interface Ethernet4 vrid 1
```

```
sonic# show vrrp
  Interface_Name  VRID   State       VIP           Cfg_Prio   Curr_Prio
  Ethernet4       1       Master     40::5         120        120
  Ethernet8       2       Backup    80::5         100        100
```

```
sonic# show vrrp interface Ethernet4 vrid 1
  Ethernet4, VRID 1
  Version is 3
  State is Master
  Virtual IP address:
    40::5
  Virtual MAC address is 0000.5e00.0201
  Track interface:
    None
  Configured Priority is 100, Current Priority is 100
  Advertisement interval is 1 sec
  Preemption is enabled
```

Features this CLI belongs to

- VRRP

show vxlan counters

Description

show command to display VXLAN tunnel counters per remote VTEP or all remotes.
SIP, DIP, creation source, operstatus are the columns.

Syntax

```
show vxlan counters [ <remote_ip_addr> ]
```

Parameters

Name	Description	Type
remote_ip_addr	A.B.C.D	String

Usage Guidelines

```
sonic# show vxlan counters
```

Examples

```
sonic-cl# show vxlan counters
```

```
Polling Rate : 8 seconds
```

Interface	RX_OK	RX_BPS	RX_PPS	RX_ERR
	TX_OK	TX_BPS	TX_PPS	TX_ERR
EVPN_4.4.4.4	0	0	0	N/A
0	0	0	N/A	
EVPN_4.4.4.5	0	0	0	N/A
0	0	0	N/A	

show vxlan interface

Description

show command to display the VXLAN global parameters.

Syntax

```
show vxlan interface
```

Usage Guidelines

```
sonic# show vxlan interface
```

Examples

```
- sonic# show vxlan interface

    VTEP Name      : vtep1
    VTEP Source IP : 1.1.1.1
    QoS Mode       : pipe (dscp:0)
    Source Interface : Loopback0
```

show vxlan remote mac

Description

Show command to display all the MACs learnt from the specified remote IP or all the remotes for the specified/all VLANs.
VLAN, MAC, RemoteVTEP, VNI, Type are the columns.

Syntax

```
show vxlan remote mac [ <remote_ip_addr> ]
```

Parameters

Name	Description	Type
remote_ip_addr	A.B.C.D	String

Usage Guidelines

```
sonic# show vxlan remote mac
```

Examples

```

sonic# show vxlan remote mac
+-----+-----+-----+-----+
| VLAN | MAC           | RemoteVTEP | VNI | Type |
+-----+-----+-----+-----+
| Vlan101 | 00:00:00:00:00:01 | 4.4.4.4    | 1001 | static |
+-----+-----+-----+-----+
| Vlan101 | 00:00:00:00:00:02 | 3.3.3.3    | 1001 | static |
+-----+-----+-----+-----+
| Vlan101 | 00:00:00:00:00:03 | 4.4.4.4    | 1001 | static |
+-----+-----+-----+-----+
| Vlan101 | 00:00:00:00:00:04 | 4.4.4.4    | 1001 | static |
+-----+-----+-----+-----+
| Vlan101 | 00:00:00:00:00:05 | 4.4.4.4    | 1001 | static |
+-----+-----+-----+-----+
| Vlan101 | 00:00:00:00:00:99 | 3.3.3.3    | 1001 | static |
+-----+-----+-----+-----+
Total count : 6

sonic# show vxlan remote mac 3.3.3.3
+-----+-----+-----+-----+
| VLAN | MAC           | RemoteVTEP | VNI | Type |
+-----+-----+-----+-----+
| Vlan101 | 00:00:00:00:00:02 | 3.3.3.3    | 1001 | static |
+-----+-----+-----+-----+
| Vlan101 | 00:00:00:00:00:99 | 3.3.3.3    | 1001 | static |
+-----+-----+-----+-----+
Total count : 2

```

show vxlan remote mac count

Description

Display number of remote MACs

Syntax

```
show vxlan remote mac count [ <remote_ip_addr> ]
```

Parameters

Name	Description	Type
remote_ip_addr	A.B.C.D	String

show vxlan remote vni

Description

Show command to display all the VLANs learnt from the specified remote IP or all the remotes.
VLAN, RemoteVTEP, VNI are the columns

Syntax

```
show vxlan remote vni [ <remote_ip_addr> ]
```

Parameters

Name	Description	Type
remote_ip_addr	A.B.C.D	String

Usage Guidelines

```
sonic# show vxlan remote vni
```

Examples

```
sonic# show vxlan remote vni

+-----+-----+-----+
| VLAN | RemoteVTEP | VNI |
+=====+=====+=====+
| Vlan101 | 3.3.3.3 | 1001 |
+-----+-----+-----+
| Vlan101 | 4.4.4.4 | 1001 |
+-----+-----+-----+
Total count : 2

sonic# show vxlan remote vni 3.3.3.3

+-----+-----+-----+
| VLAN | RemoteVTEP | VNI |
+=====+=====+=====+
| Vlan101 | 3.3.3.3 | 1001 |
+-----+-----+-----+
Total count : 1
```

show vxlan remote vni count

Description

Display the number of VLANs extended to remote VTEPs.

Syntax

```
show vxlan remote vni count [ <remote_ip_addr> ]
```

Parameters

Name	Description	Type
remote_ip_addr	A.B.C.D	String

show vxlan tunnel

Description

show command to display all the discovered tunnels.
SIP, DIP, creation source, operstatus are the columns.

Syntax

```
show vxlan tunnel
```

Usage Guidelines

```
sonic# show vxlan tunnel
```

Examples

```
sonic# show vxlan tunnel

+-----+-----+-----+-----+
| SIP   | DIP    | creation source | operstatus  |
+=====+=====+=====+=====+
| 2.2.2.2 | 4.4.4.4 | EVPN           | oper_up     |
+-----+-----+-----+-----+
| 2.2.2.2 | 3.3.3.3 | EVPN           | oper_up     |
+-----+-----+-----+-----+
Total count : 2
```

show vxlan tunnel count

Description

Display number of remote VTEPs

Syntax

```
show vxlan tunnel count
```

show vxlan vlanvnimap

Description

```
show command to display all the VLAN VNI mappings
```

Syntax

```
show vxlan vlanvnimap
```

Usage Guidelines

```
sonic# show vxlan vlanvnimap
```

Examples

```
sonic# show vxlan vlanvnimap

+-----+-----+
| VLAN | VNI |
+=====+=====+
| Vlan100 | 100 |
+-----+-----+
| Vlan101 | 101 |
+-----+-----+
Total count : 2
```

show vxlan vlanvnimap count

Description

```
shows number of VLAN VNI mappings
```

Syntax

```
show vxlan vlanvnimap count
```

show vxlan vrfvnimap

Description

```
show command to display all the VRF VNI mappings
```

Syntax

```
show vxlan vrfvnimap
```

Usage Guidelines

```
sonic# show vxlan vrfvnimap
```

Examples

```
sonic# show vxlan vrfvnimap

+-----+-----+
| VRF   |   VNI |
+=====+=====+
| Vrf1  |   600 |
+-----+-----+
Total count : 1
```

show vxlan vrfvnimap count

Description

```
shows number of VRF VNI mappings
```

Syntax

```
show vxlan vrfvnimap count
```

show warm-restart

Description

```
Show warm restart
```

Syntax

```
show warm-restart
```

Examples

```
sonic# show warm-restart
-----
Module          Restore_count  Status
-----
aclsvcd        0
bgp            0          disabled
dropmgrp       0
fdbsyncd       0          disabled
gearsyncd      0
ifamgrp        0
intfmgrp       0          disabled
iphelpermgr    0
l2mcmgrd       0
natsyncd       0
nbrmgrp        0
neighsyncd     0
orchagent      0          disabled
portmgrp       0
```

show warm-restart check

Description

Display warm restart check information

Syntax

```
show warm-restart check
```

show warm-restart config

Description

Display warm restart config information

Syntax

```
show warm-restart config
```

show warm-restart system

Description

Display warm restart system state information

Syntax

```
show warm-restart system
```

show watermark interval

Description

This command is used to display watermark snapshot interval configured in the system.

Syntax

```
show watermark interval
```

Usage Guidelines

Use this command to display watermark snapshot interval configured in the system.

Examples

```
sonic-cl# show watermark interval
      Snapshot interval : 220 seconds
```

show watermark telemetry

Description

This command is used to display watermark telemetry interval configured in the system.

Syntax

```
show watermark telemetry interval
```

Usage Guidelines

Use this command to display watermark telemetry interval configured in the system.

Examples

```
sonic-cl# show watermark telemetry interval
      interval : 220 seconds
```

show ztp-status

Description

Show the status of ZTP operation.

Syntax

```
show ztp-status
```

Usage Guidelines

Use this command to show the status of ZTP operation.

These are the possible current states or result of ZTP session:

- IN-PROGRESS: ZTP session is currently in progress. ZTP service is processing switch provisioning information.
- SUCCESS: ZTP service has successfully processed the switch provisioning information.
- FAILED: ZTP service has failed to process the switch provisioning information.
- Not Started: ZTP service has not started processing the discovered switch provisioning information.

These are the state and result of a configuration section:

- IN-PROGRESS: Corresponding configuration section is currently being processed.
- SUCCESS: Corresponding configuration section was processed successfully.
- FAILED: Corresponding configuration section failed to execute successfully.
- Not Started: ZTP service has not started processing the corresponding configuration section.
- DISABLED: Corresponding configuration section has been marked as disabled and will not be processed.

Examples

```
sonic# show ztp-status
=====
ZTP
=====
ZTP Admin Mode      : True
ZTP Service         : Inactive
ZTP Status          : Not Started
```

Features this CLI belongs to

- ZTP

shutdown

Description

Disable the interface

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
shutdown  
no shutdown
```

shutdown

Description

```
Disable the interface
```

Parent Commands (Modes)

```
interface Loopback <lo-id>
```

Syntax

```
shutdown  
no shutdown
```

shutdown

Description

```
Disable the Vlan interface
```

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
shutdown  
no shutdown
```

shutdown

Description

```
Disable the interface
```

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
shutdown  
no shutdown
```

shutdown

Description

```
Disable the interface
```

Parent Commands (Modes)

```
interface Management <mgmt-if-id>
```

Syntax

```
shutdown  
no shutdown
```

shutdown

Description

```
Disable the interface
```

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
shutdown  
no shutdown
```

shutdown

Description

```
Disable the interface
```

Parent Commands (Modes)

```
interface range create vlan_range_num  
interface range vlan_range_num
```

Syntax

```
shutdown  
no shutdown
```

shutdown

Description

```
Disable the interface
```

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]  
interface range po_range_num
```

Syntax

```
shutdown  
no shutdown
```

shutdown

Description

```
Shut Bidirectional Forwarding detection(BFD) peer administratively down.
```

Parent Commands (Modes)

```
peer <peer_ipv4>  
peer <peer_ipv6>  
peer [ interface ] <interfacename>  
peer [ local-address ] <local_ipv4>  
peer [ local-address ] <local_ipv6>  
peer [ multihop ]  
peer [ vrf ] <vriname>
```

Syntax

```
shutdown  
no shutdown
```

Usage Guidelines

This command will change the BFD session state to DOWN.

Examples

```
device()#configure terminal  
device(config)#bfd  
device(conf-bfd)# peer 192.168.0.5 interface Ethernet0  
device(conf-bfd-peer)# shutdown
```

shutdown

Description

Shut Bidirectional Forwarding detection(BFD) peer administratively down.

Parent Commands (Modes)

```
profile <profilename>
```

Syntax

```
shutdown  
no shutdown
```

Usage Guidelines

This command will change the BFD session state to DOWN.

Examples

```
device()#configure terminal  
device(config)#bfd  
device(conf-bfd)# profile fast  
device(conf-bfd-profile)# shutdown
```

shutdown

Description

Configures admin status for routed subinterface

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
shutdown
no shutdown
```

Usage Guidelines

```
[no] shutdown
```

Examples

```
sonic-cli# configure terminal
sonic-cli(config)# interface Ethernet 0.10
sonic-cli(conf-subif-Ethernet0.10)# no shutdown
sonic-cli(config)#
```

```
sonic-cli# configure terminal
sonic-cli(config)# interface Ethernet 0.10
sonic-cli(conf-subif-Ethernet0.10)# shutdown
sonic-cli(config)#
```

Features this CLI belongs to

- Subinterface ## shutdown ##### Description

This command administratively shutsdown a BGP neighbor. The CLI allows user to specify a shutdown message that can be communicated to the neighbor

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
shutdown [ message <MSG> ]
no shutdown [ message <MSG> ]
```

Parameters

Name	Description	Type
MSG	String	String

Usage Guidelines

Use this command to administratively shutdown a BGP neighbor sessions

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# shutdown
```

shutdown

Description

This command administratively shutsdown a BGP peer-group. The CLI allows user to specify a shutdown message that can be communicated to the neighbors in peer-group

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
shutdown [ message <MSG> ]
no shutdown [ message <MSG> ]
```

Parameters

Name	Description	Type
MSG	String	String

Usage Guidelines

Use this command to administratively shutdown a BGP peer-group sessions

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# shutdown
```

snmp-server agentaddress

Description

Configure one or more SNMP agent addresses and optionally set the UDP port number on which the SNMP server listens for requests, and the Virtual Routing and Forwarding (VRF) interface used by the management station to access SNMP. The default UDP port is 161.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
snmp-server agentaddress <host-addr> { { [ port <udp-port> ] } { [ interface { Ethernet | Loopback | Management | PortChannel | Vlan | SubInterface } ] } }
no snmp-server agentaddress <host-addr> { { [ port <udp-port> ] } { [ interface <ifname> ] } }
```

Parameters

Name	Description	Type
host-addr	A.B.C.D or A:B:C:D:E:F:G:H	String
udp-port		Integer

Usage Guidelines

Use this command to configure an SNMP agent address, UDP port number, and VRF interface.

Examples

```
sonic# configure terminal  
sonic(config)# snmp-server agentaddress 1.2.3.4
```

```
sonic# configure terminal  
sonic(config)# snmp-server agentaddress 1.2.3.4 port 1024
```

```
sonic# configure terminal  
sonic(config)# snmp-server agentaddress 1.2.3.5 port 1024 interface Management0
```

snmp-server community

Description

Configure one or more SNMP communities and optionally associate them with a group.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
snmp-server community <community-name> { [ group <group-name> ] }  
no snmp-server community <community-name>
```

Parameters

Name	Description	Type
community-name	WORD	String
group-name	String(Max: 32 characters)	String

Usage Guidelines

Use this command to configure an SNMP community and group.

Examples

```
sonic# configure terminal  
sonic(config)# snmp-server community comm1
```

```
sonic# configure terminal  
sonic(config)# snmp-server community comm1 group group-lab
```

snmp-server contact

Description

Configure the contact information about the organization responsible for the network.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
snmp-server contact <contact-name>  
no snmp-server contact
```

Parameters

Name	Description	Type
contact-name	String	String

Usage Guidelines

Use this command to configure the SNMP server contact information.

Examples

```
sonic# configure terminal  
sonic(config)# snmp-server contact "Broadcom Support"
```

snmp-server enable

Description

Enable the SNMP authentication flag on the switch. The flag is disabled by default.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
snmp-server enable trap  
no snmp-server enable trap
```

Usage Guidelines

Use this command to configure the SNMP authentication flag.

Examples

```
sonic# configure terminal  
sonic(config)# snmp-server enable trap
```

snmp-server engine

Description

Configure the SNMP engine identification on the local device. It is a hexadecimal string used for localizing configuration. The default engine ID is derived from the device MAC address.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
snmp-server engine <engineID>  
no snmp-server engine
```

Parameters

Name	Description	Type
engineID	Octet (hex) string, 5-32 octets	String

Usage Guidelines

Use this command to configure the SNMP engine ID.

Examples

```
sonic# configure terminal  
sonic(config)# snmp-server engine 8100013703525400abcd
```

snmp-server group

Description

Configure one or more SNMPv2c and SNMPv3 access groups and optionally set the views which the group uses for the GET/SET requests and to send traps. Authentication and privacy can be set for SNMPv3 groups only. Groups are used when configuring SNMP communities and users. Configure one or more SNMPv2c access groups and optionally set the views which the group uses for the GET/SET requests and to send traps. Groups are used when configuring SNMP communities.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
snmp-server group <group-name> { { any | v2c | { v3 { noauth | auth | priv } } } { [ read <read-view> ] } { [ write <write-view> ] } { [ notify <notify-view> ] } }  
no snmp-server group <group-name> { any | v2c | { v3 { noauth | auth | priv } } }
```

Parameters

Name	Description	Type
group-name	String(Max: 32 characters)	String
read-view	String(Max: 32 characters)	String
write-view	String(Max: 32 characters)	String
notify-view	String(Max: 32 characters)	String

Usage Guidelines

Use this command to configure an SNMP access group.

Examples

```
sonic# configure terminal  
sonic(config)# snmp-server group group1 v2c
```

```
sonic# configure terminal  
sonic(config)# snmp-server group group1 v2c notify no_view
```

```
sonic# configure terminal  
sonic(config)# snmp-server group group-floor2 v3 priv
```

```
sonic# configure terminal  
sonic(config)# snmp-server group group-floor2 v3 priv read r_view write w_view notify n_view
```

snmp-server host

Description

Configure one or more SNMP IPv4 or IPv6 hosts to which the trap or inform messages are sent to by the SNMP agent. Optionally set the timeout and number of retries for the inform messages sent to a host. Timeout indicates the number of seconds before the informs time out when sending to a host. Retries indicate the number of times the informs are sent after timing out.

The default UDP destination port traps and inform messages are sent to is 162. This may be modified using the port parameter. The source interface and VRF may also be specified using the source-interface

and vrf parameter respectivily. A valid VRF must be specified for vrf.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
snmp-server host <host-addr> { { community { <community-name> { { [ traps v2c ] } | { [ informs  
{ [ timeout <time-out> ] } { [ retries <retry> ] } ] } } { [ source-interface { Ethernet  
| Loopback | Management | PortChannel | Vlan | SubInterface } ] } { [ vrf { mgmt |  
<vrf-name> } ] } { [ port <udpPort> ] } } } | { user { <username> { { [ traps { noauth |  
auth | priv } ] } | { [ informs { noauth | auth | priv } { [ timeout <time-out> ] } { [  
retries <retry> ] } ] } } } { [ source-interface { Ethernet | Loopback | Management |  
PortChannel | Vlan | SubInterface } ] } { [ vrf { mgmt | <vrf-name> } ] } { [ port  
<udpPort> ] } } }  
no snmp-server host <host-addr>
```

Parameters

Name	Description	Type
host-addr	A.B.C.D or A:B:C:D:E:F:G:H	String
community-name	WORD	String
time-out		Integer
retry		Integer
vrf-name	VRF name (prefixed by Vrf, Max: 15 characters)	String
udpPort		Integer
username	String(Max: 32 characters)	String

Usage Guidelines

Use this command to configure an SNMP host.

Examples

```
sonic# configure terminal
sonic(config)# snmp-server host 1.2.3.4 community comm1 traps v2c
```

```
sonic# configure terminal
sonic(config)# snmp-server host 1.2.3.5 user user1 informs noauth timeout 200 retries 10
```

```
sonic# configure terminal
sonic(config)# snmp-server host 2001::1 community comm2 informs timeout 150 retries 5
```

```
sonic# configure terminal
sonic(config)# snmp-server host 3001::1 user u1 traps priv
```

```
sonic# configure terminal
sonic(config)# snmp-server host 2.3.4.5 community public traps v2c port 1492 vrf Vrf1
```

```
sonic# configure terminal
sonic(config)# snmp-server host 2.3.4.5 community public traps v2c port 1492 vrf mgmt
source-interface Loopback1
```

snmp-server location

Description

Configure the physical location of the switch.

Parent Commands (Modes)

configure terminal

Syntax

```
snmp-server location <location-name>
no snmp-server location
```

Parameters

Name	Description	Type
location-name	String	String

Usage Guidelines

Use this command to configure the SNMP server location information.

Examples

```
sonic# configure terminal
sonic(config)# snmp-server location "Lab1, Rack-10"
```

snmp-server user

Description

Configure one or more SNMPv3 users and optionally set the authentication and group association. Authentication passwords can be encrypted. If password encryption is desired, it must be specified prior to setting the authentication type.

Parent Commands (Modes)

configure terminal

Syntax

Parameters

Name	Description	Type
username	String(Max: 32 characters)	String
group-name	String(Max: 32 characters)	String
authpassword	String	String
privpassword	String	String

Usage Guidelines

Use this command to configure an SNMPv3 user.

Examples

```
sonic# configure terminal  
sonic(config)# snmp-server user user1
```

```
sonic# configure terminal  
sonic(config)# snmp-server user user1 group group-lab
```

```
sonic# configure terminal
sonic(config)# snmp-server user user1 group group-lab auth md5 auth-password pwd priv aes-128
               priv-password pwd
```

```
sonic# configure terminal
sonic(config)# snmp-server user user2 group group-floor2 encrypted auth sha auth-password
    abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
    abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
priv des
priv-password abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyz
```

snmp-server view

Description

Configure one or more SNMP views and set the OID tree to include or exclude from the view. SNMP views are used by the groups for the GET/SET requests and to send traps.

Parent Commands (Modes)

configure terminal

Syntax

```
snmp-server view <view-name> { <oid-tree> { included | excluded } }
no snmp-server view <view-name> <oid-tree>
```

Parameters

Name	Description	Type
view-name	String(Max: 32 characters)	String
oid-tree	OID(Max: 255 characters)	String

Usage Guidelines

Use this command to configure a SNMP view.

Examples

```
sonic# configure terminal
sonic(config)# snmp-server view view2 1.2.3.4.5.6.7.8.9.2 excluded
```

snsr-grp

Description

This command is used to assign an existing sensor group to the current subscription.

Parent Commands (Modes)

subscription <sub-id>

Syntax

```
snr-grp <snr-grp_val>
no snr-grp <snr-grp_val>
```

Parameters

Name	Description	Type
snr-grp_val	WORD	String

Usage Guidelines

Use this command to assign an existing sensor group to the current subscription.

Examples

```
sonic(config-telemetry)# subscription s1
sonic(conf-tm-sub-s1)# snr-grp sg1
```

soft-reconfiguration

Description

This command enables soft-reconfiguration for a BGP neighbor

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
soft-reconfiguration inbound
no soft-reconfiguration inbound
```

Usage Guidelines

Use this command to store routes received (RIB-In) from a BGP neighbor. These stored routes could be used to refresh the Loc-RIB in future as needed. If inbound policy changes, these stored routes will be used to generate LocRIB after applying the modified inbound policy.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# soft-reconfiguration
```

soft-reconfiguration

Description

This command enables soft-reconfiguration for BGP neighbors in a peer-group

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
soft-reconfiguration inbound
no soft-reconfiguration inbound
```

Usage Guidelines

Use this command to store routes received (RIB-In) from BGP neighbors in a peer-group. These stored routes could be used to refresh the Loc-RIB in future as needed. If inbound policy changes, these stored routes will be used to generate LocRIB after applying the modified inbound policy.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Int
sonic(config-router-bgp Pg)# address-family ipv4 unicast
sonic(config-router-bgp Pg-af)# soft-reconfiguration
```

soft-reconfiguration

Description

This command enables soft-reconfiguration for a BGP neighbor

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
soft-reconfiguration inbound  
no soft-reconfiguration inbound
```

Usage Guidelines

Use this command to store routes received (RIB-In) from a BGP neighbor. These stored routes could be used to refresh the Loc-RIB in future as needed. If inbound policy changes, these stored routes will be used to generate LocRIB after applying the modified inbound policy.

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 20.20.20.2  
sonic(config-router-bgp-neighbor)# remote-as 300  
sonic(config-router-bgp-neighbor)# address-family l2vpn evpn  
sonic(config-router-bgp-neighbor-af)# soft-reconfiguration
```

soft-reconfiguration

Description

This command enables soft-reconfiguration for BGP neighbors in a peer-group

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
soft-reconfiguration inbound  
no soft-reconfiguration inbound
```

Usage Guidelines

Use this command to store routes received (RIB-In) from BGP neighbors in a peer-group. These stored routes could be used to refresh the Loc-RIB in future as needed. If inbound policy changes, these stored routes will be used to generate LocRIB after applying the modified inbound policy.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Int
sonic(config-router-bgp Pg)# address-family l2vpn evpn
sonic(config-router-bgp Pg-af)# soft-reconfiguration
```

soft-reconfiguration

Description

Per neighbor soft reconfiguration

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
soft-reconfiguration inbound
no soft-reconfiguration inbound
```

soft-reconfiguration

Description

Per neighbor soft reconfiguration

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
soft-reconfiguration inbound
no soft-reconfiguration inbound
```

solo

Description

This command is used to indicate that routes advertised by the peer should not be reflected back to the peer. This command is only meaningful when there is a single peer defined in the peer-group.

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
solo  
no solo
```

Usage Guidelines

```
Use this command to set a neighbor solo
```

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor 30.30.30.3  
sonic(config-router-bgp-neighbor)# solo
```

solo

Description

This command is used to indicate that routes advertised by the peer should not be reflected back to the peer. This command is only meaningful when there is a single peer defined in the peer-group.

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
solo  
no solo
```

Usage Guidelines

```
Use this command to set a peer-group solo
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp-pg)# solo
```

source-address

Description

Configure source IP address for an IP SLA ICMP instance

Parent Commands (Modes)

```
icmp-echo <addr>
```

Syntax

```
source-address <addr>
no source-address
```

Parameters

Name	Description	Type
addr	String	String

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-10)# icmp-echo 30.30.1.2
sonic(conf-ipsla-10-icmp)# source-address 30.30.1.1
```

source-address

Description

Configure source IP address for an IP SLA TCP instance

Parent Commands (Modes)

```
tcp-connect <addr> port <portno>
```

Syntax

```
source-address <addr>
no source-address
```

Parameters

Name	Description	Type
addr	String	String

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-10)# tcp-connect 40.40.1.2
sonic(conf-ipsla-10-tcp)# source-address 40.40.1.1
```

source-interface

Description

Configure source interface for an IP SLA ICMP instance

Parent Commands (Modes)

```
icmp-echo <addr>
```

Syntax

```
source-interface { Ethernet | Subif | PortChannel | Vlan }
no source-interface
```

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-10)# icmp-echo 30.30.1.2
sonic(conf-ipsla-10-icmp)# source-interface Ethernet10
```

source-interface

Description

```
Configure source interface for an IP SLA TCP instance
```

Parent Commands (Modes)

```
tcp-connect <addr> port <portno>
```

Syntax

```
source-interface { Ethernet | Subif | PortChannel | Vlan }
no source-interface
```

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-20)# tcp-connect 40.40.1.2
sonic(conf-ipsla-20-tcp)# source-interface Ethernet50
```

source-ip

Description

```
Configures MLAG session's source ip address
```

Parent Commands (Modes)

```
mclag domain <mclag-domain-id>
```

Syntax

```
source-ip <SIP>
no source-ip
```

Parameters

Name	Description	Type
SIP	A.B.C.D	String

Usage Guidelines

Use this command to configure/change MLAG session's source ip address

Examples

```
sonic-cli(config-mclag-domain-100)#source-ip 10.1.1.1
```

source-ip

Description

Command to set the source IPv4 address

Parent Commands (Modes)

```
interface vxlan <vxlan-if-name>
```

Syntax

```
source-ip <SIP>
no source-ip
```

Parameters

Name	Description	Type
SIP	A.B.C.D	String

Usage Guidelines

```
(conf-if-vxlan-vtep)# source-ip SOURCEIP
SOURCEIP - source IPv4 address
```

Examples

```
sonic(config)# interface vxlan vtep1
sonic(conf-if-vxlan-vtep1)# source-ip 1.1.1.1
```

source-port

Description

```
Configure source port for an IP SLA TCP instance
```

Parent Commands (Modes)

```
tcp-connect <addr> port <portno>
```

Syntax

```
source-port <port>
no source-port
```

Parameters

Name	Description	Type
port		Integer

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-20)# tcp-connect 40.40.1.2
sonic(conf-ipsla-20-tcp)# source-port 200
```

source-vrf

Description

```
Configure ICMP source VRF
```

Parent Commands (Modes)

```
icmp-echo <addr>
```

Syntax

```
source-vrf <vrf-name>
no source-vrf
```

Parameters

Name	Description	Type
vrf-name	String	String

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-10)# icmp-echo 30.30.1.2
sonic(conf-ipsla-10-icmp)# source-vrf VrfRed
```

source-vrf

Description

Configure TCP VRF

Parent Commands (Modes)

```
tcp-connect <addr> port <portno>
```

Syntax

```
source-vrf <vrf-name>
no source-vrf
```

Parameters

Name	Description	Type
vrf-name	String	String

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-20)# tcp-connect 40.40.1.2
sonic(conf-ipsla-20-tcp)# source-vrf VrfBlue
```

speed

Description

```
Configure speed
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
speed <speed>
no speed
```

Parameters

Name	Description	Type
speed	Port speed	Select [10(10MBPS) 100(100MBPS) 1000(1GIGE) 2500(2.5GIGE) 5000(5GIGE) 10000(10GIGE) 20000(20GIGE) 25000(25GIGE) 40000(40GIGE) 50000(50GIGE) 100000(100GIGE) 400000(400GIGE)]

speed

Description

```
Configure the speed of a port.
```

Parent Commands (Modes)

```
interface Management <mgmt-if-id>
```

Syntax

```
speed <speed>
no speed
```

Parameters

Name	Description	Type
speed	Port speed	Select [10(10MBPS) 100(100MBPS) 1000(1GIGE) auto(1GIGE)]

Usage Guidelines

```
speed speed-in-Mbps
```

Examples

```
sonic(conf-if-Ethernet0)# speed 10000  
sonic(conf-if-Ethernet0)#
```

speed

Description

```
Configure speed
```

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
speed <speed>  
no speed
```

Parameters

Name	Description	Type
speed	Port speed	Select [10(10MBPS) 100(100MBPS) 1000(1GIGE) 2500(2.5GIGE) 5000(5GIGE) 10000(10GIGE) 20000(20GIGE) 25000(25GIGE) 40000(40GIGE) 50000(50GIGE) 100000(100GIGE) 400000(400GIGE)]

ssh login_attempts

Description

Configures the number of attempts before rejecting the session.

Parent Commands (Modes)

configure terminal

Syntax

```
ssh login_attempts <login_attempts_val>
```

Parameters

Name	Description	Type
login_attempts_val		Integer

Usage Guidelines

```
sonic-cli(config)# ssh login_attempts <value>
```

Examples

```
sonic-cli(config)# ssh login_attempts 10
sonic-cli(config)#
```

ssh login_timeout

Description

SSH session timeout.

Parent Commands (Modes)

configure terminal

Syntax

```
ssh login_timeout <login_timeout_val>
```

Parameters

Name	Description	Type
login_timeout_val		Integer

Usage Guidelines

```
sonic-clia(config)# ssh login_timeout <value>
```

Examples

```
sonic-clia(config)# ssh login_timeout 100  
sonic-clia(config)#[
```

ssh ports

Description

Sets the SSH port.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ssh ports <port_val>
```

Parameters

Name	Description	Type
port_val		Integer

Usage Guidelines

```
sonic-cli(config)# ssh ports <value>
```

Examples

```
sonic-cli(config)# ssh ports 10000
sonic-cli(config)#
```

ssh tcp-forwarding

Description

Enables or disables tcp forwarding.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
ssh tcp-forwarding <tcp_forwarding_val>
```

Parameters

Name	Description	Type
tcp_forwarding_val	Select [yes(yes) no(no)]	

Usage Guidelines

```
sonic-cli(config)# ssh tcp-forwarding <no>
```

Examples

```
sonic-cli(config)# ssh tcp-forwarding no
sonic-cli(config)#
```

ssh x11-forwarding

Description

Enables or disables x11 forwarding.

Parent Commands (Modes)

configure terminal

Syntax

```
ssh x11-forwarding <x11_forwarding_val>
```

Parameters

Name	Description	Type
x11_forwarding_val	Select [yes(yes) no(no)]	

Usage Guidelines

```
sonic-cli(config)# ssh x11-forwarding <no>
```

Examples

```
sonic-cli(config)# ssh x11-forwarding no
sonic-cli(config)#
```

ssh-server vrf

Description

Enable ssh server on give VRF

Parent Commands (Modes)

configure terminal

Syntax

```
ssh-server vrf <vrf-name>
no ssh-server vrf <vrf-name>
```

Parameters

Name	Description	Type
vrf-name	WORD	String

Examples

```
sonic(config)# ssh-server vrf Vrf2  
sonic(config)#
```

standalone-link-training

Description

```
Enable standalone-link-training
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
standalone-link-training  
no standalone-link-training
```

standalone-link-training

Description

```
Enable standalone-link-training
```

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
standalone-link-training  
no standalone-link-training
```

static

Description

```
Configure static NAT entry
```

Parent Commands (Modes)

```
nat
```

Syntax

```
static { { basic <global-ip> <local-ip> [ <natType> ] { [ twice-nat-id <twice-nat-id-value> ] } } | { <natPortType> <global-ip> <global-port> <local-ip> <local-port> [ <natType> ] { [ twice-nat-id <twice-nat-id-value> ] } } }  
no static { all | { basic <global-ip> } | { <natPortType> <global-ip> <global-port> } }
```

Parameters

Name	Description	Type
global-ip	A.B.C.D	String
local-ip	A.B.C.D	String
natType	NAT type	Select [snat dnat]
twice-nat-id-value		Integer
natPortType	NAT port type	Select [tcp udp]
global-port		Integer
local-port		Integer

storm-control broadcast

Description

```
Command to enable broadcast storm-control on the given interface.
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
storm-control broadcast <kbps>  
no storm-control broadcast
```

Parameters

Name	Description	Type
kbps	kbps	Integer

Usage Guidelines

```
sonic-cli(conf-if-Ethernet0)# storm-control broadcast 10000
```

Examples

```
sonic-cli# configure terminal
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# storm-control broadcast 10000
```

storm-control broadcast

Description

```
Configure broadcast storm-control
```

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
storm-control broadcast <kbps>
no storm-control broadcast
```

Parameters

Name	Description	Type
kbps	kbps	Integer

storm-control unknown-multicast

Description

Command to enable unknown-multicast storm-control on the given interface.

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
storm-control unknown-multicast <kbps>
no storm-control unknown-multicast
```

Parameters

Name	Description	Type
kbps	kbps	Integer

Usage Guidelines

```
sonic-cli(conf-if-Ethernet0)# storm-control unknown-multicast 30000
```

Examples

```
sonic-cli# configure terminal
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# storm-control unknown-multicast 30000
```

storm-control unknown-multicast

Description

Configure unknown-multicast storm-control

Parent Commands (Modes)

interface range iface_range_num

Syntax

```
storm-control unknown-multicast <kbps>
no storm-control unknown-multicast
```

Parameters

Name	Description	Type
kbps	kbps	Integer

storm-control unknown-unicast

Description

Command to enable unknown-unicast storm-control on the given interface.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
storm-control unknown-unicast <kbps>
no storm-control unknown-unicast
```

Parameters

Name	Description	Type
kbps	kbps	Integer

Usage Guidelines

```
sonic-cli(conf-if-Ethernet0)# storm-control unknown-unicast 20000
```

Examples

```
sonic-cli# configure terminal
sonic-cli(config)# interface Ethernet 0
sonic-cli(conf-if-Ethernet0)# storm-control unknown-unicast 20000
```

storm-control unknown-unicast

Description

```
Configure unknown-unicast storm-control
```

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
storm-control unknown-unicast <kbps>
no storm-control unknown-unicast
```

Parameters

Name	Description	Type
kbps	kbps	Integer

strict-capability-match

Description

```
This command instructs BGP to strictly compare remote capabilities and local capabilities. If capabilities are different, send Unsupported Capability error then reset connection.
```

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
strict-capability-match
no strict-capability-match
```

Usage Guidelines

Use this command for a BGP neighbor to enforce exact matching of sent and received capabilities

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# strict-capability-match
```

strict-capability-match

Description

This command instructs BGP to strictly compare remote capabilities and local capabilities. If capabilities are different, send Unsupported Capability error then reset connection.

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
strict-capability-match
no strict-capability-match
```

Usage Guidelines

Use this command for a BGP peer-group to enforce exact matching of sent and received capabilities

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# strict-capability-match
```

stub-router

Description

Make router a stub router.

Parent Commands (Modes)

router ospf6 [vrf <vrf-name>]

Syntax

```
stub-router administrative  
no stub-router administrative
```

Examples

```
sonic-clia(config-router-ospf6)# stub-router administrative
```

Features this CLI belongs to

- OSPFv3 ## subscription ### Description

This command is used to create a new telemetry subscription.

Parent Commands (Modes)

telemetry

Syntax

```
subscription <sub-id>  
no subscription <name>
```

Parameters

Name	Description	Type
sub-id	WORD	String

Usage Guidelines

Use this command to create a new telemetry subscription.

Examples

```
sonic# configure terminal
sonic(config)# telemetry
sonic(config-telemetry)# subscription s1
sonic(conf-tm-sub-s1)#[/pre>
```

switch-id

Description

This command configures a 32-bit identifier that uniquely identifies the switch, and is used in the telemetry reports.

Parent Commands (Modes)

tam

Syntax

```
switch-id <id>
no switch-id
```

Parameters

Name	Description	Type
id	1-4294967295	Integer

Usage Guidelines

This command configures a 32-bit identifier that uniquely identifies the switch, and is used in the telemetry reports. When not configured, the last 16-bits from the system mac address are used.

Examples

```
sonic# configure terminal
sonic(config)# tam
sonic(config-tam)# switch-id 1234
sonic(config-tam)# exit
sonic(config)# exit
sonic# show tam switch
TAM Device information
-----
Switch ID      : 1234
Enterprise ID : 4312
sonic#
```

switch-resource

Description

Switch Resource Configuration

Parent Commands (Modes)

configure terminal

Syntax

switch-resource

switchport access

Description

Set access mode characteristics of the interface

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
switchport access Vlan <vlan-id>
no switchport access Vlan
```

Parameters

Name	Description	Type
vlan-id		Integer

switchport access

Description

Set access mode characteristics of the interface

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
switchport access Vlan <vlan-id>
no switchport access Vlan
```

Parameters

Name	Description	Type
vlan-id		Integer

switchport access

Description

Set access mode characteristics of the interface

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
switchport access Vlan <vlan-id>
no switchport access Vlan
```

Parameters

Name	Description	Type
vlan-id		Integer

switchport access

Description

```
Set access mode characteristics of the interface
```

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ [ fallback ] [ fast_rate ]  
interface range po_range_num
```

Syntax

```
switchport access Vlan <vlan-id>  
no switchport access Vlan
```

Parameters

Name	Description	Type
vlan-id		Integer

switchport trunk

Description

```
Configure trunking parameters on an interface.
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
switchport trunk allowed { Vlan { { add <vlan_id_list> } | { remove <vlan_id_list> } | { except <vlan_id_list> } | all | none | <vlan_id_list> } }  
no switchport trunk allowed { Vlan <vlan_id_list> }
```

Parameters

Name	Description	Type
vlan_id_list	<1..4094>	String

Usage Guidelines

Use this command to add or remove trunking parameters on an interface.

Examples

```
sonic(conf-if-Ethernet4)# switchport trunk allowed Vlan add 10  
sonic(conf-if-Ethernet4)#{
```

```
sonic(conf-if-Ethernet4)# switchport trunk allowed Vlan remove 10  
sonic(conf-if-Ethernet4)#{
```

switchport trunk

Description

Configure trunking parameters on an interface.

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
switchport trunk allowed { Vlan { { add <vlan_id_list> } | { remove <vlan_id_list> } | { except <vlan_id_list> } | all | none | <vlan_id_list> } }  
no switchport trunk allowed { Vlan <vlan_id_list> }
```

Parameters

Name	Description	Type
vlan_id_list	<1..4094>	String

Usage Guidelines

Use this command to add or remove trunking parameters on an interface.

Examples

```
sonic(conf-if-po1)# switchport trunk allowed Vlan add 10  
sonic(conf-if-po1)#[/pre>
```

```
sonic(conf-if-po1)# switchport trunk allowed Vlan remove 10  
sonic(conf-if-po1)#[/pre>
```

switchport trunk

Description

Configure trunking parameters on an interface

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
switchport trunk allowed { Vlan { { add <vlan_id_list> } | { remove <vlan_id_list> } | { except  
    <vlan_id_list> } | all | none | <vlan_id_list> } }  
no switchport trunk allowed { Vlan <vlan_id_list> }
```

Parameters

Name	Description	Type
vlan_id_list	<1..4094>	String

switchport trunk

Description

```
Configure trunking parameters on an interface
```

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
interface range po_range_num
```

Syntax

```
switchport trunk allowed { Vlan { { add <vlan_id_list> } | { remove <vlan_id_list> } | { except <vlan_id_list> } | all | none | <vlan_id_list> } }
no switchport trunk allowed { Vlan <vlan_id_list> }
```

Parameters

Name	Description	Type
vlan_id_list	<1..4094>	String

system vlan

Description

```
Reserve contiguous 128 vlans
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
system vlan <vlan-id> reserve
no system vlan <vlan-id> reserve
```

Parameters

Name	Description	Type
vlan-id		Integer

Examples

```
sonic(config)# system vlan 600 reserve
```

system-mac

Description

```
Configure system mac for the portchannel
```

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
system-mac <sysmac>
no system-mac
```

Parameters

Name	Description	Type
sysmac	nn:nn:nn:nn:nn:nn	String

system-mac

Description

```
Configure system mac for the portchannels
```

Parent Commands (Modes)

```
interface range create po_range_num { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [  
    fallback ] [ fast_rate ]  
interface range po_range_num
```

Syntax

```
system-mac <sysmac>  
no system-mac
```

Parameters

Name	Description	Type
sysmac	nn:nn:nn:nn:nn:nn	String

table-map

Description

This command enables user to apply a route-map on route updates from BGP to Zebra

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
table-map <rmap>  
no table-map [ <rmap> ]
```

Parameters

Name	Description	Type
rmap	WORD	String

Usage Guidelines

This command enables user to apply a route-map on route updates from BGP to Zebra (RIB manager). All the applicable match operations are allowed, such as match on prefix, next-hop, communities, etc. Set operations for this attach-point are limited to metric and next-hop only. Any operation of this feature does not affect BGPs internal RIB.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family ipv4 unicast
sonic(config-router-bgp-af)# table-map rmap_block_private
```

table-map

Description

BGP table to RIB route download filter

Parent Commands (Modes)

address-family ipv6 unicast

Syntax

```
table-map <rtmap>
no table-map [ <rtmap> ]
```

Parameters

Name	Description	Type
rtmap	WORD	String

tacacs-server auth-type

Description

Configure global authentication type for TACACS

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
tacacs-server auth-type <auth-type>
no tacacs-server auth-type
```

Parameters

Name	Description	Type
auth-type	Select [pap(pap) chap(chap) mschap(mschap) login(login)]	

tacacs-server host

Description

```
Configure a server for TACACS
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
tacacs-server host <host> [ port <port-val> ] [ timeout <timeout-val> ] [ key { <key-val> [
    encrypted ] } ] [ type <type-val> ] [ priority <priority-val> ] [ vrf { mgmt } ] [
    source-interface { Ethernet | Loopback | Management | PortChannel | Vlan } ] [ server
    <server-val> ]
no tacacs-server host <host>
```

Parameters

Name	Description	Type
host	WORD	String
port-val	port	Integer
timeout-val	seconds	Integer

Name	Description	Type
key-val	(Valid Chars: ASCII printable except SPACE, #, and COMMA, Max Len: 65) shared secret	String
type-val		Select [pap(pap) chap(chap) mschap(mschap) login(login)]
priority-val	(1..64)	Integer
server-val	WORD	String

tacacs-server key

Description

Configure global shared secret for TACACS

Parent Commands (Modes)

configure terminal

Syntax

```
tacacs-server key <secret-key> [ encrypted ]
no tacacs-server key
```

Parameters

Name	Description	Type
secret-key	(Valid Chars: ASCII printable except SPACE, #, and COMMA, Max Len: 65) shared secret	String

tacacs-server source-interface

Description

Configure source interface to pick the source IP, used for the TACACS+ packets

Parent Commands (Modes)

configure terminal

Syntax

```
tacacs-server source-interface { Ethernet | Loopback | Management | PortChannel | Vlan }
no tacacs-server source-interface
```

tacacs-server timeout

Description

```
Configure global timeout for TACACS
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
tacacs-server timeout <timeout>
no tacacs-server timeout
```

Parameters

Name	Description	Type
timeout	seconds	Integer

tail-stamping

Description

```
Configure tail-stamping feature
```

Parent Commands (Modes)

```
tam
```

Syntax

```
tail-stamping
```

tam

Description

Telemetry and monitoring configuration

Parent Commands (Modes)

configure terminal

Syntax

tam

tcp-connect

Description

Configure operation type as TCP, destination IP and destination port for an IP SLA instance

Parent Commands (Modes)

ip sla <sla-id>

Syntax

```
tcp-connect <addr> port <portno>
no tcp-connect
```

Parameters

Name	Description	Type
addr	String	String
portno		Integer

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-10)# tcp-connect 40.40.1.2 port 100
```

tcp-timeout

Description

Configure TCP NAT entry aging timeout in seconds

Parent Commands (Modes)

nat

Syntax

```
tcp-timeout <tcp-timeout-value>
no tcp-timeout
```

Parameters

Name	Description	Type
tcp-timeout-value		Integer

tcpdump

Description

Prints the contents of packets on a network interface

Syntax

```
tcpdump <interface> [ save <file> ] [ verbose ]
```

Parameters

Name	Description	Type
interface	EthernetNUM	
file	String(Max: 72 characters)	String

tcpdump CPU

Description

CPU interface

Syntax

```
tcpdump CPU [ save <file> ] [ verbose ]
```

Parameters

Name	Description	Type
file	String(Max: 72 characters)	String

tcpdump read

Description

Read packets from tcpdump file

Syntax

```
tcpdump read <read_file> [ verbose ]
```

Parameters

Name	Description	Type
read_file	String(Max: 72 characters)	String

techsupport-export enable

Description

export enable/disable

Parent Commands (Modes)

configure terminal

Syntax

```
techsupport-export enable <ts_mode>
```

Parameters

Name	Description	Type
ts_mode	on or off	Select [on(true) off(false)]

techsupport-export interval

Description

```
the periodic export interval in minutes
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
techsupport-export interval <interval>
```

Parameters

Name	Description	Type
interval	minutes	Integer

techsupport-export remote-server

Description

```
export remote server
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
techsupport-export remote-server <servername> destdir { <ts_destdir> { username <ts_username> }  
{ password <ts_password> } { protocol <ts_protocol> } }
```

Parameters

Name	Description	Type
servername	A.B.C.D or A:B:C:D:E:F:G:H	String
ts_destdir	String	String
ts_username	String	String
ts_password	String	String
ts_protocol	scp or sftp	Select [scp sftp]

telemetry

Description

```
Telemetry global settings and dialout configuration
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
telemetry
```

terminal length

Description

```
Set terminal length
```

Syntax

```
terminal length <length>
```

Parameters

Name	Description	Type
length		Integer

terminal timeout

Description

```
Set terminal timeout
```

Syntax

```
terminal timeout <timeout>
```

Parameters

Name	Description	Type
timeout		Integer

threshold

Description

```
Configure the threshold parameters which can be used to mark a next-hop group as forwardable.
```

Parent Commands (Modes)

```
pbf next-hop-group <fbs-nhgrp-name> [ type { ip | ipv6 } ]
```

Syntax

```
threshold { { type { percentage | count } { [ up <threshold-up> ] } { [ down <threshold-down> ] } } | { up <threshold-up> { [ down <threshold-down> ] } } | { down <threshold-down> } }  
no threshold { [ up ] | [ down ] } }
```

Parameters

Name	Description	Type
threshold-up		Integer
threshold-down		Integer

Usage Guidelines

The threshold up value must be higher than threshold down value. If threshold up is not configured then its assumed as threshold down value + 1.
If threshold down is not configured then its assumed as 0.

Examples

```
sonic(config-pbf-ipv6-nh-group)# threshold type percentage up 70 down 40
sonic(config-pbf-ip-nh-group)# threshold type count up 4 down 2
sonic(config-pbf-ip-nh-group)# threshold down 1
```

threshold

Description

Configure the threshold parameters which can be used to mark a next-hop group as forwardable.

Parent Commands (Modes)

```
pbf next-hop-group <fbn-nhgrp-name> [ type { ip | ipv6 } ]
```

Syntax

```
threshold { { type { percentage | count } { [ up <threshold-up> ] } { [ down <threshold-down> ] } | { up <threshold-up> { [ down <threshold-down> ] } } | { down <threshold-down> } } no threshold { [ up ] | [ down ] } }
```

Parameters

Name	Description	Type
threshold-up		Integer
threshold-down		Integer

Usage Guidelines

The threshold up value must be higher than threshold down value. If threshold up is not configured then its assumed as threshold down value + 1.
If threshold down is not configured then its assumed as 0.

Examples

```
sonic(config-pbf-ipv6-nh-group)# threshold type percentage up 70 down 40
sonic(config-pbf-ip-nh-group)# threshold type count up 4 down 2
sonic(config-pbf-ip-nh-group)# threshold down 1
```

threshold

Description

Configure the threshold parameters which can be used to startup or shutdown downstream interfaces.

Parent Commands (Modes)

```
link state track <grp-name>
```

Syntax

```
threshold { { type percentage { [ up <threshold-up> ] } { [ down <threshold-down> ] } } | { up <threshold-up> { [ down <threshold-down> ] } } | { down <threshold-down> } }
no threshold { [ up ] | [ down ] } ]
```

Parameters

Name	Description	Type
threshold-up		Integer
threshold-down		Integer

Usage Guidelines

The threshold up value must be higher than threshold down value. If threshold up is not configured then its assumed as all upstream interface must be online to bringup downstream interfaces. If threshold down is not configured then its assumed as all upstream interfaces must be offline to shutdown downstream interfaces.

Examples

```
sonic(config-link-track)# threshold type percentage up 70 down 40
sonic(config-link-track)# threshold up 80
sonic(config-link-track)# threshold down 20
```

threshold

Description

Configure threshold count for an IP SLA instance

Parent Commands (Modes)

```
ip sla <sla-id>
```

Syntax

```
threshold <threshold-value>
no threshold
```

Parameters

Name	Description	Type
threshold-value		Integer

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-10)# threshold 5
```

threshold buffer-pool

Description

This command is to configure buffer pool threshold on ingress/egress buffers pools-lossy/lossless.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
threshold buffer-pool <buffer_pool_name> { shared | multicast } <threshold_value>
no threshold buffer-pool <buffer_pool_name> { shared | multicast }
```

Parameters

Name	Description	Type
buffer_pool_name	WORD	String
threshold_value		Integer

Usage Guidelines

Use this command to configure buffer pool threshold on ingress/egress buffer pools-lossy/lossless.

Examples

```
sonic(config)# threshold buffer-pool ingress_lossless_pool multicast 1
sonic(config)# threshold buffer-pool egress_lossy_pool shared 1
sonic(config)# threshold buffer-pool egress_lossless_pool shared 1
```

threshold buffer-pool

Description

This command is used to configure a threshold for buffer-pool of an interface. The threshold value is provided in %. Valid values are 1-100..

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
threshold buffer-pool <buffer_pool_name> { shared | unicast } <threshold_value>
no threshold buffer-pool <buffer_pool_name> { shared | unicast }
```

Parameters

Name	Description	Type
buffer_pool_name	WORD	String
threshold_value		Integer

Usage Guidelines

Use this command to configure queue threshold for buffer-pool of an interface.

Examples

```
sonic(conf-if-Ethernet0)# threshold ingress_port0_lossless_pool shared 47
```

threshold device

Description

This command is used to configure device threshold buffer.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
threshold device <threshold_value>
no threshold device
```

Parameters

Name	Description	Type
threshold_value		Integer

Usage Guidelines

Use this command to configure device threshold buffer.

Examples

```
sonic(config)# threshold device 77
```

threshold priority-group

Description

This command is used to configure a threshold for a specific priority-group shared/headroom buffer of an interface. The threshold value is provided in %. Valid values are 1-100.

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
threshold priority-group <PG-Index> { headroom | shared } <threshold_value>
no threshold priority-group <PG-Index> { headroom | shared }
```

Parameters

Name	Description	Type
PG-Index		Integer
threshold_value		Integer

Usage Guidelines

Use this command to configure a threshold for a specific priority-group shared/headroom buffer of an interface.

Examples

```
sonic(conf-if-Ethernet0)# threshold priority-group 5 headroom 57
sonic(conf-if-Ethernet0)# threshold priority-group 7 shared 78
```

threshold queue

Description

This command is used to configure a threshold for a specific unicast/multicast queue of an interface. The threshold value is provided in %. Valid values are 1-100..

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
threshold queue <queue_index> { unicast | multicast } <threshold_value>
no threshold queue <queue_index> { unicast | multicast }
```

Parameters

Name	Description	Type
queue_index		Integer
threshold_value		Integer

Usage Guidelines

Use this command to configure queue threshold for a specific unicast/multicast queue of an interface.

Examples

```
sonic(conf-if-Ethernet0)# threshold queue 4 unicast 47
sonic(conf-if-Ethernet0)# threshold queue 2 multicast 67
```

threshold queue

Description

This command is used to configure multicact queue buffer threshold of CPU interface. The threshold value is provided in %. Valid values are 1-100..

Parent Commands (Modes)

interface CPU

Syntax

```
threshold queue <cpu_queue_index> multicast <threshold_value>
no threshold queue <cpu_queue_index> multicast
```

Parameters

Name	Description	Type
cpu_queue_index		Integer
threshold_value		Integer

Usage Guidelines

Use this command to configure multicast queue buffer threshold of CPU interface.

Examples

```
sonic(conf-if-CPU)# threshold queue 3 multicast 37
```

timeout

Description

Set timeout to bring up downstream interfaces when atleast one upstream interface comes online.

Parent Commands (Modes)

```
link state track <grp-name>
```

Syntax

```
timeout <grp-tmout>
no timeout
```

Parameters

Name	Description	Type
grp-tmout		Integer

Examples

```
sonic(config-link-track)# timeout 120
```

Alternate command

```
admin@sonic:~$ sudo config linktrack update <name> --timeout <value>
```

timeout

Description

```
Configure NAT entry aging timeout in seconds
```

Parent Commands (Modes)

```
nat
```

Syntax

```
timeout <timeout-value>
no timeout
```

Parameters

Name	Description	Type
timeout-value		Integer

timeout

Description

```
Configure timeout interval for an IP SLA instance
```

Parent Commands (Modes)

```
ip sla <sla-id>
```

Syntax

```
timeout <timeout-value>
no timeout
```

Parameters

Name	Description	Type
timeout-value		Integer

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-10)# timeout 30
```

timeout

Description

Set backend communication timeout. Default is 5 seconds

Syntax

```
timeout
```

timers

Description

Configures OSPFv2 LSA and SPF intervals.

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
timers { { lsa { min-arrival <minarrivaltimer> } } | { throttle { { lsa { all <lsadelaytimer> } } | { spf <spfdelaytime> <spfinitialholdtime> <spfmaxholdtime> } } } }
no timers { { lsa min-arrival } | { throttle { { lsa all } | spf } } }
```

Parameters

Name	Description	Type
minarrivaltimer		Integer
lsadelaytimer		Integer
spfdelaytime		Integer
spfinitialholdtime		Integer
spfmaxholdtime		Integer

Usage Guidelines

Use this command to configure OSPFv2 LSA and SPF intervals.

Examples

```
sonic-clia(config-router-ospf)# timers lsa min-arrival
sonic-clia(config-router-ospf)# timers throttle lsa all
sonic-clia(config-router-ospf)# timers throttle spf 10 20 30
```

Features this CLI belongs to

- OSPFv2 ## timers ### Description

Configures OSPFv3 LSA and SPF intervals.

Parent Commands (Modes)

```
router ospf6 [ vrf <vrf-name> ]
```

Syntax

```
timers { { lsa { min-arrival <minarrivaltimer> } } | { throttle { spf <spfdelaytime>
    <spfinitialholdtime> <spfmaxholdtime> } } }
no timers { { lsa min-arrival } | { throttle spf } }
```

Parameters

Name	Description	Type
minarrivaltimer		Integer
spfdelaytime		Integer
spfinitialholdtime		Integer

Name	Description	Type
spfmaxholdtime		Integer

Usage Guidelines

Use this command to configure OSPFv3 LSA and SPF intervals.

Examples

```
sonic-cl(i(config-router-ospf6)# timers lsa min-arrival
sonic-cl(i(config-router-ospf6)# timers throttle spf 10 20 30
```

Features this CLI belongs to

- OSPFv3 ## timers ### Description

This command configures keepalive and hold timer interval (in seconds) at a global level.

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
timers <keepalive-intvl> <hold-time>
no timers <keepalive-intvl> <hold-time>
```

Parameters

Name	Description	Type
keepalive-intvl		Integer
hold-time		Integer

Usage Guidelines

Use this command to configure keepalive and hold timer interval for an instance of BGP. Note that keepalive timer interval must be smaller than hold timer interval.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# timers 10 30
```

timers

Description

This command enables user to configure keepalive, hold and connect timer intervals for a BGP neighbor.

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
timers { { <keepalive-intvl> <hold-time> } | { connect <connect-time> } }
no timers { { <keepalive-intvl> <hold-time> } | { connect <connect-time> } }
```

Parameters

Name	Description	Type
keepalive-intvl		Integer
hold-time		Integer
connect-time		Integer

Usage Guidelines

Use this command to configure keepalive, hold and connect retry timer intervals for a BGP neighbor.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# timers 3 9
```

timers

Description

This command enables user to configure keepalive, hold and connect timer intervals for a BGP peer-group.

Parent Commands (Modes)

peer-group <template-str>

Syntax

```
timers { { <keepalive-intvl> <hold-time> } | { connect <connect-time> } }
no timers { { <keepalive-intvl> <hold-time> } | { connect <connect-time> } }
```

Parameters

Name	Description	Type
keepalive-intvl		Integer
hold-time		Integer
connect-time		Integer

Usage Guidelines

Use this command to configure keepalive, hold and connect retry timer intervals for a BGP peer-group.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp-pg)# timers 3 9
```

tos

Description

Configure ICMP TOS

Parent Commands (Modes)

```
icmp-echo <addr>
```

Syntax

```
tos <size>
no tos
```

Parameters

Name	Description	Type
size		Integer

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-10)# icmp-echo 30.30.1.2
sonic(conf-ipsla-10-icmp)# tos 4
```

tos

Description

```
Configure TCP TOS
```

Parent Commands (Modes)

```
tcp-connect <addr> port <portno>
```

Syntax

```
tos <size>
no tos
```

Parameters

Name	Description	Type
size		Integer

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-20)# tcp-connect 40.40.1.2
sonic(conf-ipsla-20-tcp)# tos 4
```

traceroute

Description

Print the route packets take to the host

Syntax

```
traceroute [ source <source_dev_ip> ]
```

Parameters

Name	Description	Type
source_dev_ip	WORD	String

traceroute vrf

Description

Select VRF instance

Syntax

```
traceroute vrf <vrf-name> [ source <source_dev_ip> ]
```

Parameters

Name	Description	Type
vrf-name	VRF name (prefixed by Vrf, Max: 15 characters)	String
source_dev_ip	WORD	String

traceroute vrf mgmt

Description

Traceroute using management VRF

Syntax

```
traceroute vrf mgmt [ source <source_dev_ip> ]
```

Parameters

Name	Description	Type
source_dev_ip	WORD	String

traceroute6

Description

Print the route packets take to the IPv6 host

Syntax

```
traceroute6 [ source <source_dev_ip> ]
```

Parameters

Name	Description	Type
source_dev_ip	WORD	String

traceroute6 vrf

Description

Select VRF instance

Syntax

```
traceroute6 vrf <vrf-name> [ source <source_dev_ip> ]
```

Parameters

Name	Description	Type
vrf-name	VRF name (prefixed by Vrf, Max: 15 characters)	String
source_dev_ip	WORD	String

traceroute6 vrf mgmt

Description

Traceroute6 using management VRF

Syntax

```
traceroute6 vrf mgmt [ source <source_dev_ip> ]
```

Parameters

Name	Description	Type
source_dev_ip	WORD	String

track-interface

Description

Configure track interface for changing priority of IPv4 VRRP instance

Parent Commands (Modes)

vrrp ipv4

Syntax

```
track-interface <interface-name> { weight <wt_value> }
no track-interface <interface-name>
```

Parameters

Name	Description	Type
interface-name	String	String
wt_value		Integer

Usage Guidelines

Increases the effective priority by weight value if track interface is up

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4) # vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1) #track-interface Ethernet12 weight 10
sonic(conf-if-Ethernet4-vrrp-ipv4-1) #track-interface Ethernet13 weight 10
```

track-interface

Description

Configure track interface for changing priority of IPv6 VRRP instance

Parent Commands (Modes)

vrrp ipv6

Syntax

```
track-interface <interface-name> { weight <wt_value> }
no track-interface <interface-name>
```

Parameters

Name	Description	Type
interface-name	String	String
wt_value		Integer

Usage Guidelines

Increases the effective priority by weight value if track interface is up

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#track-interface Ethernet24 weight 10
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#track-interface Ethernet25 weight 20
```

track-interface

Description

Configure track interface for changing priority of IPv4 VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
track-interface <interface-name> { weight <wt_value> }
no track-interface <interface-name>
```

Parameters

Name	Description	Type
interface-name	String	String
wt_value		Integer

Usage Guidelines

Increases the effective priority by weight value if track interface is up

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#track-interface Ethernet12 weight 10
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#track-interface Ethernet13 weight 10
```

track-interface

Description

Configure track interface for changing priority of IPv6 VRRP instance

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
track-interface <interface-name> { weight <wt_value> }
no track-interface <interface-name>
```

Parameters

Name	Description	Type
interface-name	String	String
wt_value		Integer

Usage Guidelines

Increases the effective priority by weight value if track interface is up

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4) # vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1) #track-interface Ethernet24 weight 10
sonic(conf-if-Ethernet4-vrrp-ipv6-1) #track-interface Ethernet25 weight 20
```

track-interface

Description

Configure track interface for changing priority of IPv4 VRRP instance

Parent Commands (Modes)

vrrp ipv4

Syntax

```
track-interface <interface-name> { weight <wt_value> }
no track-interface <interface-name>
```

Parameters

Name	Description	Type
interface-name	String	String
wt_value		Integer

Usage Guidelines

Increases the effective priority by weight value if track interface is up

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4) # vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1) #track-interface Ethernet12 weight 10
sonic(conf-if-Ethernet4-vrrp-ipv4-1) #track-interface Ethernet13 weight 10
```

track-interface

Description

Configure track interface for changing priority of IPv6 VRRP instance

Parent Commands (Modes)

vrrp ipv6

Syntax

```
track-interface <interface-name> { weight <wt_value> }  
no track-interface <interface-name>
```

Parameters

Name	Description	Type
interface-name	String	String
wt_value		Integer

Usage Guidelines

Increases the effective priority by weight value if track interface is up

Examples

```
sonic(config)# interface Ethernet4  
sonic(conf-if-Ethernet4)#  
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6  
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#track-interface Ethernet24 weight 10  
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#track-interface Ethernet25 weight 20
```

track-interface

Description

Configure track interface for changing priority of IPv4 VRRP instance

Parent Commands (Modes)

vrrp ipv4

Syntax

```
track-interface <interface-name> { weight <wt_value> }
no track-interface <interface-name>
```

Parameters

Name	Description	Type
interface-name	String	String
wt_value		Integer

Usage Guidelines

Increases the effective priority by weight value if track interface is up

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4) # vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1) #track-interface Ethernet12 weight 10
sonic(conf-if-Ethernet4-vrrp-ipv4-1) #track-interface Ethernet13 weight 10
```

track-interface

Description

Configure track interface for changing priority of IPv6 VRRP instance

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
track-interface <interface-name> { weight <wt_value> }
no track-interface <interface-name>
```

Parameters

Name	Description	Type
interface-name	String	String
wt_value		Integer

Usage Guidelines

Increases the effective priority by weight value if track interface is up

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#track-interface Ethernet24 weight 10
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#track-interface Ethernet25 weight 20
```

traffic-class

Description

This command to add Traffic class to DSCP entry in map.

Parent Commands (Modes)

```
qos map tc-dscp <name>
```

Syntax

```
traffic-class <tc_list> { dscp <dscp-val> }
no traffic-class <tc_list>
```

Parameters

Name	Description	Type
tc_list		String
dscp-val		Integer

Usage Guidelines

Use this command to add entry to Traffic class to DSCP map.

Examples

```
sonic# configure terminal
sonic(config)# qos map tc-dscp tc_dscp
sonic(conf-tc-dscp-map-tc_dscp)# traffic-class 0 dscp 0
sonic(conf-tc-dscp-map-tc_dscp)# traffic-class 1 dscp 10
sonic(conf-tc-dscp-map-tc_dscp)# traffic-class 3 dscp 29
```

traffic-class

Description

This command to add Traffic class to Priority-Group entry in map.

Parent Commands (Modes)

```
qos map tc-pg <name>
```

Syntax

```
traffic-class <tc_list> { priority-group <pg> }
no traffic-class <tc_list>
```

Parameters

Name	Description	Type
tc_list		String
pg		Integer

Usage Guidelines

Use this command to add entry to map Traffic class to Priority-Group.

Examples

```
sonic# configure terminal
sonic(config)# traffic-class 0 priority-group 0
sonic(config)# traffic-class 2 priority-group 0
sonic(config)# traffic-class 1 priority-group 2
```

traffic-class

Description

This command to add Traffic class to DOT1P entry in map.

Parent Commands (Modes)

```
qos map tc-dot1p <name>
```

Syntax

```
traffic-class <tc_list> { dot1p <dot1p-val> }
no traffic-class <tc_list>
```

Parameters

Name	Description	Type
tc_list		String
dot1p-val		Integer

Usage Guidelines

Use this command to add entry to Traffic class to DOT1P map.

Examples

```
sonic# configure terminal
sonic(config)# qos map tc-dot1p tc_dot1p
sonic(conf-tc-dot1p-map-tc_dot1p)# traffic-class 0 dot1p 0
sonic(conf-tc-dot1p-map-tc_dot1p)# traffic-class 1 dot1p 1
sonic(conf-tc-dot1p-map-tc_dot1p)# traffic-class 3 dot1p 2
```

traffic-class

Description

This command to add Traffic class to Queue entry in map.

Parent Commands (Modes)

qos map tc-queue <name>

Syntax

```
traffic-class <tc_list> { queue <qid> }
no traffic-class <tc_list>
```

Parameters

Name	Description	Type
tc_list		String
qid		Integer

Usage Guidelines

Use this command to add entry to map Traffic class to Queue.

Examples

```
sonic# configure terminal
sonic(config)# traffic-class 0 queue 0
sonic(config)# traffic-class 2 queue 0
sonic(config)# traffic-class 1 queue 2
```

transmit-interval

Description

Configure desired packet transmit interval for Bidirectional Forwarding detection(BFD) peer.

Parent Commands (Modes)

```
peer <peer_ipv4>
peer <peer_ipv6>
peer [ interface ] <interfacename>
peer [ local-address ] <local_ipv4>
peer [ local-address ] <local_ipv6>
peer [ multihop ]
peer [ vrf ] <vrfname>
```

Syntax

```
transmit-interval <transmit_interval>
```

Parameters

Name	Description	Type
transmit_interval		Integer

Usage Guidelines

Default value is 300 milliseconds.

Examples

```
device()#configure terminal
device(config)#bfd
device(conf-bfd)# peer 192.168.0.5 interface Ethernet0
device(conf-bfd-peer)# transmit-interval 200
```

transmit-interval

Description

Configure desired packet transmit interval for Bidirectional Forwarding detection(BFD) peer.

Parent Commands (Modes)

```
profile <profilename>
```

Syntax

```
transmit-interval <transmit_interval>
```

Parameters

Name	Description	Type
transmit_interval		Integer

Usage Guidelines

Default value is 300 milliseconds.

Examples

```
device()#configure terminal
device(config)#bfd
device(conf-bfd)# profile fast
device(conf-bfd-profile)# transmit-interval 200
```

ttl

Description

Configure ICMP TTL

Parent Commands (Modes)

```
icmp-echo <addr>
```

Syntax

```
ttl <size>
no ttl
```

Parameters

Name	Description	Type
size		Integer

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-10)# icmp-echo 30.30.1.2
sonic(conf-ipsla-10-icmp)# ttl 16
```

ttl

Description

Configure TCP TTL

Parent Commands (Modes)

```
tcp-connect <addr> port <portno>
```

Syntax

```
ttl <size>
no ttl
```

Parameters

Name	Description	Type
size		Integer

Examples

```
sonic(config)# ip sla 10
sonic(conf-ipsla-20)# tcp-connect 40.40.1.2
sonic(conf-ipsla-20-tcp)# ttl 16
```

ttl-security hops

Description

This command enforces Generalized TTL Security Mechanism (GTSM), as specified in RFC 5082. With this command, only neighbors that are the specified number of hops away will be allowed to become neighbors. This command is mutually exclusive with ebgp-multihop.

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
ttl-security hops <nhops>
no ttl-security hops
```

Parameters

Name	Description	Type
nhops		Integer

Usage Guidelines

Use this command for a BGP neighbor to enable Generalized TTL Security Mechanism (GTSM). This is for security purposes.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# ttl-security hops 6
```

ttl-security hops

Description

This command enforces Generalized TTL Security Mechanism (GTSM), as specified in RFC 5082. With this command, only neighbors that are the specified number of hops away will be allowed to become neighbors. This command is mutually exclusive with ebgp-multihop.

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
ttl-security hops <nhops>
no ttl-security hops
```

Parameters

Name	Description	Type
nhops		Integer

Usage Guidelines

Use this command for a BGP peer-group to enable Generalized TTL Security Mechanism (GTSM). This is for security purposes.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# ttl-security hops 8
```

type

Description

```
Add dropcounter type
```

Parent Commands (Modes)

```
dropcounters <counter-name>
```

Syntax

```
type <type-str>
no type
```

Parameters

Name	Description	Type
type-str	Debug counter type. PORT_INGRESS_DROPS	Select [PORT_INGRESS_DROPS]

type

Description

```
Scheduler type (strict/dwrr/wrr)
```

Parent Commands (Modes)

```
qos scheduler-policy <name>
```

Syntax

```
type <type>
no type
```

Parameters

Name	Description	Type
type	Scheduler type Options	Select [dwrr(DWRR) wrr(WRR) strict(STRICK)]

type

Description

```
Scheduler type (strict/dwrr/wrr)
```

Syntax

```
type <type>
no type
```

Parameters

Name	Description	Type
type	Scheduler type Options	Select [dwrr(DWRR) wrr(WRR) strict(STRICT)]

udp-timeout

Description

```
Configure UDP NAT entry aging timeout in seconds
```

Parent Commands (Modes)

```
nat
```

Syntax

```
udp-timeout <udp-timeout-value>
no udp-timeout
```

Parameters

Name	Description	Type
udp-timeout-value		Integer

unreliable-los

Description

```
Configure Unreliable-LOS mode
```

Parent Commands (Modes)

```
interface <phy-if-name>
```

Syntax

```
unreliable-los <unlos>
no unreliable-los
```

Parameters

Name	Description	Type
unlos	Unreliable-LOS mode	Select [auto on off]

unreliable-los

Description

```
Configure Unreliable-LOS mode
```

Parent Commands (Modes)

```
interface range iface_range_num
```

Syntax

```
unreliable-los <unlos>
no unreliable-los
```

Parameters

Name	Description	Type
unlos	Unreliable-LOS mode	Select [auto on off]

unsuppress-map

Description

```
This command configures a Route-map to selectively unsuppress
suppressed routes
```

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
unsuppress-map <map>
no unsuppress-map <map>
```

Parameters

Name	Description	Type
map	WORD	String

Usage Guidelines

Use this command to define a route policy via route-map to unsuppress suppressed routes.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# unsuppress-map rm_unsup_ext_rt
```

unsuppress-map

Description

This command configures a Route-map to selectively unsuppress suppressed routes

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
unsuppress-map <map>
no unsuppress-map <map>
```

Parameters

Name	Description	Type
map	WORD	String

Usage Guidelines

Use this command to define a route policy via route-map to unsuppress suppressed routes.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Int
sonic(config-router-bgp-pg)# address-family ipv4 unicast
sonic(config-router-bgp-pg-af)# unsuppress-map rm_unsup_ext_rt
```

unsuppress-map

Description

Route-map to selectively unsuppress suppressed routes

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
unsuppress-map <map>
no unsuppress-map <map>
```

Parameters

Name	Description	Type
map	WORD	String

unsuppress-map

Description

```
Route-map to selectively unsuppress suppressed routes
```

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
unsuppress-map <map>
no unsuppress-map <map>
```

Parameters

Name	Description	Type
map	WORD	String

update-delay

Description

```
This command allows user to set update delay. This parameter control how long to wait before running best-path selection after Graceful Restart.
```

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
update-delay <time> [ <maxmedval> ]
no update-delay
```

Parameters

Name	Description	Type
time		Integer
maxmedval		Integer

Usage Guidelines

This feature is used to enable read-only mode on BGP process restart or when BGP process is cleared using clear ip bgp *. When applicable, read-only mode would begin as soon as the first peer reaches Established status and a timer for max-delay seconds is started.

During this mode BGP doesn't run any best-path or generate any updates to its peers. This mode continues until:

All the configured peers, except the shutdown peers, have sent explicit EOR (End-Of-RIB) or an implicit-EOR. The first keep-alive after BGP has reached Established is considered an implicit-EOR. If the establish-wait optional value is given, then BGP will wait for peers to reach established from the beginning of the update-delay till the establish-wait period is over, i.e. the minimum set of established peers for which EOR is expected would be peers established during the establish-wait window, not necessarily all the configured neighbors. max-delay period is over.

On hitting any of the above two conditions, BGP resumes the decision process and generates updates to its peers.

Default max-delay is 0, i.e. the feature is off by default.

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# update-delay 120
```

update-source

Description

This command specifies the IPv4 or IPv6 source address to use for the BGP session to this neighbor. Source address may be specified as either an IPv4/IPv6 address directly or as an interface name. The interface name could be router port or Port Channel or Vlan interface or routed Vlan sub-interface with IPv4/IPv6 address configured on it.

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
update-source { <ip> | { interface { Ethernet | PortChannel | Vlan | Loopback } } }
no update-source { [ <ip> ] | { interface { Ethernet | PortChannel | Vlan | Loopback } } }
```

Parameters

Name	Description	Type
ip	A.B.C.D/A::B	String

Usage Guidelines

Use this command to configure source interface for a BGP neighbor sessions

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 30.30.30.3
sonic(config-router-bgp-neighbor)# update-source 12.56.36.74
```

update-source

Description

This command specifies the IPv4 or IPv6 source address to use for the BGP session to neighbors in a peer-group. Source address may be specified as either an IPv4/IPv6 address directly or as an interface name. The interface name could be router port or Port Channel or Vlan interface or routed Vlan sub-interface with IPv4/IPv6 address configured on it.

Parent Commands (Modes)

```
peer-group <template-str>
```

Syntax

```
update-source { <ip> | { interface { Ethernet | PortChannel | Vlan | Loopback } } }
no update-source { [ <ip> ] | { interface { Ethernet | PortChannel | Vlan | Loopback } } }
```

Parameters

Name	Description	Type
ip	A.B.C.D/A::B	String

Usage Guidelines

Use this command to configure source interface for a BGP peer-group

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Ext
sonic(config-router-bgp Pg)# update-source Ethernet16
```

use-es-l3nhg

Description

This command enables use of L3 nexthop group for host routes with ES destination

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
use-es-l3nhg
no use-es-l3nhg
```

Usage Guidelines

```
[no] use-es-l3nhg
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# use-es-l3nhg
```

use-v2-checksum

Description

Configure checksum compatibility for VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
use-v2-checksum  
no use-v2-checksum
```

Examples

```
sonic(config)# interface Ethernet4  
sonic(conf-if-Ethernet4)#  
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4  
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#use-v2-checksum
```

use-v2-checksum

Description

Configure checksum compatibility for VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
use-v2-checksum  
no use-v2-checksum
```

Examples

```
sonic(config)# interface Ethernet4  
sonic(conf-if-Ethernet4)#  
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4  
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#use-v2-checksum
```

use-v2-checksum

Description

Configure checksum compatibility for VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
use-v2-checksum  
no use-v2-checksum
```

Examples

```
sonic(config)# interface Ethernet4  
sonic(conf-if-Ethernet4)#  
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4  
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#use-v2-checksum
```

use-v2-checksum

Description

Configure checksum compatibility for VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
use-v2-checksum  
no use-v2-checksum
```

Examples

```
sonic(config)# interface Ethernet4  
sonic(conf-if-Ethernet4)#  
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4  
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#use-v2-checksum
```

username

Description

Add new user or modify existing one

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
username <user-name> [ role <r1> ] [ password <passwd> ] [ shell <sh1> ]  
no username <user-name>
```

Parameters

Name	Description	Type
user-name	String	String
r1		Select [admin guest]
passwd	WORD(Except , and \)	String
sh1		Select [bash rbash sonic-cli]

v6only

Description

```
Enable BGP with v6 link-local only.
```

Parent Commands (Modes)

```
neighbor { <ip> | { interface { Ethernet | PortChannel | Vlan } } }
```

Syntax

```
v6only  
no v6only
```

Usage Guidelines

```
Enable BGP session with only v6 link-local on an interface neighbor.
```

Examples

```
sonic# configure terminal  
sonic(config)# router bgp 100  
sonic(config-router-bgp)# neighbor interface Ethernet 0  
sonic(config-router-bgp-neighbor)# v6only
```

version

Description

```
Configure version for IPv4 VRRP instance
```

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
version <ver>
no version
```

Parameters

Name	Description	Type
ver		Integer

Usage Guidelines

```
Default value is version 2
```

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#version 3
```

version

Description

```
Configure version for IPv4 VRRP instance
```

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
version <ver>
no version
```

Parameters

Name	Description	Type
ver		Integer

Usage Guidelines

Default value is version 2

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#version 3
```

version

Description

Configure version for IPv4 VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
version <ver>
no version
```

Parameters

Name	Description	Type
ver		Integer

Usage Guidelines

Default value is version 2

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#version 3
```

version

Description

Configure version for IPv4 VRRP instance

Parent Commands (Modes)

vrrp ipv4

Syntax

```
version <ver>
no version
```

Parameters

Name	Description	Type
ver		Integer

Usage Guidelines

Default value is version 2

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#version 3
```

vip

Description

Configure virtual IP address for IPv4 VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
vip <vip_addr>
no vip <vip_addr>
```

Parameters

Name	Description	Type
vip_addr	A.B.C.D	String

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#vip 40.0.0.5
```

vip

Description

Configure virtual IP address for IPv6 VRRP instance

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
vip <vip_addr>
no vip <vip_addr>
```

Parameters

Name	Description	Type
vip_addr	A::B	String

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#vip 40::5
```

vip

Description

```
Configure virtual IP address for IPv4 VRRP instance
```

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
vip <vip_addr>
no vip <vip_addr>
```

Parameters

Name	Description	Type
vip_addr	A.B.C.D	String

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#vip 40.0.0.5
```

vip

Description

Configure virtual IP address for IPv6 VRRP instance

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
vip <vip_addr>
no vip <vip_addr>
```

Parameters

Name	Description	Type
vip_addr	A::B	String

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#vip 40::5
```

vip

Description

Configure virtual IP address for IPv4 VRRP instance

Parent Commands (Modes)

vrrp ipv4

Syntax

```
vip <vip_addr>
no vip <vip_addr>
```

Parameters

Name	Description	Type
vip_addr	A.B.C.D	String

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#vip 40.0.0.5
```

vip

Description

Configure virtual IP address for IPv6 VRRP instance

Parent Commands (Modes)

vrrp ipv6

Syntax

```
vip <vip_addr>
no vip <vip_addr>
```

Parameters

Name	Description	Type
vip_addr	A::B	String

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#vip 40::5
```

vip

Description

Configure virtual IP address for IPv4 VRRP instance

Parent Commands (Modes)

```
vrrp ipv4
```

Syntax

```
vip <vip_addr>
no vip <vip_addr>
```

Parameters

Name	Description	Type
vip_addr	A.B.C.D	String

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#vip 40.0.0.5
```

vip

Description

```
Configure virtual IP address for IPv6 VRRP instance
```

Parent Commands (Modes)

```
vrrp ipv6
```

Syntax

```
vip <vip_addr>
no vip <vip_addr>
```

Parameters

Name	Description	Type
vip_addr	A::B	String

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
sonic(conf-if-Ethernet4-vrrp-ipv6-1)#vip 40::5
```

vni

Description

```
This command enables user to configure per-VNI EVPN parameters
```

Parent Commands (Modes)

```
address-family l2vpn evpn
```

Syntax

```
vni <vninum>
no vni <vninum>
```

Parameters

Name	Description	Type
vnum	VNI	Integer

Usage Guidelines

```
[no] vni {vni-number}
```

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# address-family l2vpn evpn
sonic(config-router-bgp-af)# vni 100
sonic(config-router-bgp-af-vni)#

```

vni-downstream

Description

Command to set the Downstream VNI configuration

Parent Commands (Modes)

```
interface vxlan <vxlan-if-name>
```

Syntax

```
vni-downstream <remoteip>
no vni-downstream <remoteip>
```

Parameters

Name	Description	Type
remoteip	A.B.C.D or external	String

Usage Guidelines

```
(conf-if-vxlan-vtep)# vni-downstream REMOTEIP  
REMOTEIP - "external" for all external tunnels and remote IPv4 address for internal tunnels
```

Examples

```
sonic(config)# interface vxlan vtep1  
sonic(conf-if-vxlan-vtep1)# vni-downstream 1.1.1.2  
sonic(conf-if-vxlan-vtep1)# vni-downstream external
```

voice

Description

Configure voice application type and its attributes.

Parent Commands (Modes)

```
network-policy profile <np_num>
```

Syntax

```
voice vlan { { <vlan-id> [ untagged ] { [ cos <cos-val> ] } { [ dscp <dscp-val> ] } } | { dot1p  
    { [ cos <cos-val> ] } { [ dscp <dscp-val> ] } } }  
no voice vlan
```

Parameters

Name	Description	Type
vlan-id		Integer
cos-val		Integer
dscp-val		Integer

Usage Guidelines

This command allows to configure voice application type and its attributes.

Examples

```
sonic-cli(config)# voice vlan 100 cos 4 dscp 20
```

Features this CLI belongs to

- LLDP-MED ## voice-signaling ### Description

Configure voice-signaling application type and its attributes.

Parent Commands (Modes)

network-policy profile <np_num>

Syntax

```
voice-signaling vlan { { <vlan-id> [ untagged ] { [ cos <cos-val> ] } { [ dscp <dscp-val> ] } }  
    | { dot1p { [ cos <cos-val> ] } { [ dscp <dscp-val> ] } } }  
no voice-signaling vlan
```

Parameters

Name	Description	Type
vlan-id		Integer
cos-val		Integer
dscp-val		Integer

Usage Guidelines

This command allows to configure voice-signaling application type and its attributes.

Examples

```
sonic-cli(config)# voice-signaling vlan dot1p cos 3 dscp 10
```

Features this CLI belongs to

- LLDP-MED ## vrrp ### Description

Configure Virtual Router Redundancy Protocol(VRRP) on Ethernet interface

Parent Commands (Modes)

interface <phy-if-name>

Syntax

```
vrrp <vrrp-id> address-family { ipv4 | ipv6 }
no vrrp <vrrp-id> address-family { ipv4 | ipv6 }
```

Parameters

Name	Description	Type
vrrp-id		Integer

Examples

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv4
sonic(conf-if-Ethernet4-vrrp-ipv4-1)#+
```

```
sonic(config)# interface Ethernet4
sonic(conf-if-Ethernet4)#
sonic(conf-if-Ethernet4)# vrrp 1 address-family ipv6
```

vrrp

Description

Configure Virtual Router Redundancy Protocol(VRRP) on PortChannel interface

Parent Commands (Modes)

```
interface PortChannel <lag-id> { [ mode <PoMode> ] } { [ min-links <min-links-value> ] } [ fallback ] [ fast_rate ]
```

Syntax

```
vrrp <vrrp-id> address-family { ipv4 | ipv6 }
no vrrp <vrrp-id> address-family { ipv4 | ipv6 }
```

Parameters

Name	Description	Type
vrrp-id		Integer

Examples

```
sonic(config)# interface PortChannel 10
sonic(conf-if-po10)#
sonic(conf-if-po10)# vrrp 1 address-family ipv4
sonic(conf-if-po10-vrrp-ipv4-1)#

```

```
sonic(config)# interface PortChannel 10
sonic(conf-if-po10)#
sonic(conf-if-po10)# vrrp 1 address-family ipv6

```

vrrp

Description

Configure Virtual Router Redundancy Protocol(VRRP) on Vlan interface

Parent Commands (Modes)

```
interface <vlan-if-name>
```

Syntax

```
vrrp <vrrp-id> address-family { ipv4 | ipv6 }
no vrrp <vrrp-id> address-family { ipv4 | ipv6 }
```

Parameters

Name	Description	Type
vrrp-id		Integer

Examples

```
sonic(config)# interface Vlan 10
sonic(conf-if-Vlan10)#
sonic(conf-if-Vlan10) # vrrp 1 address-family ipv4
sonic(conf-if-Vlan10-vrrp-ipv4-1)#
```

```
sonic(config)# interface Vlan 10
sonic(conf-if-Vlan10)#
sonic(conf-if-Vlan10) # vrrp 1 address-family ipv6
```

vrrp

Description

Configure Virtual Router Redundancy Protocol(VRRP) on Ethernet interface

Parent Commands (Modes)

```
interface <phy-sub-if-name>
interface PortChannel <lag-id-subid>
```

Syntax

```
vrrp <vrrp-id> address-family { ipv4 | ipv6 }
no vrrp <vrrp-id> address-family { ipv4 | ipv6 }
```

Parameters

Name	Description	Type
vrrp-id		Integer

Examples

```
sonic(config)# interface Eth4.1
sonic(conf-if-Eth4.1)#
sonic(conf-if-Eth4.1) # vrrp 1 address-family ipv4
sonic(conf-if-Eth4.1-vrrp-ipv4-1)#
```

```
sonic(config)# interface Eth4.1
sonic(conf-if-Eth4.1)#
sonic(conf-if-Eth4.1) # vrrp 1 address-family ipv6
```

warm-reboot

Description

```
warm-reboot
```

Syntax

```
warm-reboot [ force <_force> ] [ verbose <_verbose> ] [ ignore <_ignore> ] [ sbin-reboot
<_sbin_reboot> ] [ noprompt <_noprompt> ]
```

Parameters

Name	Description	Type
_force	[yes/no]	Select [yes no]
_verbose	[yes/no]	Select [yes no]
_ignore	[yes/no]	Select [yes no]
_sbin_reboot	[yes/no]	Select [yes no]
_noprompt	[yes/no]	Select [yes no]

warm-restart bgp

Description

```
Timer value for warm restart of BGP service
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
warm-restart bgp timer <value>
no warm-restart bgp timer
```

Parameters

Name	Description	Type
value	timeout in seconds	Integer

Examples

```
sonic(config)# warm-restart bgp timer 60
sonic(config)#
```

warm-restart bgp enable

Description

```
Enable warm restart feature
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
warm-restart bgp enable
no warm-restart bgp enable
```

Examples

```
sonic(config)# warm-restart bgp enable
sonic(config)#
```

warm-restart bgp eoui

Description

```
Enable BGP EOIU flag
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
warm-restart bgp eoui
no warm-restart bgp eoui
```

Examples

```
sonic(config)# warm-restart bgp eoui  
sonic(config)#
```

warm-restart swss

Description

```
Timer value for warm restart of SWSS service
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
warm-restart swss timer <value>  
no warm-restart swss timer
```

Parameters

Name	Description	Type
value	timeout in seconds	Integer

Examples

```
sonic(config)# warm-restart swss timer 3600  
sonic(config)#
```

warm-restart swss enable

Description

```
Enable warm restart feature
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
warm-restart swss enable  
no warm-restart swss enable
```

Examples

```
sonic(config)# warm-restart swss enable  
sonic(config)#
```

warm-restart system

Description

```
Enable warm restart feature
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
warm-restart system [ enable ]  
no warm-restart system [ enable ]
```

Examples

```
sonic(config)# warm-restart system  
sonic(config)#
```

warm-restart teamd

Description

```
Timer value for warm restart of teamd service
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
warm-restart teamd timer <value>  
no warm-restart teamd timer
```

Parameters

Name	Description	Type
value	timeout in seconds	Integer

Examples

```
sonic(config)# warm-restart teamd timer 600  
sonic(config)#
```

warm-restart teamd enable

Description

```
Enable warm restart feature
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
warm-restart teamd enable  
no warm-restart teamd enable
```

Examples

```
sonic(config)# warm-restart teamd enable  
sonic(config)#
```

watermark interval

Description

```
This command is used to configure snapshot watermark interval.
```

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
watermark interval <interval_value>
no watermark interval
```

Parameters

Name	Description	Type
interval_value		Integer

Usage Guidelines

Use this command to configure snapshot watermark interval.

Examples

```
sonic(config)# watermark interval 77
```

watermark telemetry

Description

This command is used to configure watermark telemetry interval.

Parent Commands (Modes)

```
configure terminal
```

Syntax

```
watermark telemetry interval <interval_value>
no watermark telemetry interval
```

Parameters

Name	Description	Type
interval_value		Integer

Usage Guidelines

Use this command to configure watermark telemetry interval.

Examples

```
sonic(config)# watermark telemetry interval 88
```

weight

Description

This command configures a default weight for all routes received from this BGP neighbor

Parent Commands (Modes)

```
address-family ipv4 unicast
```

Syntax

```
weight <val>
no weight [ <val> ]
```

Parameters

Name	Description	Type
val		Integer

Usage Guidelines

Use this command to assign a default weight to BGP routes received from this neighbor. Weight parameter is used in BGP route selection process. So, configuring weight may influence the outcome of the route selection process

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# neighbor 20.20.20.2
sonic(config-router-bgp-neighbor)# remote-as 300
sonic(config-router-bgp-neighbor)# address-family ipv4 unicast
sonic(config-router-bgp-neighbor-af)# weight 45
```

weight

Description

This command configures a default weight for all routes received from neighbors in this peer-group

Parent Commands (Modes)

address-family ipv4 unicast

Syntax

```
weight <val>
no weight [ <val> ]
```

Parameters

Name	Description	Type
val		Integer

Usage Guidelines

Use this command to assign a default weight to BGP routes received from neighbors in this peer-group. Weight parameter is used in BGP route selection process. So, configuring weight may influence the outcome of the route selection process

Examples

```
sonic# configure terminal
sonic(config)# router bgp 100
sonic(config-router-bgp)# peer-group PG_Int
sonic(config-router-bgp-pg)# address-family ipv4 unicast
sonic(config-router-bgp-pg-af)# weight 25
```

weight

Description

```
Scheduler weight
```

Parent Commands (Modes)

```
qos scheduler-policy <name>
```

Syntax

```
weight <weight>
no weight
```

Parameters

Name	Description	Type
weight		Integer

weight

Description

```
Scheduler weight
```

Syntax

```
weight <weight>
no weight
```

Parameters

Name	Description	Type
weight		Integer

weight

Description

```
Set default weight for routes from this neighbor
```

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
weight <val>
no weight [ <val> ]
```

Parameters

Name	Description	Type
val		Integer

weight

Description

```
Set default weight for routes from this neighbor
```

Parent Commands (Modes)

```
address-family ipv6 unicast
```

Syntax

```
weight <val>
no weight [ <val> ]
```

Parameters

Name	Description	Type
val		Integer

write

Description

Save config

Syntax

```
write [ memory ]
```

write erase install

Description

Restore SONiC switch content to default values.

The "write erase install" command removes all changes made by the user. All user installed packages and file changes are removed. It also deletes the startup configuration JSON file and the files in /etc/sonic directory. The SONiC switch is reverted to the same state as a newly installed image. After the SONiC switch is rebooted, if the Zero Touch Provisioning (ZTP) feature is enabled, the SONiC switch will start performing ZTP to discover and download the switch configuration.

Syntax

```
write erase install
```

Usage Guidelines

Use this command to restore SONiC switch content to default values.

Examples

```
sonic# configure terminal
sonic(config)# write erase install
All SONiC switch content will be restored to default values, continue? [y/N] :
```

write-multiplier

Description

Configures OSPFv2 write multiplier.

Parent Commands (Modes)

```
router ospf [ vrf <vrf-name> ]
```

Syntax

```
write-multiplier <maxinterfacewrite>
no write-multiplier
```

Parameters

Name	Description	Type
maxinterfacewrite		Integer

Usage Guidelines

Use this command to configure OSPFv2 write multiplier.

Examples

```
sonic-clia(config-router-ospf)# write-multiplier 20
```

Features this CLI belongs to

- OSPFv2 ## write-multiplier ### Description

Configures OSPFv3 write multiplier.

Parent Commands (Modes)

```
router ospf6 [ vrf <vrf-name> ]
```

Syntax

```
write-multiplier <maxinterfacewrite>
no write-multiplier
```

Parameters

Name	Description	Type
maxinterfacewrite		Integer

Usage Guidelines

Use this command to configure OSPFv3 write multiplier.

Examples

```
sonic-cli(config-router-ospf6)# write-multiplier 20
```

Features this CLI belongs to

- OSPFv3 ## write-quanta ##### Description

This command configures the maximum number of BGP packets to write to peer socket in one cycle of I/O

Parent Commands (Modes)

```
router bgp <as-num-dot> { [ vrf <vrf-name> ] }
```

Syntax

```
write-quanta <wrval>
no write-quanta
```

Parameters

Name	Description	Type
wrval		Integer

Usage Guidelines

BGP message Tx I/O is vectored. This means that multiple packets are written to the peer socket at the same time each I/O cycle, in order to minimize system call overhead. This value controls how many are written at a time. Under certain load conditions, reducing this value could make peer traffic less bursty. In practice, leave this settings on the default (64).

Examples

```
sonic# configure terminal
sonic(config)# router bgp 65300
sonic(config-router-bgp)# write-quanta 50
```

yellow

Description

This command to configure WRED minimum, maximum thresholds and drop probability for color yellow.

Parent Commands (Modes)

```
qos wred-policy <name>
```

Syntax

```
yellow minimum-threshold { <min-threshold> { maximum-threshold <max-threshold> {
    drop-probability <max-drop-rate> } } }
```

Parameters

Name	Description	Type
min-threshold		Integer
max-threshold		Integer
max-drop-rate		Integer

Usage Guidelines

Use this command to configure WRED minimum, maximum and drop probability for yellow color packets.

Examples

```
sonic(conf-wred-wred-yellow)# yellow minimum-threshold 100 maximum-threshold 200
    drop-probability 50
```

ztp

Description

Administratively enable or disable ZTP.

Parent Commands (Modes)

configure terminal

Syntax

```
ztp enable  
no ztp enable
```

Usage Guidelines

Use these commands to administratively enable or disable ZTP.

Examples

```
sonic# configure terminal  
sonic(config)# ztp enable  
sonic(config)# no ztp enable
```

Features this CLI belongs to

- ZTP